(local) Documentation

LCCHPDev

|  |  |
| --- | --- |
| Server | (local) |
| Author | liam |
| Created | 18 July 2015 13:04 |
| File Path | D:\DBSpace\DB\LCCHP\_documentation-2015-07-18T13-04-37.docx |

# Table of Contents

[Table of Contents 2](#_Toc256000000)

[(local) 10](#_Toc256000001)



[User databases 12](#_Toc256000002)



[LCCHPDev Database 13](#_Toc256000003)



[Tables 15](#_Toc256000004)



[[dbo].[Access­Agreement] 19](#_Toc256000005)



[[dbo].[Access­Agreement­Notes] 22](#_Toc256000006)



[[dbo].[Access­Purpose] 24](#_Toc256000007)



[[dbo].[Action­Status] 26](#_Toc256000008)



[[dbo].[Area] 28](#_Toc256000009)



[[dbo].[Blood­Test­Results] 30](#_Toc256000010)



[[dbo].[Blood­Test­Results­Notes] 36](#_Toc256000011)



[[dbo].[Cleanup­Status] 38](#_Toc256000012)



[[dbo].[Condition] 40](#_Toc256000013)



[[dbo].[Construction­Type] 42](#_Toc256000014)



[[dbo].[Contact­Type] 44](#_Toc256000015)



[[dbo].[Contractor] 46](#_Toc256000016)



[[dbo].[Contractorto­Property] 48](#_Toc256000017)



[[dbo].[Contractorto­Remediation] 50](#_Toc256000018)



[[dbo].[Contractorto­Remediation­Action­Plan] 53](#_Toc256000019)



[[dbo].[Country] 55](#_Toc256000020)



[[dbo].[Data­Source] 57](#_Toc256000021)



[[dbo].[Daycare] 59](#_Toc256000022)



[[dbo].[Daycare­Primary­Contact] 61](#_Toc256000023)



[[dbo].[Daycareto­Property] 63](#_Toc256000024)



[[dbo].[Employer] 65](#_Toc256000025)



[[dbo].[Employerto­Property] 67](#_Toc256000026)



[[dbo].[Environmental­Investigation] 69](#_Toc256000027)



[[dbo].[Error­Log] 72](#_Toc256000028)



[[dbo].[Ethnicity] 74](#_Toc256000029)



[[dbo].[Family] 76](#_Toc256000030)



[[dbo].[Family­Notes] 79](#_Toc256000031)



[[dbo].[Familyto­Phone­Number] 81](#_Toc256000032)



[[dbo].[Familyto­Property] 83](#_Toc256000033)



[[dbo].[File­Type] 86](#_Toc256000034)



[[dbo].[Flag] 88](#_Toc256000035)



[[dbo].[Foreign­Food] 90](#_Toc256000036)



[[dbo].[Foreign­Foodto­Country] 92](#_Toc256000037)



[[dbo].[Frequency] 94](#_Toc256000038)



[[dbo].[Gift­Card] 96](#_Toc256000039)



[[dbo].[Historic­Contribution] 98](#_Toc256000040)



[[dbo].[Hobby] 100](#_Toc256000041)



[[dbo].[Home­Remedy] 102](#_Toc256000042)



[[dbo].[Household­Sourcesof­Lead] 104](#_Toc256000043)



[[dbo].[Insurance­Provider] 106](#_Toc256000044)



[[dbo].[Lab] 108](#_Toc256000045)



[[dbo].[Lab­Notes] 110](#_Toc256000046)



[[dbo].[Language] 112](#_Toc256000047)



[[dbo].[LCCHPAttachments] 114](#_Toc256000048)



[[dbo].[Medium] 115](#_Toc256000049)



[[dbo].[Medium­Sample­Results] 118](#_Toc256000050)



[[dbo].[Medium­Sample­Results­Notes] 122](#_Toc256000051)



[[dbo].[Method] 124](#_Toc256000052)



[[dbo].[Occupation] 126](#_Toc256000053)



[[dbo].[Occupation­Notes] 128](#_Toc256000054)



[[dbo].[Person] 130](#_Toc256000055)



[[dbo].[Person­Hobby­Notes] 137](#_Toc256000056)



[[dbo].[Person­Notes] 139](#_Toc256000057)



[[dbo].[Person­Release­Notes] 141](#_Toc256000058)



[[dbo].[Person­Status] 143](#_Toc256000059)



[[dbo].[Personto­Access­Agreement] 145](#_Toc256000060)



[[dbo].[Personto­Daycare] 147](#_Toc256000061)



[[dbo].[Personto­Employer] 149](#_Toc256000062)



[[dbo].[Personto­Ethnicity] 151](#_Toc256000063)



[[dbo].[Personto­Family] 153](#_Toc256000064)



[[dbo].[Personto­Foreign­Food] 155](#_Toc256000065)



[[dbo].[Personto­Hobby] 157](#_Toc256000066)



[[dbo].[Personto­Home­Remedy] 159](#_Toc256000067)



[[dbo].[Personto­Insurance] 161](#_Toc256000068)



[[dbo].[Personto­Language] 163](#_Toc256000069)



[[dbo].[Personto­Occupation] 165](#_Toc256000070)



[[dbo].[Personto­Person] 167](#_Toc256000071)



[[dbo].[Personto­Phone­Number] 170](#_Toc256000072)



[[dbo].[Personto­Property] 172](#_Toc256000073)



[[dbo].[Person­To­Travel­Country] 175](#_Toc256000074)



[[dbo].[Person­Travel­Notes] 177](#_Toc256000075)



[[dbo].[Phone­Number] 179](#_Toc256000076)



[[dbo].[Phone­Number­Type] 182](#_Toc256000077)



[[dbo].[Property] 184](#_Toc256000078)



[[dbo].[Property­Link­Type] 189](#_Toc256000079)



[[dbo].[Property­Notes] 191](#_Toc256000080)



[[dbo].[Property­Sample­Results] 193](#_Toc256000081)



[[dbo].[Property­Sample­Results­Notes] 196](#_Toc256000082)



[[dbo].[Propertyto­Cleanup­Status] 198](#_Toc256000083)



[[dbo].[Propertyto­Household­Sourcesof­Lead] 200](#_Toc256000084)



[[dbo].[Propertyto­Medium] 202](#_Toc256000085)



[[dbo].[Questionnaire] 204](#_Toc256000086)



[[dbo].[Questionnaire­Data­Source] 211](#_Toc256000087)



[[dbo].[Questionnaire­Notes] 213](#_Toc256000088)



[[dbo].[Relationship­Type] 215](#_Toc256000089)



[[dbo].[Release­Status] 217](#_Toc256000090)



[[dbo].[Remediation] 219](#_Toc256000091)



[[dbo].[Remediation­Action­Plan] 222](#_Toc256000092)



[[dbo].[Remediation­Notes] 225](#_Toc256000093)



[[dbo].[Review­Status] 227](#_Toc256000094)



[[dbo].[Sample­Level­Category] 229](#_Toc256000095)



[[dbo].[Sample­Purpose] 231](#_Toc256000096)



[[dbo].[Sample­Type] 233](#_Toc256000097)



[[dbo].[Source] 235](#_Toc256000098)



[[dbo].[Target­Status] 236](#_Toc256000099)



[[dbo].[Travel­Notes] 238](#_Toc256000100)



[[dbo].[Units] 240](#_Toc256000101)



[Views 242](#_Toc256000102)



[[dbo].[v­Adults] 243](#_Toc256000103)



[[dbo].[v­Most­Recent­Blood­Test­Results] 245](#_Toc256000104)



[[dbo].[v­Most­Recent­Questionnaires] 247](#_Toc256000105)



[[dbo].[v­Nursing­Infants] 250](#_Toc256000106)



[[dbo].[v­Nursing­Mothers] 251](#_Toc256000107)



[[dbo].[v­Pregnant] 252](#_Toc256000108)



[Stored Procedures 253](#_Toc256000109)



[[dbo].[DELETE\_usp\_­Insert­Personto­Status] 257](#_Toc256000110)



[[dbo].[DELETE\_usp\_­Sl­Count­Clients] 259](#_Toc256000111)



[[dbo].[Trans­Proc] 262](#_Toc256000112)



[[dbo].[usp\_­Insert­Access­Agreement] 263](#_Toc256000113)



[[dbo].[usp\_­Insert­Access­Purpose] 266](#_Toc256000114)



[[dbo].[usp\_­Insert­Area] 268](#_Toc256000115)



[[dbo].[usp\_­Insert­Blood­Test­Results] 270](#_Toc256000116)



[[dbo].[usp\_­Insert­Blood­Test­Results­Notes] 274](#_Toc256000117)



[[dbo].[usp\_­Insert­Cleanup­Status] 276](#_Toc256000118)



[[dbo].[usp\_­Insert­Construction­Type] 278](#_Toc256000119)



[[dbo].[usp\_­Insert­Contractor] 280](#_Toc256000120)



[[dbo].[usp\_­Insert­Contractorto­Property] 282](#_Toc256000121)



[[dbo].[usp\_­Insert­Contractorto­Remediation] 284](#_Toc256000122)



[[dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan] 286](#_Toc256000123)



[[dbo].[usp\_­Insert­Country] 288](#_Toc256000124)



[[dbo].[usp\_­Insert­Daycare] 290](#_Toc256000125)



[[dbo].[usp\_­Insert­Daycare­Primary­Contact] 292](#_Toc256000126)



[[dbo].[usp\_­Insert­Daycareto­Property] 294](#_Toc256000127)



[[dbo].[usp\_­Insert­Employer] 296](#_Toc256000128)



[[dbo].[usp\_­Insert­Employerto­Property] 298](#_Toc256000129)



[[dbo].[usp\_­Insert­Environmental­Investigation] 300](#_Toc256000130)



[[dbo].[usp\_­Insert­Ethnicity] 303](#_Toc256000131)



[[dbo].[usp\_­Insert­Family] 305](#_Toc256000132)



[[dbo].[usp\_­Insert­Family­Notes] 308](#_Toc256000133)



[[dbo].[usp\_­Insert­Familyto­Phone­Number] 310](#_Toc256000134)



[[dbo].[usp\_­Insert­Familyto­Property] 314](#_Toc256000135)



[[dbo].[usp\_­Insert­Foreign­Food] 317](#_Toc256000136)



[[dbo].[usp\_­Insert­Foreign­Foodto­Country] 319](#_Toc256000137)



[[dbo].[usp\_­Insert­Gift­Card] 321](#_Toc256000138)



[[dbo].[usp\_­Insert­Hobby] 323](#_Toc256000139)



[[dbo].[usp\_­Insert­Home­Remedies] 325](#_Toc256000140)



[[dbo].[usp\_­Insert­Household­Sourcesof­Lead] 327](#_Toc256000141)



[[dbo].[usp\_­Insert­Insurance­Provider] 329](#_Toc256000142)



[[dbo].[usp\_­Insert­Lab] 331](#_Toc256000143)



[[dbo].[usp\_­Insert­Lab­Notes] 333](#_Toc256000144)



[[dbo].[usp\_­Insert­Language] 335](#_Toc256000145)



[[dbo].[usp\_­Insert­Medium] 337](#_Toc256000146)



[[dbo].[usp\_­Insert­Medium­Sample­Results] 339](#_Toc256000147)



[[dbo].[usp\_­Insert­Medium­Sample­Results­Notes] 342](#_Toc256000148)



[[dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen] 344](#_Toc256000149)



[[dbo].[usp\_­Insert­New­Client­Web­Screen] 348](#_Toc256000150)



[[dbo].[usp\_­Insert­New­Family­Web­Screen] 353](#_Toc256000151)



[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen] 360](#_Toc256000152)



[[dbo].[usp\_­Insert­Occupation] 365](#_Toc256000153)



[[dbo].[usp\_­Insert­Person] 367](#_Toc256000154)



[[dbo].[usp\_­Insert­Person­Hobby­Notes] 371](#_Toc256000155)



[[dbo].[usp\_­Insert­Person­Notes] 373](#_Toc256000156)



[[dbo].[usp\_­Insert­Person­Release­Notes] 375](#_Toc256000157)



[[dbo].[usp\_­Insert­Personto­Access­Agreement] 377](#_Toc256000158)



[[dbo].[usp\_­Insert­Personto­Daycare] 379](#_Toc256000159)



[[dbo].[usp\_­Insert­Personto­Employer] 381](#_Toc256000160)



[[dbo].[usp\_­Insert­Personto­Ethnicity] 383](#_Toc256000161)



[[dbo].[usp\_­Insert­Personto­Family] 385](#_Toc256000162)



[[dbo].[usp\_­Insert­Personto­Foreign­Food] 387](#_Toc256000163)



[[dbo].[usp\_­Insert­Personto­Hobby] 389](#_Toc256000164)



[[dbo].[usp\_­Insert­Personto­Home­Remedy] 391](#_Toc256000165)



[[dbo].[usp\_­Insert­Personto­Insurance] 393](#_Toc256000166)



[[dbo].[usp\_­Insert­Personto­Language] 395](#_Toc256000167)



[[dbo].[usp\_­Insert­Personto­Occupation] 398](#_Toc256000168)



[[dbo].[usp\_­Insert­Personto­Person] 400](#_Toc256000169)



[[dbo].[usp\_­Insert­Personto­Phone­Number] 403](#_Toc256000170)



[[dbo].[usp\_­Insert­Personto­Property] 405](#_Toc256000171)



[[dbo].[usp\_­Insert­Personto­Travel­Country] 407](#_Toc256000172)



[[dbo].[usp\_­Insert­Person­Travel­Notes] 409](#_Toc256000173)



[[dbo].[usp\_­Insert­Phone­Number] 411](#_Toc256000174)



[[dbo].[usp\_­Insert­Phone­Number­Type] 414](#_Toc256000175)



[[dbo].[usp\_­Insert­Property] 416](#_Toc256000176)



[[dbo].[usp\_­Insert­Property­Notes] 420](#_Toc256000177)



[[dbo].[usp\_­Insert­Property­Sample­Results] 422](#_Toc256000178)



[[dbo].[usp\_­Insert­Property­Sample­Results­Notes] 425](#_Toc256000179)



[[dbo].[usp\_­Insert­Propertyto­Cleanup­Status] 427](#_Toc256000180)



[[dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead] 429](#_Toc256000181)



[[dbo].[usp\_­Insert­Propertyto­Medium] 431](#_Toc256000182)



[[dbo].[usp\_­Insert­Questionnaire] 433](#_Toc256000183)



[[dbo].[usp\_­Insert­Questionnaire­Notes] 437](#_Toc256000184)



[[dbo].[usp\_­Insert­Remediation] 439](#_Toc256000185)



[[dbo].[usp\_­Insert­Remediation­Action­Plan] 442](#_Toc256000186)



[[dbo].[usp\_­Insert­Remediation­Notes] 445](#_Toc256000187)



[[dbo].[usp\_­Insert­Sample­Level­Category] 447](#_Toc256000188)



[[dbo].[usp\_­Insert­Sample­Type] 449](#_Toc256000189)



[[dbo].[usp\_­Insert­Status] 451](#_Toc256000190)



[[dbo].[usp\_­Insert­Travel­Notes] 453](#_Toc256000191)



[[dbo].[usp\_­SLAll­Blood­Test­Results] 455](#_Toc256000192)



[[dbo].[usp\_­SLAll­Blood­Test­Results2] 458](#_Toc256000193)



[[dbo].[usp\_­SLAll­Blood­Test­Results­Meta­Data] 461](#_Toc256000194)



[[dbo].[usp\_­Sl­Child­Status] 463](#_Toc256000195)



[[dbo].[usp\_­Sl­Client­Follow­Up] 465](#_Toc256000196)



[[dbo].[usp\_­Sl­Column­Details] 468](#_Toc256000197)



[[dbo].[usp\_­Sl­Count­Adults] 470](#_Toc256000198)



[[dbo].[usp\_­Sl­Count­Blood­Lead­Levels] 473](#_Toc256000199)



[[dbo].[usp\_­Sl­Count­Blood­Tests] 476](#_Toc256000200)



[[dbo].[usp\_­Sl­Count­Clients] 479](#_Toc256000201)



[[dbo].[usp\_­Sl­Count­Family­Members] 482](#_Toc256000202)



[[dbo].[usp\_­Sl­Count­Home­Visit­Soil­Sample] 485](#_Toc256000203)



[[dbo].[usp\_­Sl­Count­New­Clients] 488](#_Toc256000204)



[[dbo].[usp\_­Sl­Count­New­People] 491](#_Toc256000205)



[[dbo].[usp\_­Sl­Count­Nursing­Infants] 494](#_Toc256000206)



[[dbo].[usp\_­Sl­Count­Nursing­Mothers] 497](#_Toc256000207)



[[dbo].[usp\_­Sl­Count­People] 500](#_Toc256000208)



[[dbo].[usp\_­Sl­Count­People­By­Age] 502](#_Toc256000209)



[[dbo].[usp\_­Sl­Count­People­By­Age­Group] 504](#_Toc256000210)



[[dbo].[usp\_­Sl­Count­People­By­Last­Name] 506](#_Toc256000211)



[[dbo].[usp\_­Sl­Count­Pregnant­Women] 508](#_Toc256000212)



[[dbo].[usp\_­Sl­Daycare] 511](#_Toc256000213)



[[dbo].[usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information] 513](#_Toc256000214)



[[dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information] 516](#_Toc256000215)



[[dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information] 519](#_Toc256000216)



[[dbo].[usp\_­Sl­Edit­Property­Web­Screen­Information] 522](#_Toc256000217)



[[dbo].[usp\_­Sl­Edit­Questionnaire­Web­Screen­Information] 525](#_Toc256000218)



[[dbo].[usp\_­Sl­Family­Members] 528](#_Toc256000219)



[[dbo].[usp\_­Sl­Family­Nameto­Property] 531](#_Toc256000220)



[[dbo].[usp\_­Sl­Hobby] 533](#_Toc256000221)



[[dbo].[usp\_­SLInserted­Data] 535](#_Toc256000222)



[[dbo].[usp\_­SLInserted­Data­Simplified] 539](#_Toc256000223)



[[dbo].[usp\_­Sl­Lab­Name] 542](#_Toc256000224)



[[dbo].[usp\_­SLList­All­Family­Members] 544](#_Toc256000225)



[[dbo].[usp\_­Sl­List­Clients­By­Createdate] 547](#_Toc256000226)



[[dbo].[usp\_­Sl­List­Clients­By­Modifieddate] 550](#_Toc256000227)



[[dbo].[usp\_­Sl­List­Families] 553](#_Toc256000228)



[[dbo].[usp\_­Sl­List­Family­Members] 555](#_Toc256000229)



[[dbo].[usp\_­Sl­List­Nursing­Womenby­Create­Date­Range] 558](#_Toc256000230)



[[dbo].[usp\_­Sl­List­Peopleby­Create­Date­Range] 561](#_Toc256000231)



[[dbo].[usp\_­SLList­Potential­Duplicate­People] 564](#_Toc256000232)



[[dbo].[usp\_­SLList­Potential­Duplicate­Properties] 566](#_Toc256000233)



[[dbo].[usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range] 568](#_Toc256000234)



[[dbo].[usp\_­SLMost­Recent­Blood­Test­Results] 571](#_Toc256000235)



[[dbo].[usp\_­Sl­Person­Notes] 574](#_Toc256000236)



[[dbo].[usp\_­Sl­Personto­Ethnicity] 577](#_Toc256000237)



[[dbo].[usp\_­Sl­Personto­Language] 580](#_Toc256000238)



[[dbo].[usp\_­Sl­Relation­Ship­Types] 583](#_Toc256000239)



[[dbo].[usp\_­Sl­Status] 585](#_Toc256000240)



[[dbo].[usp\_­Sl­Summary­Report] 587](#_Toc256000241)



[[dbo].[usp\_­Sl­Summary­Report\_­Meta­Data] 597](#_Toc256000242)



[[dbo].[usp\_­Sl­Target­Sample­Type] 600](#_Toc256000243)



[[dbo].[usp\_up­Blood­Test­Results] 602](#_Toc256000244)



[[dbo].[usp\_up­Blood­Test­Results­Web­Screen] 606](#_Toc256000245)



[[dbo].[usp\_up­Client­Flag] 610](#_Toc256000246)



[[dbo].[usp\_up­Client­Web­Screen] 612](#_Toc256000247)



[[dbo].[usp\_up­Family] 618](#_Toc256000248)



[[dbo].[usp\_up­Familyto­Property] 622](#_Toc256000249)



[[dbo].[usp\_up­Family­Web­Screen] 625](#_Toc256000250)



[[dbo].[usp\_up­Occupation] 632](#_Toc256000251)



[[dbo].[usp\_up­Person] 635](#_Toc256000252)



[[dbo].[usp\_up­Property] 642](#_Toc256000253)



[[dbo].[usp\_up­Questionnaire] 648](#_Toc256000254)



[[dbo].[usp\_up­Questionnaire­Web­Screen] 654](#_Toc256000255)



[[dbo].[usp­Log­Error] 658](#_Toc256000256)



[[dbo].[usp­Print­Error] 663](#_Toc256000257)



[Scalar-valued Functions 667](#_Toc256000258)



[[dbo].[Remove­Special­Chars] 668](#_Toc256000259)



[[dbo].[udf\_­Calculate­Age] 670](#_Toc256000260)



[[dbo].[udf\_­Date­In­The­Past] 672](#_Toc256000261)



[[dbo].[udf\_­Does­Property­Exist] 674](#_Toc256000262)



[[dbo].[udf\_­Sl­Family­Phone­Number] 676](#_Toc256000263)



[Users 678](#_Toc256000264)



[app­User 679](#_Toc256000265)



[WIN-1M8NQQ69OEH\SQLMaintenenace 680](#_Toc256000266)



[Database Roles 681](#_Toc256000267)



[db\_accessadmin 681](#_Toc256000268)



[db\_backupoperator 681](#_Toc256000269)



[db\_datareader 682](#_Toc256000270)



[db\_datawriter 682](#_Toc256000271)



[db\_ddladmin 683](#_Toc256000272)



[db\_denydatareader 683](#_Toc256000273)



[db\_denydatawriter 684](#_Toc256000274)



[db\_owner 684](#_Toc256000275)



[db\_securityadmin 684](#_Toc256000276)



[public 685](#_Toc256000277)



|  |
| --- |
| (local) |

Databases(1)

* [LCCHPDev](#(local)/User_databases/LCCHPDev/index)



Server Properties

|  |  |
| --- | --- |
| Property | Value |
| Product | Microsoft SQL Server |
| Version | 11.0.5058.0 |
| Language | English (United States) |
| Platform | NT x64 |
| Edition | Express Edition (64-bit) |
| Processors | 2 |
| OS Version | 6.2 (9200) |
| Physical Memory | 4096 |
| Is Clustered | False |
| Root Directory | C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |

Server Settings

|  |  |
| --- | --- |
| Property | Value |
| Default data file path | D:\MSSQL\Data\ |
| Default backup file path | Y:\Backup\WIN-1M8NQQ69OEH |
| Default log file path | D:\MSSQL\Log\ |
| Recovery Interval (minutes) | 0 |
| Default index fill factor | 0 |
| Default backup media retention | 0 |

Advanced Server Settings

|  |  |
| --- | --- |
| Property | Value |
| Full text upgrade option | 2 |
| Locks | 0 |
| Nested triggers enabled | True |
| Allow triggers to fire others | True |
| Default language | English |
| Network packet size | 4096 |
| Default fulltext language LCID | 1033 |
| Two-digit year cutoff | 2049 |
| Remote login timeout | 10 |
| Cursor threshold | -1 |
| Max text replication size | 65536 |
| Parallelism cost threshold | 25 |
| Scan for startup procs | False |
| Transform noise words | False |
| Blocked process threshold | 0 |
| Filestream access level | 2 |
| Optimize for ad hoc workloads | True |

|  |
| --- |
| User databases |

Databases(1)

* [LCCHPDev](#(local)/User_databases/LCCHPDev/index)



|  |
| --- |
| LCCHPDev Database |

Database Properties

|  |  |
| --- | --- |
| Property | Value |
| SQL Server Version | SQL Server 2012 |
| Compatibility Level | SQL Server 2012 |
| Database Encryption Enabled | False |
| Last backup time | 07/18/2015 |
| Last log backup time | 07/18/2015 |
| Creation date | Jan 2 2015 |
| Users | 6 |
| Database size | 45.00 MB |
| Unallocated space | 5.47 MB |

Database Options

|  |  |
| --- | --- |
| Property | Value |
| Compatibility Level | 110 |
| Database collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Restrict access | MULTI\_­USER |
| Is read-only | False |
| Auto close | False |
| Auto shrink | False |
| Database status | ONLINE |
| In standby | False |
| Cleanly shutdown | False |
| Supplemental logging enabled | False |
| Snapshot isolation state | OFF |
| Read committed snapshot on | False |
| Recovery model | FULL |
| Page verify option | CHECKSUM |
| Auto create statistics | True |
| Auto update statistics | True |
| Auto update statistics asynchronously | False |
| ANSI NULL default | False |
| ANSI NULL enabled | False |
| ANSI padding enabled | False |
| ANSI warnings enabled | False |
| Arithmetic abort enabled | False |
| Concatenating NULL yields NULL | False |
| Numeric roundabort enabled | False |
| Quoted Identifier On | False |
| Recursive triggers enabled | False |
| Close cursors on commit | False |
| Local cursors by default | False |
| Fulltext enabled | True |
| Trustworthy | False |
| Database chaining | False |
| Forced parameterization | False |
| Master key encrypted by server | False |
| Published | False |
| Subscribed | False |
| Merge published | False |
| Is distribution database | False |
| Sync with backup | False |
| Service broker GUID | 5e0a2947-6106-454c-a34c-8e1821adb0ff |
| Service broker enabled | False |
| Log reuse wait | LOG\_­BACKUP |
| Date correlation | False |
| CDC enabled | False |
| Encrypted | False |
| Honor broker priority | False |
| Default language | English |
| Default fulltext language LCID | 1033 |
| Nested triggers enabled | True |
| Transform noise words | False |
| Two-digit year cutoff | 2049 |
| Containment | NONE |
| Target recovery time | 0 |
| Database owner | WIN-1M8NQQ69OEH\liam |

Files

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | File Group | Size | File Name |
| LCCHP | Data |  | 8.00 MB | D:\MSSQL\Data\LCCHPDev.mdf |
| LCCHP\_log | Log |  | 25.00 MB | D:\MSSQL\Log\LCCHPDev\_log.ldf |
| LCCHP\_­UData | Data | UData | 12.00 MB | D:\MSSQL\Data\LCCHPDev\_­UData.ndf |
| LCCHPAttachments | Filestream |  |  | D:\MSSQL\Filestream\LCCHPAttachments­Dev |

|  |
| --- |
| Tables |

Objects

|  |
| --- |
| Name |
| [dbo.Access­Agreement](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement) collection of access agreements |
| [dbo.Access­Agreement­Notes](#(local)/User_databases/LCCHPDev/Tables/AccessAgreementNotes) linking table for access agreement and access agreement notes |
| [dbo.Access­Purpose](#(local)/User_databases/LCCHPDev/Tables/AccessPurpose) collection of purposes for access requests/agreements |
| [dbo.Action­Status](#(local)/User_databases/LCCHPDev/Tables/ActionStatus) Collection of potential status for Action |
| [dbo.Area](#(local)/User_databases/LCCHPDev/Tables/Area) collection of areas and basic information |
| [dbo.Blood­Test­Results](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults) Collection of blood test result values and categorization |
| [dbo.Blood­Test­Results­Notes](#(local)/User_databases/LCCHPDev/Tables/BloodTestResultsNotes) linking table for access agreement and access agreement notes |
| [dbo.Cleanup­Status](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus) collection of clean up status |
| [dbo.Condition](#(local)/User_databases/LCCHPDev/Tables/Condition) Collection of potential status for Action |
| [dbo.Construction­Type](#(local)/User_databases/LCCHPDev/Tables/ConstructionType) collection of construction types |
| [dbo.Contact­Type](#(local)/User_databases/LCCHPDev/Tables/ContactType) Collection of contact types |
| [dbo.Contractor](#(local)/User_databases/LCCHPDev/Tables/Contractor) |
| [dbo.Contractorto­Property](#(local)/User_databases/LCCHPDev/Tables/ContractortoProperty) linking table for contractor and occupied properties |
| [dbo.Contractorto­Remediation](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediation) linking table for contractors and remediations |
| [dbo.Contractorto­Remediation­Action­Plan](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediationActionPlan) linking table for contractor and sampling plan |
| [dbo.Country](#(local)/User_databases/LCCHPDev/Tables/Country) collection of countries |
| [dbo.Data­Source](#(local)/User_databases/LCCHPDev/Tables/DataSource) Collection of contact types |
| [dbo.Daycare](#(local)/User_databases/LCCHPDev/Tables/Daycare) collection of daycare facilities |
| [dbo.Daycare­Primary­Contact](#(local)/User_databases/LCCHPDev/Tables/DaycarePrimaryContact) linking table for daycare and person - identifying contact person |
| [dbo.Daycareto­Property](#(local)/User_databases/LCCHPDev/Tables/DaycaretoProperty) linking table for daycare and property |
| [dbo.Employer](#(local)/User_databases/LCCHPDev/Tables/Employer) collection of employers |
| [dbo.Employerto­Property](#(local)/User_databases/LCCHPDev/Tables/EmployertoProperty) linking table for employer and property |
| [dbo.Environmental­Investigation](#(local)/User_databases/LCCHPDev/Tables/EnvironmentalInvestigation) |
| [dbo.Error­Log](#(local)/User_databases/LCCHPDev/Tables/ErrorLog) |
| [dbo.Ethnicity](#(local)/User_databases/LCCHPDev/Tables/Ethnicity) collection of ethnicities |
| [dbo.Family](#(local)/User_databases/LCCHPDev/Tables/Family) collection of families |
| [dbo.Family­Notes](#(local)/User_databases/LCCHPDev/Tables/FamilyNotes) table for Family notes |
| [dbo.Familyto­Phone­Number](#(local)/User_databases/LCCHPDev/Tables/FamilytoPhoneNumber) linking table for Family and phonenumber |
| [dbo.Familyto­Property](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty) linking table for Family and property - indicating when a Family occuppied a property |
| [dbo.File­Type](#(local)/User_databases/LCCHPDev/Tables/FileType) |
| [dbo.Flag](#(local)/User_databases/LCCHPDev/Tables/Flag) Collection of flag information |
| [dbo.Foreign­Food](#(local)/User_databases/LCCHPDev/Tables/ForeignFood) collection of various foreign foods |
| [dbo.Foreign­Foodto­Country](#(local)/User_databases/LCCHPDev/Tables/ForeignFoodtoCountry) foreign food and country linking table |
| [dbo.Frequency](#(local)/User_databases/LCCHPDev/Tables/Frequency) Collection of frequencies |
| [dbo.Gift­Card](#(local)/User_databases/LCCHPDev/Tables/GiftCard) collection of gift certificate objects |
| [dbo.Historic­Contribution](#(local)/User_databases/LCCHPDev/Tables/HistoricContribution) Collection of historic contribution classifications |
| [dbo.Hobby](#(local)/User_databases/LCCHPDev/Tables/Hobby) collection of hobbies |
| [dbo.Home­Remedy](#(local)/User_databases/LCCHPDev/Tables/HomeRemedy) collection of home remedies |
| [dbo.Household­Sourcesof­Lead](#(local)/User_databases/LCCHPDev/Tables/HouseholdSourcesofLead) household items that may contribute to EBL |
| [dbo.Insurance­Provider](#(local)/User_databases/LCCHPDev/Tables/InsuranceProvider) collection of insurance companies |
| [dbo.Lab](#(local)/User_databases/LCCHPDev/Tables/Lab) collection of lab names and basic attributes |
| [dbo.Lab­Notes](#(local)/User_databases/LCCHPDev/Tables/LabNotes) linking table for access agreement and access agreement notes |
| [dbo.Language](#(local)/User_databases/LCCHPDev/Tables/Language) collection of spoken languages |
| [dbo.LCCHPAttachments](#(local)/User_databases/LCCHPDev/Tables/LCCHPAttachments) |
| [dbo.Medium](#(local)/User_databases/LCCHPDev/Tables/Medium) collection of mediums that are tested |
| [dbo.Medium­Sample­Results](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults) collection of test results for various medums |
| [dbo.Medium­Sample­Results­Notes](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResultsNotes) linking table for access agreement and access agreement notes |
| [dbo.Method](#(local)/User_databases/LCCHPDev/Tables/Method) Collection of method classifications |
| [dbo.Occupation](#(local)/User_databases/LCCHPDev/Tables/Occupation) collection of occupation objects |
| [dbo.Occupation­Notes](#(local)/User_databases/LCCHPDev/Tables/OccupationNotes) table for Occupation notes |
| [dbo.Person](#(local)/User_databases/LCCHPDev/Tables/Person) collection of people and basic attributes |
| [dbo.Person­Hobby­Notes](#(local)/User_databases/LCCHPDev/Tables/PersonHobbyNotes) table for person hobby notes |
| [dbo.Person­Notes](#(local)/User_databases/LCCHPDev/Tables/PersonNotes) table for person notes |
| [dbo.Person­Release­Notes](#(local)/User_databases/LCCHPDev/Tables/PersonReleaseNotes) table for person release notes |
| [dbo.Person­Status](#(local)/User_databases/LCCHPDev/Tables/PersonStatus) Collection of potential status for person |
| [dbo.Personto­Access­Agreement](#(local)/User_databases/LCCHPDev/Tables/PersontoAccessAgreement) linking table for person and access agreement |
| [dbo.Personto­Daycare](#(local)/User_databases/LCCHPDev/Tables/PersontoDaycare) linking table for person and daycare for people attending daycare |
| [dbo.Personto­Employer](#(local)/User_databases/LCCHPDev/Tables/PersontoEmployer) linking table for person and employer |
| [dbo.Personto­Ethnicity](#(local)/User_databases/LCCHPDev/Tables/PersontoEthnicity) linking table for person and ethnicity |
| [dbo.Personto­Family](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily) linking table for person and family tables |
| [dbo.Personto­Foreign­Food](#(local)/User_databases/LCCHPDev/Tables/PersontoForeignFood) linking table for person and foreign food (many to many) |
| [dbo.Personto­Hobby](#(local)/User_databases/LCCHPDev/Tables/PersontoHobby) linking table for person and hobby |
| [dbo.Personto­Home­Remedy](#(local)/User_databases/LCCHPDev/Tables/PersontoHomeRemedy) linking table for perosn and home remedy |
| [dbo.Personto­Insurance](#(local)/User_databases/LCCHPDev/Tables/PersontoInsurance) linking table for person and insurance |
| [dbo.Personto­Language](#(local)/User_databases/LCCHPDev/Tables/PersontoLanguage) linking table for person and language |
| [dbo.Personto­Occupation](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation) linking table for person and occupatoin |
| [dbo.Personto­Person](#(local)/User_databases/LCCHPDev/Tables/PersontoPerson) collection of relationships between people |
| [dbo.Personto­Phone­Number](#(local)/User_databases/LCCHPDev/Tables/PersontoPhoneNumber) linking table for person and phonenumber |
| [dbo.Personto­Property](#(local)/User_databases/LCCHPDev/Tables/PersontoProperty) linking table for person and property - indicating when a person occuppied a property |
| [dbo.Person­To­Travel­Country](#(local)/User_databases/LCCHPDev/Tables/PersonToTravelCountry) linking table for person and country traveled too |
| [dbo.Person­Travel­Notes](#(local)/User_databases/LCCHPDev/Tables/PersonTravelNotes) table for person Travel notes |
| [dbo.Phone­Number](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber) collection of phone number objects |
| [dbo.Phone­Number­Type](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType) |
| [dbo.Property](#(local)/User_databases/LCCHPDev/Tables/Property) collection of properties and basic attributes |
| [dbo.Property­Link­Type](#(local)/User_databases/LCCHPDev/Tables/PropertyLinkType) Collection of property link types |
| [dbo.Property­Notes](#(local)/User_databases/LCCHPDev/Tables/PropertyNotes) linking table for access agreement and access agreement notes |
| [dbo.Property­Sample­Results](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults) collection of property test results |
| [dbo.Property­Sample­Results­Notes](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResultsNotes) linking table for access agreement and access agreement notes |
| [dbo.Propertyto­Cleanup­Status](#(local)/User_databases/LCCHPDev/Tables/PropertytoCleanupStatus) linking table for property and cleanup status |
| [dbo.Propertyto­Household­Sourcesof­Lead](#(local)/User_databases/LCCHPDev/Tables/PropertytoHouseholdSourcesofLead) linking table for property and household sources of lead |
| [dbo.Propertyto­Medium](#(local)/User_databases/LCCHPDev/Tables/PropertytoMedium) linking table for property and media |
| [dbo.Questionnaire](#(local)/User_databases/LCCHPDev/Tables/Questionnaire) collection of questionnaire questions and answers, typically only completed by flagged patients |
| [dbo.Questionnaire­Data­Source](#(local)/User_databases/LCCHPDev/Tables/QuestionnaireDataSource) source of the data (Environmental group or Blood Lead) |
| [dbo.Questionnaire­Notes](#(local)/User_databases/LCCHPDev/Tables/QuestionnaireNotes) linking table for access agreement and access agreement notes |
| [dbo.Relationship­Type](#(local)/User_databases/LCCHPDev/Tables/RelationshipType) collection of Relationship­Type names and basic attributes |
| [dbo.Release­Status](#(local)/User_databases/LCCHPDev/Tables/ReleaseStatus) Collection of Release Status |
| [dbo.Remediation](#(local)/User_databases/LCCHPDev/Tables/Remediation) collection of remediation data |
| [dbo.Remediation­Action­Plan](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan) collection of sampling plans |
| [dbo.Remediation­Notes](#(local)/User_databases/LCCHPDev/Tables/RemediationNotes) table for remediation notes |
| [dbo.Review­Status](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus) Collection of potential status for Review |
| [dbo.Sample­Level­Category](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory) collection of sample level categorizations |
| [dbo.Sample­Purpose](#(local)/User_databases/LCCHPDev/Tables/SamplePurpose) Collection of sample purposes |
| [dbo.Sample­Type](#(local)/User_databases/LCCHPDev/Tables/SampleType) collection of sample types |
| [dbo.Source](#(local)/User_databases/LCCHPDev/Tables/Source) |
| [dbo.Target­Status](#(local)/User_databases/LCCHPDev/Tables/TargetStatus) collection of status objects |
| [dbo.Travel­Notes](#(local)/User_databases/LCCHPDev/Tables/TravelNotes) Collection of family and travel notes |
| [dbo.Units](#(local)/User_databases/LCCHPDev/Tables/Units) |

|  |
| --- |
| [dbo].[Access­Agreement] |

MS\_­Description

collection of access agreements

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 5 |
| Created | 7:28:57 PM Friday, December 26, 2014 |
| Last Modified | 10:36:24 AM Saturday, February 14, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Access­Agreement­ID | int | 4 | False | 1 - 1 |  |
|  | Access­Purpose­ID id of the access purpose | int | 4 | True |  |  |
|  | Access­Agreement­File | varbinary(max) | max | True |  |  |
|  | Property­ID | int | 4 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Access­Agreement | Access­Agreement­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Access­Agreement | True | True | After Update |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Access­Agreement\_­Access­Purpose | Access­Purpose­ID->[[dbo].[Access­Purpose].[Access­Purpose­ID]](#(local)/User_databases/LCCHPDev/Tables/AccessPurpose) |
| FK\_­Access­Agreement\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Access­Agreement]  (  [Access­Agreement­ID] [int] NOT NULL IDENTITY(1, 1),  [Access­Purpose­ID] [int] NULL,  [Access­Agreement­File] [varbinary] (max) NULL,  [Property­ID] [int] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Access­Agreement\_­Created­Date] DEFAULT (getdate())  ) ON [UData] TEXTIMAGE\_­ON [UData]  GO  create trigger [dbo].[tr­Update­Access­Agreement] on [dbo].[Access­Agreement] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Access­Agreement set Modified­Date = getdate() where Access­Agreement­ID in (select Access­Agreement­ID from inserted)  end  GO  ALTER TABLE [dbo].[Access­Agreement] ADD CONSTRAINT [PK\_­Access­Agreement] PRIMARY KEY CLUSTERED ([Access­Agreement­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Access­Agreement] ADD CONSTRAINT [FK\_­Access­Agreement\_­Access­Purpose] FOREIGN KEY ([Access­Purpose­ID]) REFERENCES [dbo].[Access­Purpose] ([Access­Purpose­ID])  GO  ALTER TABLE [dbo].[Access­Agreement] ADD CONSTRAINT [FK\_­Access­Agreement\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of access agreements', 'SCHEMA', N'dbo', 'TABLE', N'Access­Agreement', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the access purpose', 'SCHEMA', N'dbo', 'TABLE', N'Access­Agreement', 'COLUMN', N'Access­Purpose­ID'  GO |

Uses

[[dbo].[Access­Purpose]](#(local)/User_databases/LCCHPDev/Tables/AccessPurpose)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[Access­Agreement­Notes]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreementNotes)

[[dbo].[Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/PersontoAccessAgreement)

[[dbo].[usp\_­Insert­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessAgreement)

|  |
| --- |
| [dbo].[Access­Agreement­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 10:36:24 AM Saturday, February 14, 2015 |
| Last Modified | 10:36:24 AM Saturday, February 14, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Access­Agreement­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Access­Agreement­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Access­Agreement­Notes | Access­Agreement­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Access­Agreement­Notes\_­Access­Agreement | Access­Agreement­ID->[[dbo].[Access­Agreement].[Access­Agreement­ID]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Access­Agreement­Notes]  (  [Access­Agreement­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Access­Agreement­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Access­Agreement­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Access­Agreement­Notes] ADD CONSTRAINT [PK\_­Access­Agreement­Notes] PRIMARY KEY CLUSTERED ([Access­Agreement­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Access­Agreement­Notes] ADD CONSTRAINT [FK\_­Access­Agreement­Notes\_­Access­Agreement] FOREIGN KEY ([Access­Agreement­ID]) REFERENCES [dbo].[Access­Agreement] ([Access­Agreement­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Access­Agreement­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Access­Agreement­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement)

Used By

[[dbo].[usp\_­Insert­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessAgreement)

|  |
| --- |
| [dbo].[Access­Purpose] |

MS\_­Description

collection of purposes for access requests/agreements

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 6 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:06 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Access­Purpose­ID | int | 4 | False | 1 - 1 |  |
|  | Access­Purpose­Name friendly name for the access purpose | varchar(50) | 50 | True |  |  |
|  | Access­Purpose­Description a description of the access purpose | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Access­Purpose | Access­Purpose­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Access­Purpose | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Access­Purpose]  (  [Access­Purpose­ID] [int] NOT NULL IDENTITY(1, 1),  [Access­Purpose­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Access­Purpose­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Access­Purpose\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Access­Purpose] on [dbo].[Access­Purpose] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Access­Purpose set Modified­Date = getdate() where Access­Purpose­ID in (select Access­Purpose­ID from inserted)  end  GO  ALTER TABLE [dbo].[Access­Purpose] ADD CONSTRAINT [PK\_­Access­Purpose] PRIMARY KEY CLUSTERED ([Access­Purpose­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of purposes for access requests/agreements', 'SCHEMA', N'dbo', 'TABLE', N'Access­Purpose', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'a description of the access purpose', 'SCHEMA', N'dbo', 'TABLE', N'Access­Purpose', 'COLUMN', N'Access­Purpose­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'friendly name for the access purpose', 'SCHEMA', N'dbo', 'TABLE', N'Access­Purpose', 'COLUMN', N'Access­Purpose­Name'  GO |

Used By

[[dbo].[Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement)

[[dbo].[usp\_­Insert­Access­Purpose]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessPurpose)

|  |
| --- |
| [dbo].[Action­Status] |

MS\_­Description

Collection of potential status for Action

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 2:12:35 PM Saturday, April 11, 2015 |
| Last Modified | 2:12:35 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Action­Status­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Action­Status­Description Detailed description of the Action status | varchar(253) | 253 | True |  |  |
|  | Action­Status­Name status for the Action | varchar(50) | 50 | True |  |  |
|  | Historic­Action­Status­ID historic status from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Action­Status | Action­Status­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Action­Status]  (  [Action­Status­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Action­Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Action­Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Action­Status­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Action­Status\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Action­Status] ADD CONSTRAINT [PK\_­Action­Status] PRIMARY KEY CLUSTERED ([Action­Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of potential status for Action', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the Action status', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', 'COLUMN', N'Action­Status­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'status for the Action', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', 'COLUMN', N'Action­Status­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic status from access database', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', 'COLUMN', N'Historic­Action­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Action­Status', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Area] |

MS\_­Description

collection of areas and basic information

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 24 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 9:07:29 PM Thursday, April 9, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Area­ID unique identifier of the area | int | 4 | False | 1 - 1 |  |
|  | Area­Description friendly description/name of the area | varchar(253) | 253 | True |  |  |
|  | Historic­Area­ID | varchar(50) | 50 | False |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Area | Area­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Area | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Area]  (  [Area­ID] [int] NOT NULL IDENTITY(1, 1),  [Area­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Area­ID] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Area\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Area] on [dbo].[Area] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Area set Modified­Date = getdate() where Area­ID in (select Area­ID from inserted)  end  GO  ALTER TABLE [dbo].[Area] ADD CONSTRAINT [PK\_­Area] PRIMARY KEY CLUSTERED ([Area­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of areas and basic information', 'SCHEMA', N'dbo', 'TABLE', N'Area', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'friendly description/name of the area', 'SCHEMA', N'dbo', 'TABLE', N'Area', 'COLUMN', N'Area­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the area', 'SCHEMA', N'dbo', 'TABLE', N'Area', 'COLUMN', N'Area­ID'  GO |

Used By

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[usp\_­Insert­Area]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertArea)

|  |
| --- |
| [dbo].[Blood­Test­Results] |

MS\_­Description

Collection of blood test result values and categorization

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 13151 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 7:36:15 PM Thursday, June 11, 2015 |

Columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Computed | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Blood­Test­Results­ID unique identifier for the blood test results object | int |  | 4 | False | 1 - 1 |  |
|  | is­Baseline 0 = no; 1 = yes | bit |  | 1 | False |  | ((0)) |
|  | Person­ID | int |  | 4 | True |  |  |
|  | Sample­Date date the sample was taken | date |  | 3 | False |  | (getdate()) |
|  | Lab­Submission­Date date the sample was submitted to the lab | date |  | 3 | True |  |  |
|  | Lead­Value | numeric(4,1) |  | 5 | True |  |  |
|  | Lead­Value­Category­ID id of the associated lead value categorization | tinyint |  | 1 | True |  |  |
|  | Hemoglobin­Value | numeric(4,1) |  | 5 | True |  |  |
|  | Hemoglobin­Value­Category­ID id of the associated hemoglobin value categorization | tinyint |  | 1 | True |  |  |
|  | Hematocrit­Value­Category­ID id of the associated hematocrit value categorization | tinyint |  | 1 | True |  |  |
|  | Lab­ID id of the lab to which the samples were submitted | int |  | 4 | True |  |  |
|  | Blood­Test­Costs cost of the blood tests | money |  | 8 | True |  |  |
|  | Sample­Type­ID id of the type of sample; i.e. venus, capo, soil, water, nitton analyzer . . . | tinyint |  | 1 | True |  |  |
|  | Taken­After­Property­Remediation­Completed 0 - No, 1 - yes; was the blood sample taken after property remediation was completed. | bit |  | 1 | True |  | ((0)) |
|  | Modified­Date | datetime |  | 8 | True |  |  |
|  | Created­Date | datetime |  | 8 | True |  | (getdate()) |
|  | Hematocrit­Value | numeric(6,1) | True | 5 | True |  |  |
|  | Exclude­Result | bit |  | 1 | True |  |  |
|  | Historic­Blood­Test­Results­ID Historic bloodpbresults id from access database | int |  | 4 | True |  |  |
|  | Historic­Lab­Results­ID historic lab results id from access database | varchar(10) |  | 10 | True |  |  |
|  | Client­Status­ID | smallint |  | 2 | True |  |  |

Computed columns

|  |  |
| --- | --- |
| Name | Column definition |
| Hematocrit­Value | ([hemoglobin­Value]\*(3)) |

Indexes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | Fill Factor | File Group |
|  | PK\_­Blood­Test­Results | Blood­Test­Results­ID | True |  | UData |
|  | IDX\_­Blood­Test­Results­Sample­Date­Lead­Value | Blood­Test­Results­ID, Client­Status­ID, Lab­Submission­Date, Person­ID, Sample­Date, Lead­Value |  | 90 | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Blood­Test­Results | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Blood­Test­Results\_­Sample­Date | Sample­Date | ([dbo].[udf\_­Date­In­The­Past]([Sample­Date])=(1)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Blood­Test­Results\_­Target­Status | Client­Status­ID->[[dbo].[Target­Status].[Status­ID]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus) |
| FK\_­Blood­Test­Results\_­Hematocrit­Level­Category | Hematocrit­Value­Category­ID->[[dbo].[Sample­Level­Category].[Sample­Level­Category­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory) |
| FK\_­Blood­Test­Results\_­Hemoglobin­Level­Category | Hemoglobin­Value­Category­ID->[[dbo].[Sample­Level­Category].[Sample­Level­Category­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory) |
| FK\_­Blood­Test­Results\_­Lab | Lab­ID->[[dbo].[Lab].[Lab­ID]](#(local)/User_databases/LCCHPDev/Tables/Lab) |
| FK\_­Blood­Test­Results\_­Lead­Level­Category | Lead­Value­Category­ID->[[dbo].[Sample­Level­Category].[Sample­Level­Category­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory) |
| FK\_­Blood­Test­Results\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Blood­Test­Results\_­Sample­Type | Sample­Type­ID->[[dbo].[Sample­Type].[Sample­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleType) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Blood­Test­Results]  (  [Blood­Test­Results­ID] [int] NOT NULL IDENTITY(1, 1),  [is­Baseline] [bit] NOT NULL CONSTRAINT [DF\_­Blood­Test­Results\_is­Baseline] DEFAULT ((0)),  [Person­ID] [int] NULL,  [Sample­Date] [date] NOT NULL CONSTRAINT [DF\_­Blood­Test­Results\_­Sample­Date] DEFAULT (getdate()),  [Lab­Submission­Date] [date] NULL,  [Lead­Value] [numeric] (4, 1) NULL,  [Lead­Value­Category­ID] [tinyint] NULL,  [Hemoglobin­Value] [numeric] (4, 1) NULL,  [Hemoglobin­Value­Category­ID] [tinyint] NULL,  [Hematocrit­Value­Category­ID] [tinyint] NULL,  [Lab­ID] [int] NULL,  [Blood­Test­Costs] [money] NULL,  [Sample­Type­ID] [tinyint] NULL,  [Taken­After­Property­Remediation­Completed] [bit] NULL CONSTRAINT [DF\_­Blood­Test­Results\_­Taken­After­Property­Remediation­Completed] DEFAULT ((0)),  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Blood­Test­Results\_­Created­Date] DEFAULT (getdate()),  [Hematocrit­Value] AS ([hemoglobin­Value]\*(3)),  [Exclude­Result] [bit] NULL,  [Historic­Blood­Test­Results­ID] [int] NULL,  [Historic­Lab­Results­ID] [varchar] (10) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Client­Status­ID] [smallint] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Blood­Test­Results] on [dbo].[Blood­Test­Results] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Blood­Test­Results set Modified­Date = getdate() where Blood­Test­Results­ID in (select Blood­Test­Results­ID from inserted)  end  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [ck\_­Blood­Test­Results\_­Sample­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Sample­Date])=(1)))  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [PK\_­Blood­Test­Results] PRIMARY KEY CLUSTERED ([Blood­Test­Results­ID]) ON [UData]  GO  CREATE NONCLUSTERED INDEX [IDX\_­Blood­Test­Results­Sample­Date­Lead­Value] ON [dbo].[Blood­Test­Results] ([Person­ID], [Sample­Date], [Lead­Value]) INCLUDE ([Blood­Test­Results­ID], [Client­Status­ID], [Lab­Submission­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Target­Status] FOREIGN KEY ([Client­Status­ID]) REFERENCES [dbo].[Target­Status] ([Status­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Hematocrit­Level­Category] FOREIGN KEY ([Hematocrit­Value­Category­ID]) REFERENCES [dbo].[Sample­Level­Category] ([Sample­Level­Category­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Hemoglobin­Level­Category] FOREIGN KEY ([Hemoglobin­Value­Category­ID]) REFERENCES [dbo].[Sample­Level­Category] ([Sample­Level­Category­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Lab] FOREIGN KEY ([Lab­ID]) REFERENCES [dbo].[Lab] ([Lab­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Lead­Level­Category] FOREIGN KEY ([Lead­Value­Category­ID]) REFERENCES [dbo].[Sample­Level­Category] ([Sample­Level­Category­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Blood­Test­Results] ADD CONSTRAINT [FK\_­Blood­Test­Results\_­Sample­Type] FOREIGN KEY ([Sample­Type­ID]) REFERENCES [dbo].[Sample­Type] ([Sample­Type­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of blood test result values and categorization', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'cost of the blood tests', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Blood­Test­Costs'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the blood test results object', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Blood­Test­Results­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the associated hematocrit value categorization', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Hematocrit­Value­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the associated hemoglobin value categorization', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Hemoglobin­Value­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Historic bloodpbresults id from access database', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Historic­Blood­Test­Results­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic lab results id from access database', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Historic­Lab­Results­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'is­Baseline'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the lab to which the samples were submitted', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Lab­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the sample was submitted to the lab', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Lab­Submission­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the associated lead value categorization', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Lead­Value­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the sample was taken', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Sample­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the type of sample; i.e. venus, capo, soil, water, nitton analyzer . . . ', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Sample­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 - No, 1 - yes; was the blood sample taken after property remediation was completed.', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results', 'COLUMN', N'Taken­After­Property­Remediation­Completed'  GO |

Uses

[[dbo].[Lab]](#(local)/User_databases/LCCHPDev/Tables/Lab)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory)

[[dbo].[Sample­Type]](#(local)/User_databases/LCCHPDev/Tables/SampleType)

[[dbo].[Target­Status]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResultsNotes)

[[dbo].[v­Most­Recent­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Views/vMostRecentBloodTestResults)

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_­SLAll­Blood­Test­Results­Meta­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResultsMetaData)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

[[dbo].[usp\_up­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults)

[[dbo].[usp\_up­Blood­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen)

[[dbo].[usp\_up­Client­Flag]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag)

|  |
| --- |
| [dbo].[Blood­Test­Results­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 6 |
| Created | 12:18:19 AM Tuesday, February 17, 2015 |
| Last Modified | 12:18:19 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Blood­Test­Results­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Blood­Test­Results­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Blood­Test­Results­Notes | Blood­Test­Results­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Blood­Test­Results­Notes\_­Blood­Test­Results | Blood­Test­Results­ID->[[dbo].[Blood­Test­Results].[Blood­Test­Results­ID]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Blood­Test­Results­Notes]  (  [Blood­Test­Results­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Blood­Test­Results­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Blood­Test­Results­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Blood­Test­Results­Notes] ADD CONSTRAINT [PK\_­Blood­Test­Results­Notes] PRIMARY KEY CLUSTERED ([Blood­Test­Results­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Blood­Test­Results­Notes] ADD CONSTRAINT [FK\_­Blood­Test­Results­Notes\_­Blood­Test­Results] FOREIGN KEY ([Blood­Test­Results­ID]) REFERENCES [dbo].[Blood­Test­Results] ([Blood­Test­Results­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Blood­Test­Results­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

Used By

[[dbo].[usp\_­Insert­Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes)

|  |
| --- |
| [dbo].[Cleanup­Status] |

MS\_­Description

collection of clean up status

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 14 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 3:07:34 PM Sunday, April 19, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Cleanup­Status­ID unique identifier of the cleanup status object | tinyint | 1 | False | 1 - 1 |  |
|  | Cleanup­Status­Description description of the cleanup status | varchar(253) | 253 | True |  |  |
|  | Cleanup­Status­Name short name for the cleanup status | varchar(50) | 50 | True |  |  |
|  | Historic­Cleanup­Status­ID | char(1) | 1 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Cleanup­Status | Cleanup­Status­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Cleanup­Status | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Cleanup­Status]  (  [Cleanup­Status­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Cleanup­Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Cleanup­Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Cleanup­Status­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Cleanup­Status\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Cleanup­Status] on [dbo].[Cleanup­Status] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Cleanup­Status set Modified­Date = getdate() where Cleanup­Status­ID in (select Cleanup­Status­ID from inserted)  end  GO  ALTER TABLE [dbo].[Cleanup­Status] ADD CONSTRAINT [PK\_­Cleanup­Status] PRIMARY KEY CLUSTERED ([Cleanup­Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of clean up status', 'SCHEMA', N'dbo', 'TABLE', N'Cleanup­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'description of the cleanup status', 'SCHEMA', N'dbo', 'TABLE', N'Cleanup­Status', 'COLUMN', N'Cleanup­Status­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the cleanup status object', 'SCHEMA', N'dbo', 'TABLE', N'Cleanup­Status', 'COLUMN', N'Cleanup­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the cleanup status', 'SCHEMA', N'dbo', 'TABLE', N'Cleanup­Status', 'COLUMN', N'Cleanup­Status­Name'  GO |

Used By

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/PropertytoCleanupStatus)

[[dbo].[usp\_­Insert­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCleanupStatus)

|  |
| --- |
| [dbo].[Condition] |

MS\_­Description

Collection of potential status for Action

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 2:27:18 PM Saturday, April 11, 2015 |
| Last Modified | 2:27:18 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Condition­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Condition­Description Detailed description of the Action status | varchar(253) | 253 | True |  |  |
|  | Condition­Name status for the Action | varchar(50) | 50 | True |  |  |
|  | Historic­Condition­ID historic status from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Condition | Condition­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Condition]  (  [Condition­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Condition­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Condition­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Condition­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Condition\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Condition] ADD CONSTRAINT [PK\_­Condition] PRIMARY KEY CLUSTERED ([Condition­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of potential status for Action', 'SCHEMA', N'dbo', 'TABLE', N'Condition', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the Action status', 'SCHEMA', N'dbo', 'TABLE', N'Condition', 'COLUMN', N'Condition­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'status for the Action', 'SCHEMA', N'dbo', 'TABLE', N'Condition', 'COLUMN', N'Condition­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Condition', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic status from access database', 'SCHEMA', N'dbo', 'TABLE', N'Condition', 'COLUMN', N'Historic­Condition­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Condition', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Construction­Type] |

MS\_­Description

collection of construction types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 10 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:06 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Construction­Type­ID unique identifier of the construction type | tinyint | 1 | False | 1 - 1 |  |
|  | Construction­Type­Name description of the construction type | varchar(50) | 50 | False |  |  |
|  | Construction­Type­Description | varchar(253) | 253 | True |  |  |
|  | Historic­Construction­Type­ID | char(1) | 1 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Construction­Type | Construction­Type­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Construction­Type | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Construction­Type]  (  [Construction­Type­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Construction­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Construction­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Construction­Type­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Construction­Type\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Construction­Type] on [dbo].[Construction­Type] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Construction­Type set Modified­Date = getdate() where Construction­Type­ID in (select Construction­Type­ID from inserted)  end  GO  ALTER TABLE [dbo].[Construction­Type] ADD CONSTRAINT [PK\_­Construction­Type] PRIMARY KEY CLUSTERED ([Construction­Type­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of construction types', 'SCHEMA', N'dbo', 'TABLE', N'Construction­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the construction type', 'SCHEMA', N'dbo', 'TABLE', N'Construction­Type', 'COLUMN', N'Construction­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'description of the construction type', 'SCHEMA', N'dbo', 'TABLE', N'Construction­Type', 'COLUMN', N'Construction­Type­Name'  GO |

Used By

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[usp\_­Insert­Construction­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertConstructionType)

|  |
| --- |
| [dbo].[Contact­Type] |

MS\_­Description

Collection of contact types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 6:17:34 PM Thursday, April 16, 2015 |
| Last Modified | 6:17:34 PM Thursday, April 16, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Contact­Type­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Contact­Type­Description Detailed description of the Action status | varchar(253) | 253 | True |  |  |
|  | Contact­Type­Name status for the Action | varchar(50) | 50 | True |  |  |
|  | Historic­Contact­Type­ID historic status from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Contact­Type | Contact­Type­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Contact­Type]  (  [Contact­Type­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Contact­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Contact­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Contact­Type­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Contact­Type\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Contact­Type] ADD CONSTRAINT [PK\_­Contact­Type] PRIMARY KEY CLUSTERED ([Contact­Type­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of contact types', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the Action status', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', 'COLUMN', N'Contact­Type­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'status for the Action', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', 'COLUMN', N'Contact­Type­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic status from access database', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', 'COLUMN', N'Historic­Contact­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Contact­Type', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Contractor] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:06 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Contractor­ID | int | 4 | False | 1 - 1 |  |
|  | Contractor­Name | varchar(50) | 50 | True |  |  |
|  | Contractor­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Contractor | Contractor­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Contractor | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Contractor]  (  [Contractor­ID] [int] NOT NULL IDENTITY(1, 1),  [Contractor­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Contractor­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Contractor\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Contractor] on [dbo].[Contractor] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Contractor set Modified­Date = getdate() where Contractor­ID in (select Contractor­ID from inserted)  end  GO  ALTER TABLE [dbo].[Contractor] ADD CONSTRAINT [PK\_­Contractor] PRIMARY KEY CLUSTERED ([Contractor­ID]) ON [UData]  GO |

Used By

[[dbo].[Contractorto­Property]](#(local)/User_databases/LCCHPDev/Tables/ContractortoProperty)

[[dbo].[Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediation)

[[dbo].[Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediationActionPlan)

[[dbo].[usp\_­Insert­Contractor]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractor)

|  |
| --- |
| [dbo].[Contractorto­Property] |

MS\_­Description

linking table for contractor and occupied properties

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:04:12 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Contractor­ID | int | 4 | False |  |
|  | Property­ID | int | 4 | False |  |
|  | Start­Date date the contractor started occuping the property | date | 3 | True |  |
|  | End­Date date contractor ended property occupation | date | 3 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Contractorto­Property | Contractor­ID, Property­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Contractorto­Property\_­Contractor building which the contractor occupies for purpose of business (contractor offices) | Contractor­ID->[[dbo].[Contractor].[Contractor­ID]](#(local)/User_databases/LCCHPDev/Tables/Contractor) |
| FK\_­Contractorto­Property\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Contractorto­Property]  (  [Contractor­ID] [int] NOT NULL,  [Property­ID] [int] NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Contractorto­Property\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Property] ADD CONSTRAINT [PK\_­Contractorto­Property] PRIMARY KEY CLUSTERED ([Contractor­ID], [Property­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Property] ADD CONSTRAINT [FK\_­Contractorto­Property\_­Contractor] FOREIGN KEY ([Contractor­ID]) REFERENCES [dbo].[Contractor] ([Contractor­ID])  GO  ALTER TABLE [dbo].[Contractorto­Property] ADD CONSTRAINT [FK\_­Contractorto­Property\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for contractor and occupied properties', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date contractor ended property occupation', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Property', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the contractor started occuping the property', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Property', 'COLUMN', N'Start­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'building which the contractor occupies for purpose of business (contractor offices)', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Property', 'CONSTRAINT', N'FK\_­Contractorto­Property\_­Contractor'  GO |

Uses

[[dbo].[Contractor]](#(local)/User_databases/LCCHPDev/Tables/Contractor)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Contractorto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoProperty)

|  |
| --- |
| [dbo].[Contractorto­Remediation] |

MS\_­Description

linking table for contractors and remediations

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:08:33 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Contractor­ID | int | 4 | False |  |
|  | Remediation­ID | int | 4 | False |  |
|  | Start­Date date the contractor started working on the remidiation | date | 3 | True |  |
|  | End­Date date the contractor stopped working on the remediation | date | 3 | True |  |
|  | is­Sub­Contractor 0 - no, 1 - yes. is this contractor a sub contractor | bit | 1 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Contractorto­Remediation | Contractor­ID, Remediation­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Contractorto­Remediation\_­Contractor | Contractor­ID->[[dbo].[Contractor].[Contractor­ID]](#(local)/User_databases/LCCHPDev/Tables/Contractor) |
| FK\_­Contractorto­Remediation\_­Remediation | Remediation­ID->[[dbo].[Remediation].[Remediation­ID]](#(local)/User_databases/LCCHPDev/Tables/Remediation) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Contractorto­Remediation]  (  [Contractor­ID] [int] NOT NULL,  [Remediation­ID] [int] NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [is­Sub­Contractor] [bit] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Contractorto­Remediation\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Remediation] ADD CONSTRAINT [PK\_­Contractorto­Remediation] PRIMARY KEY CLUSTERED ([Contractor­ID], [Remediation­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Remediation] ADD CONSTRAINT [FK\_­Contractorto­Remediation\_­Contractor] FOREIGN KEY ([Contractor­ID]) REFERENCES [dbo].[Contractor] ([Contractor­ID])  GO  ALTER TABLE [dbo].[Contractorto­Remediation] ADD CONSTRAINT [FK\_­Contractorto­Remediation\_­Remediation] FOREIGN KEY ([Remediation­ID]) REFERENCES [dbo].[Remediation] ([Remediation­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for contractors and remediations', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Remediation', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the contractor stopped working on the remediation', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Remediation', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 - no, 1 - yes. is this contractor a sub contractor', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Remediation', 'COLUMN', N'is­Sub­Contractor'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the contractor started working on the remidiation', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Remediation', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Contractor]](#(local)/User_databases/LCCHPDev/Tables/Contractor)

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

Used By

[[dbo].[usp\_­Insert­Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediation)

|  |
| --- |
| [dbo].[Contractorto­Remediation­Action­Plan] |

MS\_­Description

linking table for contractor and sampling plan

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:08:33 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Contractor­ID | int | 4 | False |  |
|  | Remediation­Action­Plan­ID | int | 4 | False |  |
|  | Start­Date | date | 3 | True |  |
|  | End­Date | date | 3 | True |  |
|  | is­Sub­Contractor | bit | 1 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Contractorto­Remediation­Action­Plan | Contractor­ID, Remediation­Action­Plan­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Contractorto­Remediation­Action­Plan\_­Contractor | Contractor­ID->[[dbo].[Contractor].[Contractor­ID]](#(local)/User_databases/LCCHPDev/Tables/Contractor) |
| FK\_­Contractorto­Sampling­Plan\_­Contractor | Contractor­ID->[[dbo].[Contractor].[Contractor­ID]](#(local)/User_databases/LCCHPDev/Tables/Contractor) |
| FK\_­Contractorto­Remediation­Action­Plan\_­Remediation­Action­Plan | Remediation­Action­Plan­ID->[[dbo].[Remediation­Action­Plan].[Remediation­Action­Plan­ID]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan) |
| FK\_­Contractorto­Remediation­Plan\_­Remediation­Action­Plan | Remediation­Action­Plan­ID->[[dbo].[Remediation­Action­Plan].[Remediation­Action­Plan­ID]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Contractorto­Remediation­Action­Plan]  (  [Contractor­ID] [int] NOT NULL,  [Remediation­Action­Plan­ID] [int] NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [is­Sub­Contractor] [bit] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Contractorto­Remediation­Action­Plan\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Remediation­Action­Plan] ADD CONSTRAINT [PK\_­Contractorto­Remediation­Action­Plan] PRIMARY KEY CLUSTERED ([Contractor­ID], [Remediation­Action­Plan­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Contractorto­Remediation­Action­Plan] ADD CONSTRAINT [FK\_­Contractorto­Remediation­Action­Plan\_­Contractor] FOREIGN KEY ([Contractor­ID]) REFERENCES [dbo].[Contractor] ([Contractor­ID])  GO  ALTER TABLE [dbo].[Contractorto­Remediation­Action­Plan] ADD CONSTRAINT [FK\_­Contractorto­Sampling­Plan\_­Contractor] FOREIGN KEY ([Contractor­ID]) REFERENCES [dbo].[Contractor] ([Contractor­ID])  GO  ALTER TABLE [dbo].[Contractorto­Remediation­Action­Plan] ADD CONSTRAINT [FK\_­Contractorto­Remediation­Action­Plan\_­Remediation­Action­Plan] FOREIGN KEY ([Remediation­Action­Plan­ID]) REFERENCES [dbo].[Remediation­Action­Plan] ([Remediation­Action­Plan­ID])  GO  ALTER TABLE [dbo].[Contractorto­Remediation­Action­Plan] ADD CONSTRAINT [FK\_­Contractorto­Remediation­Plan\_­Remediation­Action­Plan] FOREIGN KEY ([Remediation­Action­Plan­ID]) REFERENCES [dbo].[Remediation­Action­Plan] ([Remediation­Action­Plan­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for contractor and sampling plan', 'SCHEMA', N'dbo', 'TABLE', N'Contractorto­Remediation­Action­Plan', NULL, NULL  GO |

Uses

[[dbo].[Contractor]](#(local)/User_databases/LCCHPDev/Tables/Contractor)

[[dbo].[Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan)

Used By

[[dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediationActionPlan)

|  |
| --- |
| [dbo].[Country] |

MS\_­Description

collection of countries

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 3 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:06 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Country­ID unique identifier of the country | tinyint | 1 | False | 1 - 1 |  |
|  | Country­Name name of the country | varchar(50) | 50 | False |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Country | Country­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Country | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Country]  (  [Country­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Country­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Country\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Country] on [dbo].[Country] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Country set Modified­Date = getdate() where Country­ID in (select Country­ID from inserted)  end  GO  ALTER TABLE [dbo].[Country] ADD CONSTRAINT [PK\_­Country] PRIMARY KEY CLUSTERED ([Country­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of countries', 'SCHEMA', N'dbo', 'TABLE', N'Country', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the country', 'SCHEMA', N'dbo', 'TABLE', N'Country', 'COLUMN', N'Country­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'name of the country', 'SCHEMA', N'dbo', 'TABLE', N'Country', 'COLUMN', N'Country­Name'  GO |

Used By

[[dbo].[Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Tables/ForeignFoodtoCountry)

[[dbo].[Person­To­Travel­Country]](#(local)/User_databases/LCCHPDev/Tables/PersonToTravelCountry)

[[dbo].[usp\_­Insert­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCountry)

|  |
| --- |
| [dbo].[Data­Source] |

MS\_­Description

Collection of contact types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 2:40:10 PM Saturday, April 11, 2015 |
| Last Modified | 2:40:10 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Data­Source­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Data­Source­Description Detailed description of the data source | varchar(253) | 253 | True |  |  |
|  | Data­Source­Name short name for the data source | varchar(50) | 50 | True |  |  |
|  | Historic­Data­Source­ID historic data source ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Data­Source | Data­Source­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Data­Source]  (  [Data­Source­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Data­Source­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Data­Source­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Data­Source­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Data­Source\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Data­Source] ADD CONSTRAINT [PK\_­Data­Source] PRIMARY KEY CLUSTERED ([Data­Source­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of contact types', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the data source', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', 'COLUMN', N'Data­Source­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the data source', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', 'COLUMN', N'Data­Source­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic data source ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', 'COLUMN', N'Historic­Data­Source­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Data­Source', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Daycare] |

MS\_­Description

collection of daycare facilities

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 6 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 11:53:03 PM Monday, April 27, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Daycare­ID | int | 4 | False | 1 - 1 |  |
|  | Daycare­Name name of the daycare | varchar(50) | 50 | False |  |  |
|  | Daycare­Description short description of the daycare business | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Daycare | Daycare­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Daycare | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Daycare]  (  [Daycare­ID] [int] NOT NULL IDENTITY(1, 1),  [Daycare­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Daycare­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Daycare\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Daycare] on [dbo].[Daycare] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Daycare set Modified­Date = getdate() where Daycare­ID in (select Daycare­ID from inserted)  end  GO  ALTER TABLE [dbo].[Daycare] ADD CONSTRAINT [PK\_­Daycare] PRIMARY KEY CLUSTERED ([Daycare­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of daycare facilities', 'SCHEMA', N'dbo', 'TABLE', N'Daycare', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short description of the daycare business', 'SCHEMA', N'dbo', 'TABLE', N'Daycare', 'COLUMN', N'Daycare­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'name of the daycare', 'SCHEMA', N'dbo', 'TABLE', N'Daycare', 'COLUMN', N'Daycare­Name'  GO |

Used By

[[dbo].[Daycareto­Property]](#(local)/User_databases/LCCHPDev/Tables/DaycaretoProperty)

[[dbo].[Personto­Daycare]](#(local)/User_databases/LCCHPDev/Tables/PersontoDaycare)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp\_­Insert­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycare)

[[dbo].[usp\_­Sl­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlDaycare)

|  |
| --- |
| [dbo].[Daycare­Primary­Contact] |

MS\_­Description

linking table for daycare and person - identifying contact person

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 3 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 6:58:11 PM Saturday, April 4, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Daycare­ID | int | 4 | False |  |
|  | Person­ID | int | 4 | False |  |
|  | Contact­Priority priority of this person in the contact list (1 being highest priority) | tinyint | 1 | False | ((1)) |
|  | Primary­Phone­Number­ID id of the primary contact number | int | 4 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Daycare­Contact­Person | Daycare­ID, Person­ID, Contact­Priority | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Daycare­Primary­Contact]  (  [Daycare­ID] [int] NOT NULL,  [Person­ID] [int] NOT NULL,  [Contact­Priority] [tinyint] NOT NULL CONSTRAINT [DF\_­Daycare­Contact­Person\_­Contact­Priority] DEFAULT ((1)),  [Primary­Phone­Number­ID] [int] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Daycare­Primary­Contact\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Daycare­Primary­Contact] ADD CONSTRAINT [PK\_­Daycare­Contact­Person] PRIMARY KEY CLUSTERED ([Daycare­ID], [Person­ID], [Contact­Priority]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for daycare and person - identifying contact person', 'SCHEMA', N'dbo', 'TABLE', N'Daycare­Primary­Contact', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'priority of this person in the contact list (1 being highest priority)', 'SCHEMA', N'dbo', 'TABLE', N'Daycare­Primary­Contact', 'COLUMN', N'Contact­Priority'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the primary contact number', 'SCHEMA', N'dbo', 'TABLE', N'Daycare­Primary­Contact', 'COLUMN', N'Primary­Phone­Number­ID'  GO |

Used By

[[dbo].[usp\_­Insert­Daycare­Primary­Contact]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycarePrimaryContact)

|  |
| --- |
| [dbo].[Daycareto­Property] |

MS\_­Description

linking table for daycare and property

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:08:33 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Daycare­ID | int | 4 | False |  |
|  | Property­ID | int | 4 | False |  |
|  | Start­Date date the daycare started occupying the property | date | 3 | False | (getdate()) |
|  | End­Date date the daycare stopped occupying the property | date | 3 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Daycareto­Property | Daycare­ID, Property­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Daycareto­Property\_­Daycare | Daycare­ID->[[dbo].[Daycare].[Daycare­ID]](#(local)/User_databases/LCCHPDev/Tables/Daycare) |
| FK\_­Daycareto­Property\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Daycareto­Property]  (  [Daycare­ID] [int] NOT NULL,  [Property­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Daycareto­Property\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Daycareto­Property\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Daycareto­Property] ADD CONSTRAINT [PK\_­Daycareto­Property] PRIMARY KEY CLUSTERED ([Daycare­ID], [Property­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Daycareto­Property] ADD CONSTRAINT [FK\_­Daycareto­Property\_­Daycare] FOREIGN KEY ([Daycare­ID]) REFERENCES [dbo].[Daycare] ([Daycare­ID])  GO  ALTER TABLE [dbo].[Daycareto­Property] ADD CONSTRAINT [FK\_­Daycareto­Property\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for daycare and property', 'SCHEMA', N'dbo', 'TABLE', N'Daycareto­Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the daycare stopped occupying the property', 'SCHEMA', N'dbo', 'TABLE', N'Daycareto­Property', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the daycare started occupying the property', 'SCHEMA', N'dbo', 'TABLE', N'Daycareto­Property', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Daycare]](#(local)/User_databases/LCCHPDev/Tables/Daycare)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Daycareto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycaretoProperty)

|  |
| --- |
| [dbo].[Employer] |

MS\_­Description

collection of employers

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 5 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:06 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Employer­ID unique identifier of the employer | int | 4 | False | 1 - 1 |  |
|  | Employer­Name name of the employer | varchar(50) | 50 | False |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Employer | Employer­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Employer | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Employer]  (  [Employer­ID] [int] NOT NULL IDENTITY(1, 1),  [Employer­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Employer\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Employer] on [dbo].[Employer] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Employer set Modified­Date = getdate() where Employer­ID in (select Employer­ID from inserted)  end  GO  ALTER TABLE [dbo].[Employer] ADD CONSTRAINT [PK\_­Employer] PRIMARY KEY CLUSTERED ([Employer­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of employers', 'SCHEMA', N'dbo', 'TABLE', N'Employer', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the employer', 'SCHEMA', N'dbo', 'TABLE', N'Employer', 'COLUMN', N'Employer­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'name of the employer', 'SCHEMA', N'dbo', 'TABLE', N'Employer', 'COLUMN', N'Employer­Name'  GO |

Used By

[[dbo].[Employerto­Property]](#(local)/User_databases/LCCHPDev/Tables/EmployertoProperty)

[[dbo].[Personto­Employer]](#(local)/User_databases/LCCHPDev/Tables/PersontoEmployer)

[[dbo].[usp\_­Insert­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployer)

|  |
| --- |
| [dbo].[Employerto­Property] |

MS\_­Description

linking table for employer and property

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:08:33 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Employer­ID | int | 4 | False |  |
|  | Property­ID | int | 4 | False |  |
|  | Start­Date date the employer started occuppying the property | date | 3 | False | (getdate()) |
|  | End­Date date the employer stopped occuppying the property | date | 3 | True |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Employerto­Property | Employer­ID, Property­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Employerto­Property\_­Employer | Employer­ID->[[dbo].[Employer].[Employer­ID]](#(local)/User_databases/LCCHPDev/Tables/Employer) |
| FK\_­Employerto­Property\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Employerto­Property]  (  [Employer­ID] [int] NOT NULL,  [Property­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Employerto­Property\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Employerto­Property\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Employerto­Property] ADD CONSTRAINT [PK\_­Employerto­Property] PRIMARY KEY CLUSTERED ([Employer­ID], [Property­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Employerto­Property] ADD CONSTRAINT [FK\_­Employerto­Property\_­Employer] FOREIGN KEY ([Employer­ID]) REFERENCES [dbo].[Employer] ([Employer­ID])  GO  ALTER TABLE [dbo].[Employerto­Property] ADD CONSTRAINT [FK\_­Employerto­Property\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for employer and property', 'SCHEMA', N'dbo', 'TABLE', N'Employerto­Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the employer stopped occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Employerto­Property', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the employer started occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Employerto­Property', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Employer]](#(local)/User_databases/LCCHPDev/Tables/Employer)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Employerto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployertoProperty)

|  |
| --- |
| [dbo].[Environmental­Investigation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Environmental­Investigation­ID | int | 4 | False | 1 - 1 |  |
|  | Conduct­Environmental­Investigation 0 - no, 1 - yes; is an environmental investigation going to be conducted | bit | 1 | True |  |  |
|  | Conduct­Environmental­Investigation­Decision­Date date the workgroup decided whether to conduct an environmental investigation or not | date | 3 | True |  |  |
|  | Cost cost of the environmental investigation | money | 8 | True |  |  |
|  | Environmental­Investigation­Date | date | 3 | True |  |  |
|  | Property­ID | int | 4 | False |  |  |
|  | Start­Date | date | 3 | True |  |  |
|  | End­Date | date | 3 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Environmental­Investigation | Environmental­Investigation­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Environmental­Investigation | True | True | After Update |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Environmental­Investigation\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Environmental­Investigation]  (  [Environmental­Investigation­ID] [int] NOT NULL IDENTITY(1, 1),  [Conduct­Environmental­Investigation] [bit] NULL,  [Conduct­Environmental­Investigation­Decision­Date] [date] NULL,  [Cost] [money] NULL,  [Environmental­Investigation­Date] [date] NULL,  [Property­ID] [int] NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Environmental­Investigation\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Environmental­Investigation] on [dbo].[Environmental­Investigation] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Environmental­Investigation set Modified­Date = getdate() where Environmental­Investigation­ID in (select Environmental­Investigation­ID from inserted)  end  GO  ALTER TABLE [dbo].[Environmental­Investigation] ADD CONSTRAINT [PK\_­Environmental­Investigation] PRIMARY KEY CLUSTERED ([Environmental­Investigation­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Environmental­Investigation] ADD CONSTRAINT [FK\_­Environmental­Investigation\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 - no, 1 - yes; is an environmental investigation going to be conducted', 'SCHEMA', N'dbo', 'TABLE', N'Environmental­Investigation', 'COLUMN', N'Conduct­Environmental­Investigation'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the workgroup decided whether to conduct an environmental investigation or not', 'SCHEMA', N'dbo', 'TABLE', N'Environmental­Investigation', 'COLUMN', N'Conduct­Environmental­Investigation­Decision­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'cost of the environmental investigation', 'SCHEMA', N'dbo', 'TABLE', N'Environmental­Investigation', 'COLUMN', N'Cost'  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan)

[[dbo].[usp\_­Insert­Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEnvironmentalInvestigation)

|  |
| --- |
| [dbo].[Error­Log] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 815 |
| Created | 10:31:53 PM Tuesday, December 23, 2014 |
| Last Modified | 2:08:33 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Error­ID | int | 4 | False | 1 - 1 |  |
|  | Username | nvarchar(128) | 256 | False |  |  |
|  | Error­Number | int | 4 | True |  |  |
|  | Error­Severity | int | 4 | True |  |  |
|  | Error­State | int | 4 | True |  |  |
|  | Error­Procedure | nvarchar(128) | 256 | True |  |  |
|  | Error­Line | int | 4 | True |  |  |
|  | Error­Message | nvarchar(4000) | 8000 | True |  |  |
|  | Error­Time | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Error | Error­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Error­Log]  (  [Error­ID] [int] NOT NULL IDENTITY(1, 1),  [Username] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Error­Number] [int] NULL,  [Error­Severity] [int] NULL,  [Error­State] [int] NULL,  [Error­Procedure] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Error­Line] [int] NULL,  [Error­Message] [nvarchar] (4000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Error­Time] [datetime] NULL CONSTRAINT [DF\_­Error­Log\_­Error­Time] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Error­Log\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Error­Log] ADD CONSTRAINT [PK\_­Error] PRIMARY KEY CLUSTERED ([Error­ID]) ON [UData]  GO |

Used By

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

|  |
| --- |
| [dbo].[Ethnicity] |

MS\_­Description

collection of ethnicities

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 8 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Ethnicity­ID unique identifier of ethnicities | tinyint | 1 | False | 1 - 1 |  |
|  | Ethnicity friendly shortname of ethnicity | varchar(50) | 50 | False |  |  |
|  | Historic­Ethnicity­Code | char(1) | 1 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Ethnicity | Ethnicity­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Ethnicity | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Ethnicity]  (  [Ethnicity­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Ethnicity] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Historic­Ethnicity­Code] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Ethnicity\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Ethnicity] on [dbo].[Ethnicity] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Ethnicity set Modified­Date = getdate() where Ethnicity­ID in (select Ethnicity­ID from inserted)  end  GO  ALTER TABLE [dbo].[Ethnicity] ADD CONSTRAINT [PK\_­Ethnicity] PRIMARY KEY CLUSTERED ([Ethnicity­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of ethnicities', 'SCHEMA', N'dbo', 'TABLE', N'Ethnicity', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'friendly shortname of ethnicity', 'SCHEMA', N'dbo', 'TABLE', N'Ethnicity', 'COLUMN', N'Ethnicity'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of ethnicities', 'SCHEMA', N'dbo', 'TABLE', N'Ethnicity', 'COLUMN', N'Ethnicity­ID'  GO |

Used By

[[dbo].[Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/PersontoEthnicity)

[[dbo].[usp\_­Insert­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEthnicity)

|  |
| --- |
| [dbo].[Family] |

MS\_­Description

collection of families

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2028 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 9:53:25 AM Saturday, April 18, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Family­ID unique identifier for the family object | int | 4 | False | 1 - 1 |  |
|  | Lastname family name | varchar(50) | 50 | False |  |  |
|  | Numberof­Smokers number of smokers in the family | tinyint | 1 | True |  |  |
|  | Primary­Language­ID id of the families primary language; default = 1 (English) | tinyint | 1 | True |  | ((1)) |
|  | Pets | tinyint | 1 | True |  |  |
|  | Petsinandout | bit | 1 | True |  |  |
|  | Historic­Family­ID | smallint | 2 | True |  |  |
|  | Primary­Property­ID | int | 4 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Frequently­Wash­Pets | bit | 1 | True |  |  |
|  | Foreign­Travel does the family travel to foreign countries | bit | 1 | True |  |  |
|  | Review­Status­ID Review status id | tinyint | 1 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Family | Family­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Family | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Family]  (  [Family­ID] [int] NOT NULL IDENTITY(1, 1),  [Lastname] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Numberof­Smokers] [tinyint] NULL,  [Primary­Language­ID] [tinyint] NULL CONSTRAINT [DF\_­Family\_­Primary­Language­ID] DEFAULT ((1)),  [Pets] [tinyint] NULL,  [Petsinandout] [bit] NULL,  [Historic­Family­ID] [smallint] NULL,  [Primary­Property­ID] [int] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Family\_­Created­Date] DEFAULT (getdate()),  [Frequently­Wash­Pets] [bit] NULL,  [Foreign­Travel] [bit] NULL,  [Review­Status­ID] [tinyint] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Family] on [dbo].[Family] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Family set Modified­Date = getdate() where Family­ID in (select Family­ID from inserted)  end  GO  ALTER TABLE [dbo].[Family] ADD CONSTRAINT [PK\_­Family] PRIMARY KEY CLUSTERED ([Family­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of families', 'SCHEMA', N'dbo', 'TABLE', N'Family', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the family object', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Family­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'does the family travel to foreign countries', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Foreign­Travel'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'family name', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Lastname'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'number of smokers in the family', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Numberof­Smokers'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the families primary language; default = 1 (English)', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Primary­Language­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Review status id', 'SCHEMA', N'dbo', 'TABLE', N'Family', 'COLUMN', N'Review­Status­ID'  GO |

Used By

[[dbo].[Family­Notes]](#(local)/User_databases/LCCHPDev/Tables/FamilyNotes)

[[dbo].[Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/FamilytoPhoneNumber)

[[dbo].[Familyto­Property]](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty)

[[dbo].[Personto­Family]](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily)

[[dbo].[Travel­Notes]](#(local)/User_databases/LCCHPDev/Tables/TravelNotes)

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

[[dbo].[usp\_up­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

[[dbo].[udf\_­Sl­Family­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_SlFamilyPhoneNumber)

|  |
| --- |
| [dbo].[Family­Notes] |

MS\_­Description

table for Family notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 127 |
| Created | 11:10:54 AM Saturday, February 14, 2015 |
| Last Modified | 11:10:54 AM Saturday, February 14, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Family­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Family­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Family­Notes | Family­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Family­Notes\_­Family | Family­ID->[[dbo].[Family].[Family­ID]](#(local)/User_databases/LCCHPDev/Tables/Family) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Family­Notes]  (  [Family­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Family­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Family­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Family­Notes] ADD CONSTRAINT [PK\_­Family­Notes] PRIMARY KEY CLUSTERED ([Family­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Family­Notes] ADD CONSTRAINT [FK\_­Family­Notes\_­Family] FOREIGN KEY ([Family­ID]) REFERENCES [dbo].[Family] ([Family­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for Family notes', 'SCHEMA', N'dbo', 'TABLE', N'Family­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Family­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

Used By

[[dbo].[usp\_­Insert­Family­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

|  |
| --- |
| [dbo].[Familyto­Phone­Number] |

MS\_­Description

linking table for Family and phonenumber

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 51 |
| Created | 5:47:40 PM Saturday, April 4, 2015 |
| Last Modified | 7:07:13 PM Saturday, April 4, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Family­ID | int | 4 | False |  |
|  | Phone­Number­ID | int | 4 | False |  |
|  | Number­Priority order which this number should be used to contact the Family (1 being first, 2 being 2nd . . . ) | tinyint | 1 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Familyto­Phone­Number | Family­ID, Phone­Number­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Familyto­Phone­Number\_­Family | Family­ID->[[dbo].[Family].[Family­ID]](#(local)/User_databases/LCCHPDev/Tables/Family) |
| FK\_­Familyto­Phone­Number\_­Phone­Number | Phone­Number­ID->[[dbo].[Phone­Number].[Phone­Number­ID]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Familyto­Phone­Number]  (  [Family­ID] [int] NOT NULL,  [Phone­Number­ID] [int] NOT NULL,  [Number­Priority] [tinyint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Familyto­Phone­Number\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Familyto­Phone­Number] ADD CONSTRAINT [PK\_­Familyto­Phone­Number] PRIMARY KEY CLUSTERED ([Family­ID], [Phone­Number­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Familyto­Phone­Number] ADD CONSTRAINT [FK\_­Familyto­Phone­Number\_­Family] FOREIGN KEY ([Family­ID]) REFERENCES [dbo].[Family] ([Family­ID])  GO  ALTER TABLE [dbo].[Familyto­Phone­Number] ADD CONSTRAINT [FK\_­Familyto­Phone­Number\_­Phone­Number] FOREIGN KEY ([Phone­Number­ID]) REFERENCES [dbo].[Phone­Number] ([Phone­Number­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for Family and phonenumber', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Phone­Number', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'order which this number should be used to contact the Family (1 being first, 2 being 2nd . . . )', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Phone­Number', 'COLUMN', N'Number­Priority'  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber)

Used By

[[dbo].[usp\_­Insert­Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber)

[[dbo].[udf\_­Sl­Family­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_SlFamilyPhoneNumber)

|  |
| --- |
| [dbo].[Familyto­Property] |

MS\_­Description

linking table for Family and property - indicating when a Family occuppied a property

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2024 |
| Created | 5:59:48 PM Saturday, April 11, 2015 |
| Last Modified | 8:17:29 PM Thursday, April 16, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Familyto­Property­ID | int | 4 | False | 1 - 1 |  |
|  | Family­ID Primary family id mainly from legacy system | int | 4 | False |  |  |
|  | Property­ID | int | 4 | False |  |  |
|  | Property­Link­Type­ID | tinyint | 1 | True |  |  |
|  | Review­Status­ID | tinyint | 1 | True |  |  |
|  | Start­Date date the Family started occuppying the property | date | 3 | True |  | (getdate()) |
|  | End­Date date the Family stopped occuppying the property | date | 3 | True |  |  |
|  | is­Primary­Residence | bit | 1 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Familyto­Property | Familyto­Property­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Familyto­Property | True | True | After Update |

Foreign Keys

|  |  |  |
| --- | --- | --- |
| Name | No Check | Columns |
| FK\_­Familyto­Property\_­Family | True | Family­ID->[[dbo].[Family].[Family­ID]](#(local)/User_databases/LCCHPDev/Tables/Family) |
| FK\_­Familyto­Property\_­Property | True | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |
| FK\_­Familyto­Property\_­Property­Link­Type | True | Property­Link­Type­ID->[[dbo].[Property­Link­Type].[Property­Link­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/PropertyLinkType) |
| FK\_­Familyto­Property\_­Review­Status | True | Review­Status­ID->[[dbo].[Review­Status].[Review­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Familyto­Property]  (  [Familyto­Property­ID] [int] NOT NULL IDENTITY(1, 1),  [Family­ID] [int] NOT NULL,  [Property­ID] [int] NOT NULL,  [Property­Link­Type­ID] [tinyint] NULL,  [Review­Status­ID] [tinyint] NULL,  [Start­Date] [date] NULL CONSTRAINT [DF\_­Familyto­Property\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [is­Primary­Residence] [bit] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Familyto­Property\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Familyto­Property] on [dbo].[Familyto­Property] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Familyto­Property set Modified­Date = getdate() where Familyto­Property­ID in (select Familyto­Property­ID from inserted)  end  GO  ALTER TABLE [dbo].[Familyto­Property] ADD CONSTRAINT [PK\_­Familyto­Property] PRIMARY KEY CLUSTERED ([Familyto­Property­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Familyto­Property] WITH NOCHECK ADD CONSTRAINT [FK\_­Familyto­Property\_­Family] FOREIGN KEY ([Family­ID]) REFERENCES [dbo].[Family] ([Family­ID])  GO  ALTER TABLE [dbo].[Familyto­Property] WITH NOCHECK ADD CONSTRAINT [FK\_­Familyto­Property\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  ALTER TABLE [dbo].[Familyto­Property] WITH NOCHECK ADD CONSTRAINT [FK\_­Familyto­Property\_­Property­Link­Type] FOREIGN KEY ([Property­Link­Type­ID]) REFERENCES [dbo].[Property­Link­Type] ([Property­Link­Type­ID])  GO  ALTER TABLE [dbo].[Familyto­Property] WITH NOCHECK ADD CONSTRAINT [FK\_­Familyto­Property\_­Review­Status] FOREIGN KEY ([Review­Status­ID]) REFERENCES [dbo].[Review­Status] ([Review­Status­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for Family and property - indicating when a Family occuppied a property', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the Family stopped occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Property', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary family id mainly from legacy system', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Property', 'COLUMN', N'Family­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the Family started occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Familyto­Property', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[Property­Link­Type]](#(local)/User_databases/LCCHPDev/Tables/PropertyLinkType)

[[dbo].[Review­Status]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus)

Used By

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

|  |
| --- |
| [dbo].[File­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 6:03:29 PM Friday, January 2, 2015 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | File­Type­ID | smallint | 2 | False | 1 - 1 |  |
|  | File­Type­Name | varchar(50) | 50 | False |  |  |
|  | File­Type­Description | varchar(253) | 253 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­File­Types | File­Type­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­File­Type | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[File­Type]  (  [File­Type­ID] [smallint] NOT NULL IDENTITY(1, 1),  [File­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [File­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­File­Type\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­File­Type] on [dbo].[File­Type] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update File­Type set Modified­Date = getdate() where File­Type­ID in (select File­Type­ID from inserted)  end  GO  ALTER TABLE [dbo].[File­Type] ADD CONSTRAINT [PK\_­File­Types] PRIMARY KEY CLUSTERED ([File­Type­ID]) ON [UData]  GO |

|  |
| --- |
| [dbo].[Flag] |

MS\_­Description

Collection of flag information

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 3:00:25 PM Saturday, April 11, 2015 |
| Last Modified | 3:00:25 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Flag­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Flag­Description Detailed description of the flag | varchar(253) | 253 | True |  |  |
|  | Flag­Name short name for the flag | varchar(50) | 50 | True |  |  |
|  | Historic­Flag­ID historic flg ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Flag | Flag­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Flag]  (  [Flag­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Flag­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Flag­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Flag­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Flag\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Flag] ADD CONSTRAINT [PK\_­Flag] PRIMARY KEY CLUSTERED ([Flag­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of flag information', 'SCHEMA', N'dbo', 'TABLE', N'Flag', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Flag', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the flag', 'SCHEMA', N'dbo', 'TABLE', N'Flag', 'COLUMN', N'Flag­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the flag', 'SCHEMA', N'dbo', 'TABLE', N'Flag', 'COLUMN', N'Flag­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic flg ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Flag', 'COLUMN', N'Historic­Flag­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Flag', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Foreign­Food] |

MS\_­Description

collection of various foreign foods

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Foreign­Food­ID | int | 4 | False | 1 - 1 |  |
|  | Foreign­Food­Name | varchar(50) | 50 | True |  |  |
|  | Foreign­Food­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Foreign­Food | Foreign­Food­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Foreign­Food | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Foreign­Food]  (  [Foreign­Food­ID] [int] NOT NULL IDENTITY(1, 1),  [Foreign­Food­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Foreign­Food­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Foreign­Food\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Foreign­Food] on [dbo].[Foreign­Food] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Foreign­Food set Modified­Date = getdate() where Foreign­Food­ID in (select Foreign­Food­ID from inserted)  end  GO  ALTER TABLE [dbo].[Foreign­Food] ADD CONSTRAINT [PK\_­Foreign­Food] PRIMARY KEY CLUSTERED ([Foreign­Food­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of various foreign foods', 'SCHEMA', N'dbo', 'TABLE', N'Foreign­Food', NULL, NULL  GO |

Used By

[[dbo].[Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Tables/ForeignFoodtoCountry)

[[dbo].[Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/PersontoForeignFood)

[[dbo].[usp\_­Insert­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFood)

|  |
| --- |
| [dbo].[Foreign­Foodto­Country] |

MS\_­Description

foreign food and country linking table

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:10:04 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Foreign­Food­ID | int | 4 | False |  |
|  | Country­ID | tinyint | 1 | False |  |
|  | Modified­Date | datetime | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Foreign­Foodto­Country | Foreign­Food­ID, Country­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Foreign­Foodto­Country\_­Country | Country­ID->[[dbo].[Country].[Country­ID]](#(local)/User_databases/LCCHPDev/Tables/Country) |
| FK\_­Foreign­Foodto­Country\_­Foreign­Food | Foreign­Food­ID->[[dbo].[Foreign­Food].[Foreign­Food­ID]](#(local)/User_databases/LCCHPDev/Tables/ForeignFood) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Foreign­Foodto­Country]  (  [Foreign­Food­ID] [int] NOT NULL,  [Country­ID] [tinyint] NOT NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Foreign­Foodto­Country\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Foreign­Foodto­Country] ADD CONSTRAINT [PK\_­Foreign­Foodto­Country] PRIMARY KEY CLUSTERED ([Foreign­Food­ID], [Country­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Foreign­Foodto­Country] ADD CONSTRAINT [FK\_­Foreign­Foodto­Country\_­Country] FOREIGN KEY ([Country­ID]) REFERENCES [dbo].[Country] ([Country­ID])  GO  ALTER TABLE [dbo].[Foreign­Foodto­Country] ADD CONSTRAINT [FK\_­Foreign­Foodto­Country\_­Foreign­Food] FOREIGN KEY ([Foreign­Food­ID]) REFERENCES [dbo].[Foreign­Food] ([Foreign­Food­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'foreign food and country linking table', 'SCHEMA', N'dbo', 'TABLE', N'Foreign­Foodto­Country', NULL, NULL  GO |

Uses

[[dbo].[Country]](#(local)/User_databases/LCCHPDev/Tables/Country)

[[dbo].[Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/ForeignFood)

Used By

[[dbo].[usp\_­Insert­Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFoodtoCountry)

|  |
| --- |
| [dbo].[Frequency] |

MS\_­Description

Collection of frequencies

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 3:44:11 PM Saturday, April 11, 2015 |
| Last Modified | 3:44:11 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Frequency­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Frequency­Description Detailed description of the frequency type | varchar(253) | 253 | True |  |  |
|  | Frequency­Name short name for the frequency type | varchar(50) | 50 | True |  |  |
|  | Historic­Frequency­ID historic frequency ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Frequency | Frequency­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Frequency]  (  [Frequency­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Frequency­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Frequency­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Frequency­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Frequency\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Frequency] ADD CONSTRAINT [PK\_­Frequency] PRIMARY KEY CLUSTERED ([Frequency­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of frequencies', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the frequency type', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', 'COLUMN', N'Frequency­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the frequency type', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', 'COLUMN', N'Frequency­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic frequency ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', 'COLUMN', N'Historic­Frequency­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Frequency', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Gift­Card] |

MS\_­Description

collection of gift certificate objects

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 8 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Gift­Card­ID unique identifier for the gift certificate | int | 4 | False | 1 - 1 |  |
|  | Gift­Card­Value value of the gift certificate | money | 8 | False |  | ((25)) |
|  | Issue­Date | date | 3 | False |  | (getdate()) |
|  | Person­ID | int | 4 | False |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Gift­Certificate | Gift­Card­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Gift­Card | True | True | After Update |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Gift­Card\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Gift­Card]  (  [Gift­Card­ID] [int] NOT NULL IDENTITY(1, 1),  [Gift­Card­Value] [money] NOT NULL CONSTRAINT [DF\_­Gift­Certificate\_­Gift­Certificate­Value] DEFAULT ((25)),  [Issue­Date] [date] NOT NULL CONSTRAINT [DF\_­Gift­Card\_­Issue­Date] DEFAULT (getdate()),  [Person­ID] [int] NOT NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Gift­Card\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Gift­Card] on [dbo].[Gift­Card] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Gift­Card set Modified­Date = getdate() where Gift­Card­ID in (select Gift­Card­ID from inserted)  end  GO  ALTER TABLE [dbo].[Gift­Card] ADD CONSTRAINT [PK\_­Gift­Certificate] PRIMARY KEY CLUSTERED ([Gift­Card­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Gift­Card] ADD CONSTRAINT [FK\_­Gift­Card\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of gift certificate objects', 'SCHEMA', N'dbo', 'TABLE', N'Gift­Card', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the gift certificate', 'SCHEMA', N'dbo', 'TABLE', N'Gift­Card', 'COLUMN', N'Gift­Card­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'value of the gift certificate', 'SCHEMA', N'dbo', 'TABLE', N'Gift­Card', 'COLUMN', N'Gift­Card­Value'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Gift­Card]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertGiftCard)

|  |
| --- |
| [dbo].[Historic­Contribution] |

MS\_­Description

Collection of historic contribution classifications

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 4:28:26 PM Saturday, April 11, 2015 |
| Last Modified | 4:28:26 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Historic­Contribution­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Historic­Contribution­Description Detailed description of the historic contribution | varchar(253) | 253 | True |  |  |
|  | Historic­Contribution­Name short name for the historic contribution name | varchar(50) | 50 | True |  |  |
|  | Historic­Historic­Contribution­ID historic historic contribution ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Historic­Contribution | Historic­Contribution­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Historic­Contribution]  (  [Historic­Contribution­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Historic­Contribution­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Contribution­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Historic­Contribution­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Historic­Contribution\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Historic­Contribution] ADD CONSTRAINT [PK\_­Historic­Contribution] PRIMARY KEY CLUSTERED ([Historic­Contribution­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of historic contribution classifications', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the historic contribution', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', 'COLUMN', N'Historic­Contribution­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the historic contribution name', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', 'COLUMN', N'Historic­Contribution­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic historic contribution ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', 'COLUMN', N'Historic­Historic­Contribution­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Historic­Contribution', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Hobby] |

MS\_­Description

collection of hobbies

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 22 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:26:09 PM Thursday, February 12, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Hobby­ID unique identifier of hobby objects | smallint | 2 | False | 1 - 1 |  |
|  | Hobby­Description short description of the hobby | varchar(253) | 253 | True |  |  |
|  | Hobby­Name | varchar(50) | 50 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Lead­Exposure | bit | 1 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Hobby | Hobby­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Hobby | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Hobby]  (  [Hobby­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Hobby­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Hobby­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Hobby\_­Created­Date] DEFAULT (getdate()),  [Lead­Exposure] [bit] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Hobby] on [dbo].[Hobby] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Hobby set Modified­Date = getdate() where Hobby­ID in (select Hobby­ID from inserted)  end  GO  ALTER TABLE [dbo].[Hobby] ADD CONSTRAINT [PK\_­Hobby] PRIMARY KEY CLUSTERED ([Hobby­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of hobbies', 'SCHEMA', N'dbo', 'TABLE', N'Hobby', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short description of the hobby', 'SCHEMA', N'dbo', 'TABLE', N'Hobby', 'COLUMN', N'Hobby­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of hobby objects', 'SCHEMA', N'dbo', 'TABLE', N'Hobby', 'COLUMN', N'Hobby­ID'  GO |

Used By

[[dbo].[Personto­Hobby]](#(local)/User_databases/LCCHPDev/Tables/PersontoHobby)

[[dbo].[usp\_­Insert­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHobby)

[[dbo].[usp\_­Sl­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlHobby)

|  |
| --- |
| [dbo].[Home­Remedy] |

MS\_­Description

collection of home remedies

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 3 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Home­Remedy­ID | int | 4 | False | 1 - 1 |  |
|  | Home­Remedy­Name | varchar(50) | 50 | False |  |  |
|  | Home­Remedy­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Home­Remedies | Home­Remedy­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Home­Remedy | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Home­Remedy]  (  [Home­Remedy­ID] [int] NOT NULL IDENTITY(1, 1),  [Home­Remedy­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Home­Remedy­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Home­Remedy\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Home­Remedy] on [dbo].[Home­Remedy] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Home­Remedy set Modified­Date = getdate() where Home­Remedy­ID in (select Home­Remedy­ID from inserted)  end  GO  ALTER TABLE [dbo].[Home­Remedy] ADD CONSTRAINT [PK\_­Home­Remedies] PRIMARY KEY CLUSTERED ([Home­Remedy­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of home remedies', 'SCHEMA', N'dbo', 'TABLE', N'Home­Remedy', NULL, NULL  GO |

Used By

[[dbo].[Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Tables/PersontoHomeRemedy)

[[dbo].[usp\_­Insert­Home­Remedies]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHomeRemedies)

|  |
| --- |
| [dbo].[Household­Sourcesof­Lead] |

MS\_­Description

household items that may contribute to EBL

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Household­Sourcesof­Lead­ID | int | 4 | False | 1 - 1 |  |
|  | Household­Item­Name | varchar(50) | 50 | True |  |  |
|  | Household­Item­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Household­Sourcesof­Lead | Household­Sourcesof­Lead­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Household­Sourcesof­Lead | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Household­Sourcesof­Lead]  (  [Household­Sourcesof­Lead­ID] [int] NOT NULL IDENTITY(1, 1),  [Household­Item­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Household­Item­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Household­Sourcesof­Lead\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Household­Sourcesof­Lead] on [dbo].[Household­Sourcesof­Lead] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Household­Sourcesof­Lead set Modified­Date = getdate() where Household­Sourcesof­Lead­ID in (select Household­Sourcesof­Lead­ID from inserted)  end  GO  ALTER TABLE [dbo].[Household­Sourcesof­Lead] ADD CONSTRAINT [PK\_­Household­Sourcesof­Lead] PRIMARY KEY CLUSTERED ([Household­Sourcesof­Lead­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'household items that may contribute to EBL', 'SCHEMA', N'dbo', 'TABLE', N'Household­Sourcesof­Lead', NULL, NULL  GO |

Used By

[[dbo].[Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Tables/PropertytoHouseholdSourcesofLead)

[[dbo].[usp\_­Insert­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHouseholdSourcesofLead)

|  |
| --- |
| [dbo].[Insurance­Provider] |

MS\_­Description

collection of insurance companies

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 6 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Insurance­Provider­ID unique identifier for insurance company | smallint | 2 | False | 1 - 1 |  |
|  | Insurance­Provider­Name name of the insurance company | varchar(50) | 50 | False |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Insurance­Provider | Insurance­Provider­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Insurance­Provider | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Insurance­Provider]  (  [Insurance­Provider­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Insurance­Provider­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Insurance­Provider\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Insurance­Provider] on [dbo].[Insurance­Provider] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Insurance­Provider set Modified­Date = getdate() where Insurance­Provider­ID in (select Insurance­Provider­ID from inserted)  end  GO  ALTER TABLE [dbo].[Insurance­Provider] ADD CONSTRAINT [PK\_­Insurance­Provider] PRIMARY KEY CLUSTERED ([Insurance­Provider­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of insurance companies', 'SCHEMA', N'dbo', 'TABLE', N'Insurance­Provider', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for insurance company', 'SCHEMA', N'dbo', 'TABLE', N'Insurance­Provider', 'COLUMN', N'Insurance­Provider­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'name of the insurance company', 'SCHEMA', N'dbo', 'TABLE', N'Insurance­Provider', 'COLUMN', N'Insurance­Provider­Name'  GO |

Used By

[[dbo].[Personto­Insurance]](#(local)/User_databases/LCCHPDev/Tables/PersontoInsurance)

[[dbo].[usp\_­Insert­Insurance­Provider]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertInsuranceProvider)

|  |
| --- |
| [dbo].[Lab] |

MS\_­Description

collection of lab names and basic attributes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 10 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 8:24:48 PM Wednesday, March 4, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Lab­ID unique identifier for the lab object | int | 4 | False | 1 - 1 |  |
|  | Lab­Name | varchar(50) | 50 | True |  |  |
|  | Lab­Description | varchar(253) | 253 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Lab | Lab­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Lab | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Lab]  (  [Lab­ID] [int] NOT NULL IDENTITY(1, 1),  [Lab­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Lab­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Lab\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Lab] on [dbo].[Lab] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Lab set Modified­Date = getdate() where Lab­ID in (select Lab­ID from inserted)  end  GO  ALTER TABLE [dbo].[Lab] ADD CONSTRAINT [PK\_­Lab] PRIMARY KEY CLUSTERED ([Lab­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of lab names and basic attributes', 'SCHEMA', N'dbo', 'TABLE', N'Lab', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the lab object', 'SCHEMA', N'dbo', 'TABLE', N'Lab', 'COLUMN', N'Lab­ID'  GO |

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Lab­Notes]](#(local)/User_databases/LCCHPDev/Tables/LabNotes)

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[usp\_­Insert­Lab]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLab)

[[dbo].[usp\_­Sl­Lab­Name]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlLabName)

|  |
| --- |
| [dbo].[Lab­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 12:21:25 AM Tuesday, February 17, 2015 |
| Last Modified | 12:21:25 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Lab­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Lab­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Lab­Notes | Lab­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Lab­Notes\_­Lab | Lab­ID->[[dbo].[Lab].[Lab­ID]](#(local)/User_databases/LCCHPDev/Tables/Lab) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Lab­Notes]  (  [Lab­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Lab­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Lab­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Lab­Notes] ADD CONSTRAINT [PK\_­Lab­Notes] PRIMARY KEY CLUSTERED ([Lab­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Lab­Notes] ADD CONSTRAINT [FK\_­Lab­Notes\_­Lab] FOREIGN KEY ([Lab­ID]) REFERENCES [dbo].[Lab] ([Lab­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Lab­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Lab­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Lab]](#(local)/User_databases/LCCHPDev/Tables/Lab)

Used By

[[dbo].[usp\_­Insert­Lab­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLabNotes)

|  |
| --- |
| [dbo].[Language] |

MS\_­Description

collection of spoken languages

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 26 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 6:08:22 PM Thursday, April 16, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Language­ID unique identifier of langauge object | tinyint | 1 | False | 1 - 1 |  |
|  | Language­Name spoken language | varchar(50) | 50 | False |  |  |
|  | Historic­Primary­Language­Code | char(1) | 1 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Language | Language­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Language | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Language]  (  [Language­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Language­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Historic­Primary­Language­Code] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Language\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Language] on [dbo].[Language] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Language set Modified­Date = getdate() where Language­ID in (select Language­ID from inserted)  end  GO  ALTER TABLE [dbo].[Language] ADD CONSTRAINT [PK\_­Language] PRIMARY KEY CLUSTERED ([Language­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of spoken languages', 'SCHEMA', N'dbo', 'TABLE', N'Language', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of langauge object', 'SCHEMA', N'dbo', 'TABLE', N'Language', 'COLUMN', N'Language­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'spoken language', 'SCHEMA', N'dbo', 'TABLE', N'Language', 'COLUMN', N'Language­Name'  GO |

Used By

[[dbo].[Personto­Language]](#(local)/User_databases/LCCHPDev/Tables/PersontoLanguage)

[[dbo].[usp\_­Insert­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLanguage)

|  |
| --- |
| [dbo].[LCCHPAttachments] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Filestream File Group | LCCHPAttachments |
| Heap | True |
| Row Count (~) | 4 |
| Created | 8:36:33 PM Friday, January 2, 2015 |
| Last Modified | 8:36:33 PM Friday, January 2, 2015 |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[LCCHPAttachments] AS FILETABLE ON [UData] FILESTREAM\_­ON [LCCHPAttachments]  WITH  (  FILETABLE\_­DIRECTORY = N'LCCHPAttachments­Dev', FILETABLE\_­COLLATE\_­FILENAME = SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS  )  GO |

|  |
| --- |
| [dbo].[Medium] |

MS\_­Description

collection of mediums that are tested

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 6 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 4:55:07 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Medium­ID unique identifier of the medium | int | 4 | False | 1 - 1 |  |
|  | Medium­Name short name for the medium | varchar(50) | 50 | False |  |  |
|  | Medium­Description short description of the medium | varchar(253) | 253 | True |  |  |
|  | Trigger­Level | int | 4 | True |  |  |
|  | Trigger­Level­Units­ID | int | 4 | True |  |  |
|  | Historic­Medium­Code mediumcode identifier from legacy database | char(1) | 1 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Medium | Medium­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Medium | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Medium]  (  [Medium­ID] [int] NOT NULL IDENTITY(1, 1),  [Medium­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Medium­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Trigger­Level] [int] NULL,  [Trigger­Level­Units­ID] [int] NULL,  [Historic­Medium­Code] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Medium\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Medium] on [dbo].[Medium] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Medium set Modified­Date = getdate() where Medium­ID in (select Medium­ID from inserted)  end  GO  ALTER TABLE [dbo].[Medium] ADD CONSTRAINT [PK\_­Medium] PRIMARY KEY CLUSTERED ([Medium­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of mediums that are tested', 'SCHEMA', N'dbo', 'TABLE', N'Medium', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'mediumcode identifier from legacy database', 'SCHEMA', N'dbo', 'TABLE', N'Medium', 'COLUMN', N'Historic­Medium­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short description of the medium', 'SCHEMA', N'dbo', 'TABLE', N'Medium', 'COLUMN', N'Medium­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the medium', 'SCHEMA', N'dbo', 'TABLE', N'Medium', 'COLUMN', N'Medium­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the medium', 'SCHEMA', N'dbo', 'TABLE', N'Medium', 'COLUMN', N'Medium­Name'  GO |

Used By

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Tables/PropertytoMedium)

[[dbo].[usp\_­Insert­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMedium)

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

|  |
| --- |
| [dbo].[Medium­Sample­Results] |

MS\_­Description

collection of test results for various medums

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 12 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 7:00:52 PM Thursday, April 16, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Medium­Sample­Results­ID | int | 4 | False | 1 - 1 |  |
|  | Medium­ID tested medium id | int | 4 | False |  |  |
|  | Medium­Sample­Value value of the test result for the medium | numeric(9,4) | 5 | True |  |  |
|  | Sample­Level­Category­ID sample level category | tinyint | 1 | True |  |  |
|  | Medium­Sample­Date date the medium was tested | date | 3 | False |  | (getdate()) |
|  | Lab­ID id of the lab to which the sample was submitted | int | 4 | True |  |  |
|  | Lab­Submission­Date date the sample was submitted to the lab | date | 3 | True |  |  |
|  | Is­Above­Trigger­Level | bit | 1 | True |  |  |
|  | Units­ID | smallint | 2 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Medium­Test­Results | Medium­Sample­Results­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Medium­Sample­Results | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Medium­Sample­Results\_­Medium­Sample­Date | Medium­Sample­Date | ([dbo].[udf\_­Date­In­The­Past]([Medium­Sample­Date])=(1)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Medium­Sample­Results\_­Lab | Lab­ID->[[dbo].[Lab].[Lab­ID]](#(local)/User_databases/LCCHPDev/Tables/Lab) |
| FK\_­Medium­Sample­Results\_­Medium | Medium­ID->[[dbo].[Medium].[Medium­ID]](#(local)/User_databases/LCCHPDev/Tables/Medium) |
| FK\_­Medium­Sample­Results\_­Sample­Level­Category | Sample­Level­Category­ID->[[dbo].[Sample­Level­Category].[Sample­Level­Category­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory) |
| FK\_­Medium­Sample­Results\_­Units | Units­ID->[[dbo].[Units].[Units­ID]](#(local)/User_databases/LCCHPDev/Tables/Units) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Medium­Sample­Results]  (  [Medium­Sample­Results­ID] [int] NOT NULL IDENTITY(1, 1),  [Medium­ID] [int] NOT NULL,  [Medium­Sample­Value] [numeric] (9, 4) NULL,  [Sample­Level­Category­ID] [tinyint] NULL,  [Medium­Sample­Date] [date] NOT NULL CONSTRAINT [DF\_­Medium­Test­Results\_­Medium­Test­Date] DEFAULT (getdate()),  [Lab­ID] [int] NULL,  [Lab­Submission­Date] [date] NULL,  [Is­Above­Trigger­Level] [bit] NULL,  [Units­ID] [smallint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Medium­Sample­Results\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Medium­Sample­Results] on [dbo].[Medium­Sample­Results] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Medium­Sample­Results set Modified­Date = getdate() where Medium­Sample­Results­ID in (select Medium­Sample­Results­ID from inserted)  end  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [ck\_­Medium­Sample­Results\_­Medium­Sample­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Medium­Sample­Date])=(1)))  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [PK\_­Medium­Test­Results] PRIMARY KEY CLUSTERED ([Medium­Sample­Results­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [FK\_­Medium­Sample­Results\_­Lab] FOREIGN KEY ([Lab­ID]) REFERENCES [dbo].[Lab] ([Lab­ID])  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [FK\_­Medium­Sample­Results\_­Medium] FOREIGN KEY ([Medium­ID]) REFERENCES [dbo].[Medium] ([Medium­ID])  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [FK\_­Medium­Sample­Results\_­Sample­Level­Category] FOREIGN KEY ([Sample­Level­Category­ID]) REFERENCES [dbo].[Sample­Level­Category] ([Sample­Level­Category­ID])  GO  ALTER TABLE [dbo].[Medium­Sample­Results] ADD CONSTRAINT [FK\_­Medium­Sample­Results\_­Units] FOREIGN KEY ([Units­ID]) REFERENCES [dbo].[Units] ([Units­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of test results for various medums', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the lab to which the sample was submitted', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Lab­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the sample was submitted to the lab', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Lab­Submission­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'tested medium id', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Medium­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the medium was tested', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Medium­Sample­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'value of the test result for the medium', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Medium­Sample­Value'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'sample level category', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results', 'COLUMN', N'Sample­Level­Category­ID'  GO |

Uses

[[dbo].[Lab]](#(local)/User_databases/LCCHPDev/Tables/Lab)

[[dbo].[Medium]](#(local)/User_databases/LCCHPDev/Tables/Medium)

[[dbo].[Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory)

[[dbo].[Units]](#(local)/User_databases/LCCHPDev/Tables/Units)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResultsNotes)

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

|  |
| --- |
| [dbo].[Medium­Sample­Results­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:00:52 PM Thursday, April 16, 2015 |
| Last Modified | 7:00:52 PM Thursday, April 16, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Medium­Sample­Results­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Medium­Sample­Results­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Medium­Sample­Results­Notes | Medium­Sample­Results­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Medium­Sample­Results­Notes\_­Medium­Sample­Results | Medium­Sample­Results­ID->[[dbo].[Medium­Sample­Results].[Medium­Sample­Results­ID]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Medium­Sample­Results­Notes]  (  [Medium­Sample­Results­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Medium­Sample­Results­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Medium­Sample­Results­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Medium­Sample­Results­Notes] ADD CONSTRAINT [PK\_­Medium­Sample­Results­Notes] PRIMARY KEY CLUSTERED ([Medium­Sample­Results­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Medium­Sample­Results­Notes] ADD CONSTRAINT [FK\_­Medium­Sample­Results­Notes\_­Medium­Sample­Results] FOREIGN KEY ([Medium­Sample­Results­ID]) REFERENCES [dbo].[Medium­Sample­Results] ([Medium­Sample­Results­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Medium­Sample­Results­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

Used By

[[dbo].[usp\_­Insert­Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResultsNotes)

|  |
| --- |
| [dbo].[Method] |

MS\_­Description

Collection of method classifications

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 4:43:56 PM Saturday, April 11, 2015 |
| Last Modified | 4:43:56 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Method­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Method­Description Detailed description of the method | varchar(253) | 253 | True |  |  |
|  | Method­Name short name for the method | varchar(50) | 50 | True |  |  |
|  | Historic­Method­ID historic method ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Method | Method­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Method]  (  [Method­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Method­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Method­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Method­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Method\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Method] ADD CONSTRAINT [PK\_­Method] PRIMARY KEY CLUSTERED ([Method­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of method classifications', 'SCHEMA', N'dbo', 'TABLE', N'Method', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Method', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic method ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Method', 'COLUMN', N'Historic­Method­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the method', 'SCHEMA', N'dbo', 'TABLE', N'Method', 'COLUMN', N'Method­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the method', 'SCHEMA', N'dbo', 'TABLE', N'Method', 'COLUMN', N'Method­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Method', 'COLUMN', N'Modified­Date'  GO |

|  |
| --- |
| [dbo].[Occupation] |

MS\_­Description

collection of occupation objects

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 21 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 1:03:20 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Occupation­ID unique identifier of the occupation | int | 4 | False | 1 - 1 |  |
|  | Occupation­Name name of the occupation | varchar(50) | 50 | False |  |  |
|  | Occupation­Description short description of the occupation | varchar(253) | 253 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Lead­Exposure | bit | 1 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Occupation | Occupation­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Occupation | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Occupation]  (  [Occupation­ID] [int] NOT NULL IDENTITY(1, 1),  [Occupation­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Occupation­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Occupation\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Lead­Exposure] [bit] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Occupation] on [dbo].[Occupation] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Occupation set Modified­Date = getdate() where Occupation­ID in (select Occupation­ID from inserted)  end  GO  ALTER TABLE [dbo].[Occupation] ADD CONSTRAINT [PK\_­Occupation] PRIMARY KEY CLUSTERED ([Occupation­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of occupation objects', 'SCHEMA', N'dbo', 'TABLE', N'Occupation', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short description of the occupation', 'SCHEMA', N'dbo', 'TABLE', N'Occupation', 'COLUMN', N'Occupation­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of the occupation', 'SCHEMA', N'dbo', 'TABLE', N'Occupation', 'COLUMN', N'Occupation­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'name of the occupation', 'SCHEMA', N'dbo', 'TABLE', N'Occupation', 'COLUMN', N'Occupation­Name'  GO |

Used By

[[dbo].[Occupation­Notes]](#(local)/User_databases/LCCHPDev/Tables/OccupationNotes)

[[dbo].[Personto­Occupation]](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation)

[[dbo].[usp\_­Insert­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertOccupation)

|  |
| --- |
| [dbo].[Occupation­Notes] |

MS\_­Description

table for Occupation notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 12:43:48 AM Tuesday, February 17, 2015 |
| Last Modified | 12:54:37 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Occupation­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Occupation­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Occupation­Notes | Occupation­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Occupation­Notes\_­Occupation | Occupation­ID->[[dbo].[Occupation].[Occupation­ID]](#(local)/User_databases/LCCHPDev/Tables/Occupation) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Occupation­Notes]  (  [Occupation­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Occupation­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Occupation­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Occupation­Notes] ADD CONSTRAINT [PK\_­Occupation­Notes] PRIMARY KEY CLUSTERED ([Occupation­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Occupation­Notes] ADD CONSTRAINT [FK\_­Occupation­Notes\_­Occupation] FOREIGN KEY ([Occupation­ID]) REFERENCES [dbo].[Occupation] ([Occupation­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for Occupation notes', 'SCHEMA', N'dbo', 'TABLE', N'Occupation­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Occupation­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Occupation]](#(local)/User_databases/LCCHPDev/Tables/Occupation)

|  |
| --- |
| [dbo].[Person] |

MS\_­Description

collection of people and basic attributes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 3446 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 5:31:42 PM Thursday, June 11, 2015 |

Columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Computed | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­ID unique identifier for each person | int |  | 4 | False | 1 - 1 |  |
|  | First­Name Person's first name | varchar(50) |  | 50 | False |  |  |
|  | Middle­Name Person's middle name or initial | varchar(50) |  | 50 | True |  |  |
|  | Last­Name Person's last name | varchar(50) |  | 50 | False |  |  |
|  | Birth­Date Person's date of birth | date |  | 3 | True |  |  |
|  | Gender Person's gender (i.e. male or femal) | char(1) |  | 1 | True |  |  |
|  | Status­ID | smallint |  | 2 | True |  |  |
|  | Foreign­Travel 0 = No; 1 = Yes; does the person travel to foreign countries | bit |  | 1 | True |  |  |
|  | Outof­Site 0 = No; 1 = Yes; is the person living out of the lead study area. | bit |  | 1 | True |  |  |
|  | Eats­Foreign­Food 0 = No; 1 = Yes; does the person eat foreign candy | bit |  | 1 | True |  |  |
|  | Historic­Child­ID Child­ID from the access db system | smallint |  | 2 | True |  |  |
|  | Retest­Date date for the next scheduled test | date |  | 3 | True |  |  |
|  | Moved 0 = No; 1 = Yes; has the person moved outside the lead study area | bit |  | 1 | True |  |  |
|  | Moved­Date Date the person moved outside of the study area | date |  | 3 | True |  |  |
|  | is­Closed 0 = No; 1 = Yes; is the person's lead study closed | bit |  | 1 | True |  |  |
|  | is­Resolved 0 = No; 1 = Yes; has the lead issue been resolved | bit |  | 1 | True |  |  |
|  | Guardian­ID person­ID of the person's guardian | int |  | 4 | True |  |  |
|  | person­Code | smallint |  | 2 | True |  |  |
|  | is­Smoker 0 = No; 1 = Yes; does the person smoke | bit |  | 1 | True |  |  |
|  | Created­Date Date the record was created | datetime |  | 8 | True |  | (getdate()) |
|  | Modified­Date Date the record was last modified | datetime |  | 8 | True |  |  |
|  | Age Calculated age of the person in years | int | True | 4 | True |  |  |
|  | is­Client 0 = No; 1 = Yes; Is the person a participant in the lead study program | bit |  | 1 | True |  | ((1)) |
|  | Nursing­Mother 0 = No; 1 = Yes; Is the person a nursing mother | bit |  | 1 | True |  |  |
|  | Pregnant 0 = No; 1 = Yes; Is the person pregnant | bit |  | 1 | True |  |  |
|  | Release­Status­ID | tinyint |  | 1 | True |  |  |
|  | Review­Status­ID | tinyint |  | 1 | True |  |  |
|  | Email­Address Person's email address | varchar(320) |  | 320 | True |  |  |
|  | Nursing­Infant 0 = No; 1 = Yes; Is the person a nursing infant | bit |  | 1 | True |  |  |
|  | Client­Status­ID | smallint |  | 2 | True |  |  |

Computed columns

|  |  |
| --- | --- |
| Name | Column definition |
| Age | ([dbo].[udf\_­Calculate­Age]([Birth­Date],getdate())) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person | Person­ID | True | UData |
|  | Non­Clustered­Index-20141220-115023 | Last­Name, Retest­Date |  | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Person | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Person\_­Birth­Date | Birth­Date | ([dbo].[udf\_­Date­In­The­Past]([Birth­Date])=(1)) |
| ck\_­Person\_­Moved­Date | Moved­Date | ([dbo].[udf\_­Date­In­The­Past]([Moved­Date])=(1) OR [Moved­Date] IS NULL) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person\_­Target­Status | Client­Status­ID->[[dbo].[Target­Status].[Status­ID]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus) |
| FK\_­Person\_­Review­Status | Review­Status­ID->[[dbo].[Review­Status].[Review­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person]  (  [Person­ID] [int] NOT NULL IDENTITY(1, 1),  [First­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Middle­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Last­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Birth­Date] [date] NULL,  [Gender] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Status­ID] [smallint] NULL,  [Foreign­Travel] [bit] NULL,  [Outof­Site] [bit] NULL,  [Eats­Foreign­Food] [bit] NULL,  [Historic­Child­ID] [smallint] NULL,  [Retest­Date] [date] NULL,  [Moved] [bit] NULL,  [Moved­Date] [date] NULL,  [is­Closed] [bit] NULL,  [is­Resolved] [bit] NULL,  [Guardian­ID] [int] NULL,  [person­Code] [smallint] NULL,  [is­Smoker] [bit] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Age] AS ([dbo].[udf\_­Calculate­Age]([Birth­Date],getdate())),  [is­Client] [bit] NULL CONSTRAINT [DF\_­Person\_is­Client] DEFAULT ((1)),  [Nursing­Mother] [bit] NULL,  [Pregnant] [bit] NULL,  [Release­Status­ID] [tinyint] NULL,  [Review­Status­ID] [tinyint] NULL,  [Email­Address] [varchar] (320) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Nursing­Infant] [bit] NULL,  [Client­Status­ID] [smallint] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Person] on [dbo].[Person] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Person set Modified­Date = getdate() where Person­ID in (select Person­ID from inserted)  end  GO  ALTER TABLE [dbo].[Person] ADD CONSTRAINT [ck\_­Person\_­Birth­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Birth­Date])=(1)))  GO  ALTER TABLE [dbo].[Person] ADD CONSTRAINT [ck\_­Person\_­Moved­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Moved­Date])=(1) OR [Moved­Date] IS NULL))  GO  ALTER TABLE [dbo].[Person] ADD CONSTRAINT [PK\_­Person] PRIMARY KEY CLUSTERED ([Person­ID]) ON [UData]  GO  CREATE NONCLUSTERED INDEX [Non­Clustered­Index-20141220-115023] ON [dbo].[Person] ([Last­Name], [Retest­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Person] ADD CONSTRAINT [FK\_­Person\_­Target­Status] FOREIGN KEY ([Client­Status­ID]) REFERENCES [dbo].[Target­Status] ([Status­ID])  GO  ALTER TABLE [dbo].[Person] ADD CONSTRAINT [FK\_­Person\_­Review­Status] FOREIGN KEY ([Review­Status­ID]) REFERENCES [dbo].[Review­Status] ([Review­Status­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of people and basic attributes', 'SCHEMA', N'dbo', 'TABLE', N'Person', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Calculated age of the person in years', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Age'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s date of birth', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Birth­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; does the person eat foreign candy', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Eats­Foreign­Food'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s email address', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Email­Address'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s first name', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'First­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; does the person travel to foreign countries', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Foreign­Travel'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s gender (i.e. male or femal)', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Gender'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'person­ID of the person''s guardian', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Guardian­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Child­ID from the access db system', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Historic­Child­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; Is the person a participant in the lead study program', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'is­Client'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; is the person''s lead study closed', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'is­Closed'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; has the lead issue been resolved', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'is­Resolved'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; does the person smoke', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'is­Smoker'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s last name', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Last­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Person''s middle name or initial', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Middle­Name'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the record was last modified', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; has the person moved outside the lead study area', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Moved'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the person moved outside of the study area', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Moved­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; Is the person a nursing infant', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Nursing­Infant'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; Is the person a nursing mother', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Nursing­Mother'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; is the person living out of the lead study area.', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Outof­Site'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for each person', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = No; 1 = Yes; Is the person pregnant', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Pregnant'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date for the next scheduled test', 'SCHEMA', N'dbo', 'TABLE', N'Person', 'COLUMN', N'Retest­Date'  GO |

Uses

[[dbo].[Review­Status]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus)

[[dbo].[Target­Status]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus)

[[dbo].[udf\_­Calculate­Age]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_CalculateAge)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Gift­Card]](#(local)/User_databases/LCCHPDev/Tables/GiftCard)

[[dbo].[Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonHobbyNotes)

[[dbo].[Person­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonNotes)

[[dbo].[Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonReleaseNotes)

[[dbo].[Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/PersontoAccessAgreement)

[[dbo].[Personto­Daycare]](#(local)/User_databases/LCCHPDev/Tables/PersontoDaycare)

[[dbo].[Personto­Employer]](#(local)/User_databases/LCCHPDev/Tables/PersontoEmployer)

[[dbo].[Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/PersontoEthnicity)

[[dbo].[Personto­Family]](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily)

[[dbo].[Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/PersontoForeignFood)

[[dbo].[Personto­Hobby]](#(local)/User_databases/LCCHPDev/Tables/PersontoHobby)

[[dbo].[Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Tables/PersontoHomeRemedy)

[[dbo].[Personto­Insurance]](#(local)/User_databases/LCCHPDev/Tables/PersontoInsurance)

[[dbo].[Personto­Language]](#(local)/User_databases/LCCHPDev/Tables/PersontoLanguage)

[[dbo].[Personto­Occupation]](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation)

[[dbo].[Personto­Person]](#(local)/User_databases/LCCHPDev/Tables/PersontoPerson)

[[dbo].[Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PersontoPhoneNumber)

[[dbo].[Personto­Property]](#(local)/User_databases/LCCHPDev/Tables/PersontoProperty)

[[dbo].[Person­To­Travel­Country]](#(local)/User_databases/LCCHPDev/Tables/PersonToTravelCountry)

[[dbo].[Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonTravelNotes)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[v­Adults]](#(local)/User_databases/LCCHPDev/Views/vAdults)

[[dbo].[v­Most­Recent­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Views/vMostRecentBloodTestResults)

[[dbo].[v­Most­Recent­Questionnaires]](#(local)/User_databases/LCCHPDev/Views/vMostRecentQuestionnaires)

[[dbo].[v­Nursing­Infants]](#(local)/User_databases/LCCHPDev/Views/vNursingInfants)

[[dbo].[v­Nursing­Mothers]](#(local)/User_databases/LCCHPDev/Views/vNursingMothers)

[[dbo].[v­Pregnant]](#(local)/User_databases/LCCHPDev/Views/vPregnant)

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_­Insert­Gift­Card]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertGiftCard)

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_­SLAll­Blood­Test­Results­Meta­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResultsMetaData)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Count­People­By­Age­Group]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByAgeGroup)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

[[dbo].[usp\_up­Client­Flag]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

|  |
| --- |
| [dbo].[Person­Hobby­Notes] |

MS\_­Description

table for person hobby notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 8 |
| Created | 12:58:55 AM Tuesday, April 14, 2015 |
| Last Modified | 12:58:55 AM Tuesday, April 14, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­Hobby­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Person­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­Hobby­Notes | Person­Hobby­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Hobby­Notes\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­Hobby­Notes]  (  [Person­Hobby­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Person­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­Hobby­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­Hobby­Notes] ADD CONSTRAINT [PK\_­Person­Hobby­Notes] PRIMARY KEY CLUSTERED ([Person­Hobby­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Person­Hobby­Notes] ADD CONSTRAINT [FK\_­Person­Hobby­Notes\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for person hobby notes', 'SCHEMA', N'dbo', 'TABLE', N'Person­Hobby­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Person­Hobby­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

|  |
| --- |
| [dbo].[Person­Notes] |

MS\_­Description

table for person notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 125 |
| Created | 10:54:43 AM Saturday, February 14, 2015 |
| Last Modified | 3:18:50 PM Sunday, February 15, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Person­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­Notes | Person­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Notes\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­Notes]  (  [Person­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Person­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­Notes] ADD CONSTRAINT [PK\_­Person­Notes] PRIMARY KEY CLUSTERED ([Person­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Person­Notes] ADD CONSTRAINT [FK\_­Person­Notes\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for person notes', 'SCHEMA', N'dbo', 'TABLE', N'Person­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Person­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes)

|  |
| --- |
| [dbo].[Person­Release­Notes] |

MS\_­Description

table for person release notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 37 |
| Created | 6:55:41 PM Saturday, April 11, 2015 |
| Last Modified | 6:55:41 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­Release­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Person­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­Release­Notes | Person­Release­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Release­Notes\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­Release­Notes]  (  [Person­Release­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Person­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­Release­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­Release­Notes] ADD CONSTRAINT [PK\_­Person­Release­Notes] PRIMARY KEY CLUSTERED ([Person­Release­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Person­Release­Notes] ADD CONSTRAINT [FK\_­Person­Release­Notes\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for person release notes', 'SCHEMA', N'dbo', 'TABLE', N'Person­Release­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Person­Release­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes)

|  |
| --- |
| [dbo].[Person­Status] |

MS\_­Description

Collection of potential status for person

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 1:55:09 PM Saturday, April 11, 2015 |
| Last Modified | 1:55:09 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­Status­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Person­Status­Description Detailed description of the person status | varchar(253) | 253 | True |  |  |
|  | Person­Status­Name status for the person | varchar(50) | 50 | True |  |  |
|  | Historic­Person­Status­ID historic status from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­Status | Person­Status­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­Status]  (  [Person­Status­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Person­Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Person­Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Person­Status­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­Status\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­Status] ADD CONSTRAINT [PK\_­Person­Status] PRIMARY KEY CLUSTERED ([Person­Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of potential status for person', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic status from access database', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', 'COLUMN', N'Historic­Person­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the person status', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', 'COLUMN', N'Person­Status­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'status for the person', 'SCHEMA', N'dbo', 'TABLE', N'Person­Status', 'COLUMN', N'Person­Status­Name'  GO |

|  |
| --- |
| [dbo].[Personto­Access­Agreement] |

MS\_­Description

linking table for person and access agreement

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:10 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Access­Agreement­ID | int | 4 | False |  |
|  | Access­Agreement­Date date the access agreement was signed | date | 3 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Access­Agreement | Person­ID, Access­Agreement­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Access­Agreement\_­Access­Agreement | Access­Agreement­ID->[[dbo].[Access­Agreement].[Access­Agreement­ID]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement) |
| FK\_­Personto­Access­Agreement\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Access­Agreement]  (  [Person­ID] [int] NOT NULL,  [Access­Agreement­ID] [int] NOT NULL,  [Access­Agreement­Date] [date] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Access­Agreement\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Access­Agreement] ADD CONSTRAINT [PK\_­Personto­Access­Agreement] PRIMARY KEY CLUSTERED ([Person­ID], [Access­Agreement­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Access­Agreement] ADD CONSTRAINT [FK\_­Personto­Access­Agreement\_­Access­Agreement] FOREIGN KEY ([Access­Agreement­ID]) REFERENCES [dbo].[Access­Agreement] ([Access­Agreement­ID])  GO  ALTER TABLE [dbo].[Personto­Access­Agreement] ADD CONSTRAINT [FK\_­Personto­Access­Agreement\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and access agreement', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Access­Agreement', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the access agreement was signed', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Access­Agreement', 'COLUMN', N'Access­Agreement­Date'  GO |

Uses

[[dbo].[Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoAccessAgreement)

|  |
| --- |
| [dbo].[Personto­Daycare] |

MS\_­Description

linking table for person and daycare for people attending daycare

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:44 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Daycare­ID | int | 4 | False |  |
|  | Start­Date date the person started attending the daycare | date | 3 | False | (getdate()) |
|  | End­Date date the person stopped attending the daycare | date | 3 | True |  |
|  | Daycare­Notes | varchar(3000) | 3000 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Daycare | Person­ID, Daycare­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Daycare\_­Daycare | Daycare­ID->[[dbo].[Daycare].[Daycare­ID]](#(local)/User_databases/LCCHPDev/Tables/Daycare) |
| FK\_­Personto­Daycare\_­Personto­Daycare | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Daycare]  (  [Person­ID] [int] NOT NULL,  [Daycare­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Personto­Daycare\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Daycare­Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Daycare\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Daycare] ADD CONSTRAINT [PK\_­Personto­Daycare] PRIMARY KEY CLUSTERED ([Person­ID], [Daycare­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Daycare] ADD CONSTRAINT [FK\_­Personto­Daycare\_­Daycare] FOREIGN KEY ([Daycare­ID]) REFERENCES [dbo].[Daycare] ([Daycare­ID])  GO  ALTER TABLE [dbo].[Personto­Daycare] ADD CONSTRAINT [FK\_­Personto­Daycare\_­Personto­Daycare] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and daycare for people attending daycare', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Daycare', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person stopped attending the daycare', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Daycare', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person started attending the daycare', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Daycare', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Daycare]](#(local)/User_databases/LCCHPDev/Tables/Daycare)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoDaycare)

|  |
| --- |
| [dbo].[Personto­Employer] |

MS\_­Description

linking table for person and employer

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Employer­ID | int | 4 | False |  |
|  | Start­Date date the person started working for the employer | date | 3 | False | (getdate()) |
|  | End­Date date the person stopped working for the employer | date | 3 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Employer | Person­ID, Employer­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Employer\_­Employer | Employer­ID->[[dbo].[Employer].[Employer­ID]](#(local)/User_databases/LCCHPDev/Tables/Employer) |
| FK\_­Personto­Employer\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Employer]  (  [Person­ID] [int] NOT NULL,  [Employer­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Personto­Employer\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Employer\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Employer] ADD CONSTRAINT [PK\_­Personto­Employer] PRIMARY KEY CLUSTERED ([Person­ID], [Employer­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Employer] ADD CONSTRAINT [FK\_­Personto­Employer\_­Employer] FOREIGN KEY ([Employer­ID]) REFERENCES [dbo].[Employer] ([Employer­ID])  GO  ALTER TABLE [dbo].[Personto­Employer] ADD CONSTRAINT [FK\_­Personto­Employer\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and employer', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Employer', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person stopped working for the employer', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Employer', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person started working for the employer', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Employer', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Employer]](#(local)/User_databases/LCCHPDev/Tables/Employer)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEmployer)

|  |
| --- |
| [dbo].[Personto­Ethnicity] |

MS\_­Description

linking table for person and ethnicity

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 52 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:36:44 PM Saturday, March 28, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Ethnicity­ID | tinyint | 1 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Ethnicity\_1 | Person­ID, Ethnicity­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Ethnicity\_­Ethnicity | Ethnicity­ID->[[dbo].[Ethnicity].[Ethnicity­ID]](#(local)/User_databases/LCCHPDev/Tables/Ethnicity) |
| FK\_­Personto­Ethnicity\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Ethnicity]  (  [Person­ID] [int] NOT NULL,  [Ethnicity­ID] [tinyint] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Ethnicity\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Ethnicity] ADD CONSTRAINT [PK\_­Personto­Ethnicity\_1] PRIMARY KEY CLUSTERED ([Person­ID], [Ethnicity­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Ethnicity] ADD CONSTRAINT [FK\_­Personto­Ethnicity\_­Ethnicity] FOREIGN KEY ([Ethnicity­ID]) REFERENCES [dbo].[Ethnicity] ([Ethnicity­ID])  GO  ALTER TABLE [dbo].[Personto­Ethnicity] ADD CONSTRAINT [FK\_­Personto­Ethnicity\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and ethnicity', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Ethnicity', NULL, NULL  GO |

Uses

[[dbo].[Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/Ethnicity)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[Personto­Family] |

MS\_­Description

linking table for person and family tables

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 3427 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:18:50 PM Sunday, April 19, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID id of the corresponding person | int | 4 | False |  |
|  | Family­ID id of the corresponding family | int | 4 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |
|  | Start­Date | date | 3 | True |  |
|  | End­Date | date | 3 | True |  |
|  | Primary­Contact­Family | bit | 1 | True |  |
|  | Review­Status­ID | tinyint | 1 | True |  |
|  | Modified­Date | date | 3 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Family | Person­ID, Family­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Family\_­Family | Family­ID->[[dbo].[Family].[Family­ID]](#(local)/User_databases/LCCHPDev/Tables/Family) |
| FK\_­Personto­Family\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Family]  (  [Person­ID] [int] NOT NULL,  [Family­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Family\_­Created­Date] DEFAULT (getdate()),  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [Primary­Contact­Family] [bit] NULL,  [Review­Status­ID] [tinyint] NULL,  [Modified­Date] [date] NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Family] ADD CONSTRAINT [PK\_­Personto­Family] PRIMARY KEY CLUSTERED ([Person­ID], [Family­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Family] ADD CONSTRAINT [FK\_­Personto­Family\_­Family] FOREIGN KEY ([Family­ID]) REFERENCES [dbo].[Family] ([Family­ID])  GO  ALTER TABLE [dbo].[Personto­Family] ADD CONSTRAINT [FK\_­Personto­Family\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and family tables', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Family', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the corresponding family', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Family', 'COLUMN', N'Family­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the corresponding person', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Family', 'COLUMN', N'Person­ID'  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[Personto­Foreign­Food] |

MS\_­Description

linking table for person and foreign food (many to many)

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 3 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Foreign­Food­ID | int | 4 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Foreign­Food | Person­ID, Foreign­Food­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Foreign­Food\_­Foreign­Food | Foreign­Food­ID->[[dbo].[Foreign­Food].[Foreign­Food­ID]](#(local)/User_databases/LCCHPDev/Tables/ForeignFood) |
| FK\_­Personto­Foreign­Food\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Foreign­Food]  (  [Person­ID] [int] NOT NULL,  [Foreign­Food­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Foreign­Food\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Foreign­Food] ADD CONSTRAINT [PK\_­Personto­Foreign­Food] PRIMARY KEY CLUSTERED ([Person­ID], [Foreign­Food­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Foreign­Food] ADD CONSTRAINT [FK\_­Personto­Foreign­Food\_­Foreign­Food] FOREIGN KEY ([Foreign­Food­ID]) REFERENCES [dbo].[Foreign­Food] ([Foreign­Food­ID])  GO  ALTER TABLE [dbo].[Personto­Foreign­Food] ADD CONSTRAINT [FK\_­Personto­Foreign­Food\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and foreign food (many to many)', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Foreign­Food', NULL, NULL  GO |

Uses

[[dbo].[Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/ForeignFood)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoForeignFood)

|  |
| --- |
| [dbo].[Personto­Hobby] |

MS\_­Description

linking table for person and hobby

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 49 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Hobby­ID | smallint | 2 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Hobby | Person­ID, Hobby­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Hobby\_­Hobby | Hobby­ID->[[dbo].[Hobby].[Hobby­ID]](#(local)/User_databases/LCCHPDev/Tables/Hobby) |
| FK\_­Personto­Hobby\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Hobby]  (  [Person­ID] [int] NOT NULL,  [Hobby­ID] [smallint] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Hobby\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Hobby] ADD CONSTRAINT [PK\_­Personto­Hobby] PRIMARY KEY CLUSTERED ([Person­ID], [Hobby­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Hobby] ADD CONSTRAINT [FK\_­Personto­Hobby\_­Hobby] FOREIGN KEY ([Hobby­ID]) REFERENCES [dbo].[Hobby] ([Hobby­ID])  GO  ALTER TABLE [dbo].[Personto­Hobby] ADD CONSTRAINT [FK\_­Personto­Hobby\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and hobby', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Hobby', NULL, NULL  GO |

Uses

[[dbo].[Hobby]](#(local)/User_databases/LCCHPDev/Tables/Hobby)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[Personto­Home­Remedy] |

MS\_­Description

linking table for perosn and home remedy

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Home­Remedy­ID | int | 4 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Home­Remedy | Person­ID, Home­Remedy­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Home­Remedy\_­Personto­Home­Remedy | Home­Remedy­ID->[[dbo].[Home­Remedy].[Home­Remedy­ID]](#(local)/User_databases/LCCHPDev/Tables/HomeRemedy) |
| FK\_­Personto­Home­Remedy\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Home­Remedy]  (  [Person­ID] [int] NOT NULL,  [Home­Remedy­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Home­Remedy\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Home­Remedy] ADD CONSTRAINT [PK\_­Personto­Home­Remedy] PRIMARY KEY CLUSTERED ([Person­ID], [Home­Remedy­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Home­Remedy] ADD CONSTRAINT [FK\_­Personto­Home­Remedy\_­Personto­Home­Remedy] FOREIGN KEY ([Home­Remedy­ID]) REFERENCES [dbo].[Home­Remedy] ([Home­Remedy­ID])  GO  ALTER TABLE [dbo].[Personto­Home­Remedy] ADD CONSTRAINT [FK\_­Personto­Home­Remedy\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for perosn and home remedy', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Home­Remedy', NULL, NULL  GO |

Uses

[[dbo].[Home­Remedy]](#(local)/User_databases/LCCHPDev/Tables/HomeRemedy)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHomeRemedy)

|  |
| --- |
| [dbo].[Personto­Insurance] |

MS\_­Description

linking table for person and insurance

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Insurance­ID | smallint | 2 | False |  |
|  | Start­Date Date the person started the insurance policy with the provider | date | 3 | True |  |
|  | End­Date Date the person stopped the insurance policy with the provider | date | 3 | True |  |
|  | Group­ID insurance company and policy group id | varchar(20) | 20 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Insurance | Person­ID, Insurance­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Insurance\_­Personto­Insurance | Insurance­ID->[[dbo].[Insurance­Provider].[Insurance­Provider­ID]](#(local)/User_databases/LCCHPDev/Tables/InsuranceProvider) |
| FK\_­Personto­Insurance\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Insurance]  (  [Person­ID] [int] NOT NULL,  [Insurance­ID] [smallint] NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL,  [Group­ID] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Insurance\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Insurance] ADD CONSTRAINT [PK\_­Personto­Insurance] PRIMARY KEY CLUSTERED ([Person­ID], [Insurance­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Insurance] ADD CONSTRAINT [FK\_­Personto­Insurance\_­Personto­Insurance] FOREIGN KEY ([Insurance­ID]) REFERENCES [dbo].[Insurance­Provider] ([Insurance­Provider­ID])  GO  ALTER TABLE [dbo].[Personto­Insurance] ADD CONSTRAINT [FK\_­Personto­Insurance\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and insurance', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Insurance', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the person stopped the insurance policy with the provider', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Insurance', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'insurance company and policy group id', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Insurance', 'COLUMN', N'Group­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the person started the insurance policy with the provider', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Insurance', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Insurance­Provider]](#(local)/User_databases/LCCHPDev/Tables/InsuranceProvider)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Insurance]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoInsurance)

|  |
| --- |
| [dbo].[Personto­Language] |

MS\_­Description

linking table for person and language

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 203 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:48 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Language­ID | tinyint | 1 | False |  |
|  | is­Primary­Language 0 = no; 1 = yes; is this language the person's primary language | bit | 1 | False | ((1)) |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Language | Person­ID, Language­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Language\_­Language | Language­ID->[[dbo].[Language].[Language­ID]](#(local)/User_databases/LCCHPDev/Tables/Language) |
| FK\_­Personto­Language\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Language]  (  [Person­ID] [int] NOT NULL,  [Language­ID] [tinyint] NOT NULL,  [is­Primary­Language] [bit] NOT NULL CONSTRAINT [DF\_­Personto­Language\_is­Primary­Language] DEFAULT ((1)),  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Language\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Language] ADD CONSTRAINT [PK\_­Personto­Language] PRIMARY KEY CLUSTERED ([Person­ID], [Language­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Language] ADD CONSTRAINT [FK\_­Personto­Language\_­Language] FOREIGN KEY ([Language­ID]) REFERENCES [dbo].[Language] ([Language­ID])  GO  ALTER TABLE [dbo].[Personto­Language] ADD CONSTRAINT [FK\_­Personto­Language\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and language', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Language', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes; is this language the person''s primary language', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Language', 'COLUMN', N'is­Primary­Language'  GO |

Uses

[[dbo].[Language]](#(local)/User_databases/LCCHPDev/Tables/Language)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage)

[[dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditClientInfoWebScreenInformation)

|  |
| --- |
| [dbo].[Personto­Occupation] |

MS\_­Description

linking table for person and occupatoin

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 7 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 12:53:47 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Occupation­ID | int | 4 | False |  |
|  | Start­Date date the person started the occupation | date | 3 | False | (getdate()) |
|  | End­Date date the person ceased the occupation | date | 3 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Occupation | Person­ID, Occupation­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Occupation\_­Occupation | Occupation­ID->[[dbo].[Occupation].[Occupation­ID]](#(local)/User_databases/LCCHPDev/Tables/Occupation) |
| FK\_­Personto­Occupation\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Occupation]  (  [Person­ID] [int] NOT NULL,  [Occupation­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Personto­Occupation\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Occupation\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Occupation] ADD CONSTRAINT [PK\_­Personto­Occupation] PRIMARY KEY CLUSTERED ([Person­ID], [Occupation­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Occupation] ADD CONSTRAINT [FK\_­Personto­Occupation\_­Occupation] FOREIGN KEY ([Occupation­ID]) REFERENCES [dbo].[Occupation] ([Occupation­ID])  GO  ALTER TABLE [dbo].[Personto­Occupation] ADD CONSTRAINT [FK\_­Personto­Occupation\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and occupatoin', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Occupation', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person ceased the occupation', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Occupation', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person started the occupation', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Occupation', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Occupation]](#(local)/User_databases/LCCHPDev/Tables/Occupation)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

[[dbo].[usp\_up­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upOccupation)

|  |
| --- |
| [dbo].[Personto­Person] |

MS\_­Description

collection of relationships between people

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 1:27:15 AM Thursday, March 26, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person1ID | int | 4 | False |  |
|  | Person2ID | int | 4 | False |  |
|  | Relationship­Type­ID relationship­Type is how P1 relates to P2 | int | 4 | False |  |
|  | is­Guardian is­Guardian is 1 if P1 is P2's guardian | bit | 1 | True |  |
|  | is­Primary­Contact is­Primary­Contact is 1 if P1 is P2's primary­Contact | bit | 1 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |
|  | Modified­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Person | Person1ID, Person2ID, Relationship­Type­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Person\_­Person1ID 1st person in the relationship | Person1ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Personto­Person\_­Person2ID 2nd person in the relationship | Person2ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Personto­Person\_­Relationship­Type how is person1 related to person2 | Relationship­Type­ID->[[dbo].[Relationship­Type].[Relationship­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/RelationshipType) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Person]  (  [Person1ID] [int] NOT NULL,  [Person2ID] [int] NOT NULL,  [Relationship­Type­ID] [int] NOT NULL,  [is­Guardian] [bit] NULL,  [is­Primary­Contact] [bit] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Person\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Person\_­Modified­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Person] ADD CONSTRAINT [PK\_­Personto­Person] PRIMARY KEY CLUSTERED ([Person1ID], [Person2ID], [Relationship­Type­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Person] ADD CONSTRAINT [FK\_­Personto­Person\_­Person1ID] FOREIGN KEY ([Person1ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Personto­Person] ADD CONSTRAINT [FK\_­Personto­Person\_­Person2ID] FOREIGN KEY ([Person2ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Personto­Person] ADD CONSTRAINT [FK\_­Personto­Person\_­Relationship­Type] FOREIGN KEY ([Relationship­Type­ID]) REFERENCES [dbo].[Relationship­Type] ([Relationship­Type­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of relationships between people', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'is­Guardian is 1 if P1 is P2''s guardian', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'COLUMN', N'is­Guardian'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'is­Primary­Contact is 1 if P1 is P2''s primary­Contact', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'COLUMN', N'is­Primary­Contact'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'relationship­Type is how P1 relates to P2', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'COLUMN', N'Relationship­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'1st person in the relationship', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'CONSTRAINT', N'FK\_­Personto­Person\_­Person1ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'2nd person in the relationship', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'CONSTRAINT', N'FK\_­Personto­Person\_­Person2ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'how is person1 related to person2', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Person', 'CONSTRAINT', N'FK\_­Personto­Person\_­Relationship­Type'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Relationship­Type]](#(local)/User_databases/LCCHPDev/Tables/RelationshipType)

Used By

[[dbo].[usp\_­Insert­Personto­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPerson)

|  |
| --- |
| [dbo].[Personto­Phone­Number] |

MS\_­Description

linking table for person and phonenumber

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 7:07:13 PM Saturday, April 4, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Phone­Number­ID | int | 4 | False |  |
|  | Number­Priority order which this number should be used to contact the person (1 being first, 2 being 2nd . . . ) | tinyint | 1 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Phone­Number | Person­ID, Phone­Number­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Phone­Number\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Personto­Phone­Number\_­Phone­Number | Phone­Number­ID->[[dbo].[Phone­Number].[Phone­Number­ID]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Phone­Number]  (  [Person­ID] [int] NOT NULL,  [Phone­Number­ID] [int] NOT NULL,  [Number­Priority] [tinyint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Phone­Number\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Phone­Number] ADD CONSTRAINT [PK\_­Personto­Phone­Number] PRIMARY KEY CLUSTERED ([Person­ID], [Phone­Number­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Phone­Number] ADD CONSTRAINT [FK\_­Personto­Phone­Number\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Personto­Phone­Number] ADD CONSTRAINT [FK\_­Personto­Phone­Number\_­Phone­Number] FOREIGN KEY ([Phone­Number­ID]) REFERENCES [dbo].[Phone­Number] ([Phone­Number­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and phonenumber', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Phone­Number', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'order which this number should be used to contact the person (1 being first, 2 being 2nd . . . )', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Phone­Number', 'COLUMN', N'Number­Priority'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber)

Used By

[[dbo].[usp\_­Insert­Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPhoneNumber)

|  |
| --- |
| [dbo].[Personto­Property] |

MS\_­Description

linking table for person and property - indicating when a person occuppied a property

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 4765 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:57 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­ID | int | 4 | False |  |  |
|  | Property­ID | int | 4 | False |  |  |
|  | Start­Date date the person started occuppying the property | date | 3 | False |  | (getdate()) |
|  | End­Date date the person stopped occuppying the property | date | 3 | True |  |  |
|  | is­Primary­Residence | bit | 1 | True |  |  |
|  | Family­ID Primary family id mainly from legacy system | int | 4 | True |  |  |
|  | Personto­Property­ID | int | 4 | False | 1 - 1 |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Personto­Property | Personto­Property­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Personto­Property\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Personto­Property\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Personto­Property]  (  [Person­ID] [int] NOT NULL,  [Property­ID] [int] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Personto­Property\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [is­Primary­Residence] [bit] NULL,  [Family­ID] [int] NULL,  [Personto­Property­ID] [int] NOT NULL IDENTITY(1, 1),  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Personto­Property\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Property] ADD CONSTRAINT [PK\_­Personto­Property] PRIMARY KEY CLUSTERED ([Personto­Property­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Personto­Property] ADD CONSTRAINT [FK\_­Personto­Property\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Personto­Property] ADD CONSTRAINT [FK\_­Personto­Property\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and property - indicating when a person occuppied a property', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person stopped occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Property', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Primary family id mainly from legacy system', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Property', 'COLUMN', N'Family­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person started occuppying the property', 'SCHEMA', N'dbo', 'TABLE', N'Personto­Property', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Personto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoProperty)

|  |
| --- |
| [dbo].[Person­To­Travel­Country] |

MS\_­Description

linking table for person and country traveled too

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:56:57 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Person­ID | int | 4 | False |  |
|  | Country­ID | tinyint | 1 | False |  |
|  | Start­Date date the person entered the country | date | 3 | False | (getdate()) |
|  | End­Date date the person left the country | date | 3 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­To­Travel­Country | Person­ID, Country­ID, Start­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­To­Travel­Country\_­Country | Country­ID->[[dbo].[Country].[Country­ID]](#(local)/User_databases/LCCHPDev/Tables/Country) |
| FK\_­Person­To­Travel­Country\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­To­Travel­Country]  (  [Person­ID] [int] NOT NULL,  [Country­ID] [tinyint] NOT NULL,  [Start­Date] [date] NOT NULL CONSTRAINT [DF\_­Person­To­Travel­Country\_­Start­Date] DEFAULT (getdate()),  [End­Date] [date] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­To­Travel­Country\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­To­Travel­Country] ADD CONSTRAINT [PK\_­Person­To­Travel­Country] PRIMARY KEY CLUSTERED ([Person­ID], [Country­ID], [Start­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Person­To­Travel­Country] ADD CONSTRAINT [FK\_­Person­To­Travel­Country\_­Country] FOREIGN KEY ([Country­ID]) REFERENCES [dbo].[Country] ([Country­ID])  GO  ALTER TABLE [dbo].[Person­To­Travel­Country] ADD CONSTRAINT [FK\_­Person­To­Travel­Country\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for person and country traveled too', 'SCHEMA', N'dbo', 'TABLE', N'Person­To­Travel­Country', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person left the country', 'SCHEMA', N'dbo', 'TABLE', N'Person­To­Travel­Country', 'COLUMN', N'End­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the person entered the country', 'SCHEMA', N'dbo', 'TABLE', N'Person­To­Travel­Country', 'COLUMN', N'Start­Date'  GO |

Uses

[[dbo].[Country]](#(local)/User_databases/LCCHPDev/Tables/Country)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Personto­Travel­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoTravelCountry)

|  |
| --- |
| [dbo].[Person­Travel­Notes] |

MS\_­Description

table for person Travel notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 36 |
| Created | 7:08:32 PM Saturday, April 11, 2015 |
| Last Modified | 7:08:32 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Person­Travel­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Person­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Person­Travel­Notes | Person­Travel­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Person­Travel­Notes\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Person­Travel­Notes]  (  [Person­Travel­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Person­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Person­Travel­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Person­Travel­Notes] ADD CONSTRAINT [PK\_­Person­Travel­Notes] PRIMARY KEY CLUSTERED ([Person­Travel­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Person­Travel­Notes] ADD CONSTRAINT [FK\_­Person­Travel­Notes\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for person Travel notes', 'SCHEMA', N'dbo', 'TABLE', N'Person­Travel­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Person­Travel­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

Used By

[[dbo].[usp\_­Insert­Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes)

|  |
| --- |
| [dbo].[Phone­Number] |

MS\_­Description

collection of phone number objects

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 73 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 7:07:13 PM Saturday, April 4, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Phone­Number­ID | int | 4 | False | 1 - 1 |  |
|  | Country­Code code for the country | tinyint | 1 | False |  | ((1)) |
|  | Phone­Number telephone number | bigint | 8 | False |  |  |
|  | Phone­Number­Type­ID | tinyint | 1 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Phone­Number | Phone­Number­ID | True | UData |
|  | IX\_­Phone­Number | Phone­Number | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Phone­Number | True | True | After Update |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Phone­Number\_­Phone­Number | Phone­Number­Type­ID->[[dbo].[Phone­Number­Type].[Phone­Number­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Phone­Number]  (  [Phone­Number­ID] [int] NOT NULL IDENTITY(1, 1),  [Country­Code] [tinyint] NOT NULL CONSTRAINT [DF\_­Phone­Number\_­Country­Code] DEFAULT ((1)),  [Phone­Number] [bigint] NOT NULL,  [Phone­Number­Type­ID] [tinyint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Phone­Number\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Phone­Number] on [dbo].[Phone­Number] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Phone­Number set Modified­Date = getdate() where Phone­Number­ID in (select Phone­Number­ID from inserted)  end  GO  ALTER TABLE [dbo].[Phone­Number] ADD CONSTRAINT [PK\_­Phone­Number] PRIMARY KEY CLUSTERED ([Phone­Number­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Phone­Number] ADD CONSTRAINT [IX\_­Phone­Number] UNIQUE NONCLUSTERED ([Phone­Number]) ON [UData]  GO  ALTER TABLE [dbo].[Phone­Number] ADD CONSTRAINT [FK\_­Phone­Number\_­Phone­Number] FOREIGN KEY ([Phone­Number­Type­ID]) REFERENCES [dbo].[Phone­Number­Type] ([Phone­Number­Type­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of phone number objects', 'SCHEMA', N'dbo', 'TABLE', N'Phone­Number', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'code for the country', 'SCHEMA', N'dbo', 'TABLE', N'Phone­Number', 'COLUMN', N'Country­Code'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'telephone number', 'SCHEMA', N'dbo', 'TABLE', N'Phone­Number', 'COLUMN', N'Phone­Number'  GO |

Uses

[[dbo].[Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType)

Used By

[[dbo].[Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/FamilytoPhoneNumber)

[[dbo].[Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PersontoPhoneNumber)

[[dbo].[usp\_­Insert­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber)

[[dbo].[udf\_­Sl­Family­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_SlFamilyPhoneNumber)

|  |
| --- |
| [dbo].[Phone­Number­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 11:30:26 PM Friday, December 19, 2014 |
| Last Modified | 9:34:40 AM Saturday, April 18, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Phone­Number­Type­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Phone­Number­Type­Name | varchar(50) | 50 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Historic­Phone­Code | char(1) | 1 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Phone­Number­Type | Phone­Number­Type­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Phone­Number­Type | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Phone­Number­Type]  (  [Phone­Number­Type­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Phone­Number­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Phone­Number­Type\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Historic­Phone­Code] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Phone­Number­Type] on [dbo].[Phone­Number­Type] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Phone­Number­Type set Modified­Date = getdate() where Phone­Number­Type­ID in (select Phone­Number­Type­ID from inserted)  end  GO  ALTER TABLE [dbo].[Phone­Number­Type] ADD CONSTRAINT [PK\_­Phone­Number­Type] PRIMARY KEY CLUSTERED ([Phone­Number­Type­ID]) ON [UData]  GO |

Used By

[[dbo].[Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber)

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_­Insert­Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumberType)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[Property] |

MS\_­Description

collection of properties and basic attributes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 10916 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 10:57:42 PM Monday, June 1, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Property­ID unique identifier for the property object | int | 4 | False | 1 - 1 |  |
|  | Construction­Type­ID | tinyint | 1 | True |  |  |
|  | Area­ID | int | 4 | True |  |  |
|  | isin­Historic­District | bit | 1 | True |  |  |
|  | is­Remodeled | bit | 1 | True |  |  |
|  | Remodel­Date | date | 3 | True |  |  |
|  | isin­City­Limits | bit | 1 | True |  |  |
|  | Street­Number | varchar(15) | 15 | True |  |  |
|  | Address­Line1 | varchar(100) | 100 | True |  |  |
|  | Street­Suffix | varchar(20) | 20 | True |  |  |
|  | Address­Line2 | varchar(100) | 100 | True |  |  |
|  | City | varchar(50) | 50 | True |  |  |
|  | State | char(2) | 2 | True |  |  |
|  | Zipcode | varchar(12) | 12 | True |  |  |
|  | Owner­ID | int | 4 | True |  |  |
|  | is­Owner­Occuppied | bit | 1 | True |  |  |
|  | Replaced­Pipes­Faucets | tinyint | 1 | True |  |  |
|  | Total­Remediation­Costs | money | 8 | True |  |  |
|  | is­Residential | bit | 1 | True |  |  |
|  | is­Currently­Being­Remodeled | bit | 1 | True |  |  |
|  | has­Peeling­Chipping­Paint | bit | 1 | True |  |  |
|  | County | varchar(50) | 50 | True |  |  |
|  | is­Rental | bit | 1 | True |  |  |
|  | Historic­Property­ID | smallint | 2 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Year­Built | date | 3 | True |  |  |
|  | Street | varchar(50) | 50 | True |  |  |
|  | Review­Status­ID | tinyint | 1 | True |  |  |
|  | Assessors­Office­ID Identification number from Assessor's office | varchar(50) | 50 | True |  |  |
|  | Kids­First­ID Kids First identification number | int | 4 | True |  |  |
|  | Clean­UPStatus­ID Cleanup status id for the current property cleanup state | tinyint | 1 | True |  |  |
|  | Owner­Contact­Information basic contact information for the property owner | varchar(1000) | 1000 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Property | Property­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Property | True | True | After Update |

Foreign Keys

|  |  |  |
| --- | --- | --- |
| Name | No Check | Columns |
| FK\_­Property\_­Area |  | Area­ID->[[dbo].[Area].[Area­ID]](#(local)/User_databases/LCCHPDev/Tables/Area) |
| FK\_­Property\_­Cleanup­Status | True | Clean­UPStatus­ID->[[dbo].[Cleanup­Status].[Cleanup­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus) |
| FK\_­Property\_­Construction­Type |  | Construction­Type­ID->[[dbo].[Construction­Type].[Construction­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/ConstructionType) |
| FK\_­Property\_­Person |  | Owner­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Property\_­Release­Status |  | Review­Status­ID->[[dbo].[Release­Status].[Release­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/ReleaseStatus) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Property]  (  [Property­ID] [int] NOT NULL IDENTITY(1, 1),  [Construction­Type­ID] [tinyint] NULL,  [Area­ID] [int] NULL,  [isin­Historic­District] [bit] NULL,  [is­Remodeled] [bit] NULL,  [Remodel­Date] [date] NULL,  [isin­City­Limits] [bit] NULL,  [Street­Number] [varchar] (15) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Address­Line1] [varchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Street­Suffix] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Address­Line2] [varchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [City] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [State] [char] (2) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Zipcode] [varchar] (12) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Owner­ID] [int] NULL,  [is­Owner­Occuppied] [bit] NULL,  [Replaced­Pipes­Faucets] [tinyint] NULL,  [Total­Remediation­Costs] [money] NULL,  [is­Residential] [bit] NULL,  [is­Currently­Being­Remodeled] [bit] NULL,  [has­Peeling­Chipping­Paint] [bit] NULL,  [County] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [is­Rental] [bit] NULL,  [Historic­Property­ID] [smallint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Property\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Year­Built] [date] NULL,  [Street] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Review­Status­ID] [tinyint] NULL,  [Assessors­Office­ID] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Kids­First­ID] [int] NULL,  [Clean­UPStatus­ID] [tinyint] NULL,  [Owner­Contact­Information] [varchar] (1000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Property] on [dbo].[Property] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Property set Modified­Date = getdate() where Property­ID in (select Property­ID from inserted)  end  GO  ALTER TABLE [dbo].[Property] ADD CONSTRAINT [PK\_­Property] PRIMARY KEY CLUSTERED ([Property­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Property] ADD CONSTRAINT [FK\_­Property\_­Area] FOREIGN KEY ([Area­ID]) REFERENCES [dbo].[Area] ([Area­ID])  GO  ALTER TABLE [dbo].[Property] WITH NOCHECK ADD CONSTRAINT [FK\_­Property\_­Cleanup­Status] FOREIGN KEY ([Clean­UPStatus­ID]) REFERENCES [dbo].[Cleanup­Status] ([Cleanup­Status­ID])  GO  ALTER TABLE [dbo].[Property] ADD CONSTRAINT [FK\_­Property\_­Construction­Type] FOREIGN KEY ([Construction­Type­ID]) REFERENCES [dbo].[Construction­Type] ([Construction­Type­ID])  GO  ALTER TABLE [dbo].[Property] ADD CONSTRAINT [FK\_­Property\_­Person] FOREIGN KEY ([Owner­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Property] ADD CONSTRAINT [FK\_­Property\_­Release­Status] FOREIGN KEY ([Review­Status­ID]) REFERENCES [dbo].[Release­Status] ([Release­Status­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of properties and basic attributes', 'SCHEMA', N'dbo', 'TABLE', N'Property', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Identification number from Assessor''s office', 'SCHEMA', N'dbo', 'TABLE', N'Property', 'COLUMN', N'Assessors­Office­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Cleanup status id for the current property cleanup state', 'SCHEMA', N'dbo', 'TABLE', N'Property', 'COLUMN', N'Clean­UPStatus­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Kids First identification number', 'SCHEMA', N'dbo', 'TABLE', N'Property', 'COLUMN', N'Kids­First­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'basic contact information for the property owner', 'SCHEMA', N'dbo', 'TABLE', N'Property', 'COLUMN', N'Owner­Contact­Information'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the property object', 'SCHEMA', N'dbo', 'TABLE', N'Property', 'COLUMN', N'Property­ID'  GO |

Uses

[[dbo].[Area]](#(local)/User_databases/LCCHPDev/Tables/Area)

[[dbo].[Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus)

[[dbo].[Construction­Type]](#(local)/User_databases/LCCHPDev/Tables/ConstructionType)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Release­Status]](#(local)/User_databases/LCCHPDev/Tables/ReleaseStatus)

Used By

[[dbo].[Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement)

[[dbo].[Contractorto­Property]](#(local)/User_databases/LCCHPDev/Tables/ContractortoProperty)

[[dbo].[Daycareto­Property]](#(local)/User_databases/LCCHPDev/Tables/DaycaretoProperty)

[[dbo].[Employerto­Property]](#(local)/User_databases/LCCHPDev/Tables/EmployertoProperty)

[[dbo].[Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Tables/EnvironmentalInvestigation)

[[dbo].[Familyto­Property]](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty)

[[dbo].[Personto­Property]](#(local)/User_databases/LCCHPDev/Tables/PersontoProperty)

[[dbo].[Property­Notes]](#(local)/User_databases/LCCHPDev/Tables/PropertyNotes)

[[dbo].[Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults)

[[dbo].[Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/PropertytoCleanupStatus)

[[dbo].[Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Tables/PropertytoHouseholdSourcesofLead)

[[dbo].[Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Tables/PropertytoMedium)

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp\_up­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty)

[[dbo].[udf\_­Does­Property­Exist]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DoesPropertyExist)

|  |
| --- |
| [dbo].[Property­Link­Type] |

MS\_­Description

Collection of property link types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 3:33:40 PM Saturday, April 11, 2015 |
| Last Modified | 5:59:48 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Property­Link­Type­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Property­Link­Type­Description Detailed description of the property link type | varchar(253) | 253 | True |  |  |
|  | Property­Link­Type­Name short name for the property link type | varchar(50) | 50 | True |  |  |
|  | Historic­Property­Link­Type­ID historic flg ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Property­Link­Type | Property­Link­Type­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Property­Link­Type]  (  [Property­Link­Type­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Property­Link­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Property­Link­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Property­Link­Type­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Property­Link­Type\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Property­Link­Type] ADD CONSTRAINT [PK\_­Property­Link­Type] PRIMARY KEY CLUSTERED ([Property­Link­Type­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of property link types', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic flg ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', 'COLUMN', N'Historic­Property­Link­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the property link type', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', 'COLUMN', N'Property­Link­Type­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the property link type', 'SCHEMA', N'dbo', 'TABLE', N'Property­Link­Type', 'COLUMN', N'Property­Link­Type­Name'  GO |

Used By

[[dbo].[Familyto­Property]](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty)

|  |
| --- |
| [dbo].[Property­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 5439 |
| Created | 12:32:17 AM Tuesday, February 17, 2015 |
| Last Modified | 12:32:17 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Property­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Property­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Property­Notes | Property­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Property­Notes\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Property­Notes]  (  [Property­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Property­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Property­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Property­Notes] ADD CONSTRAINT [PK\_­Property­Notes] PRIMARY KEY CLUSTERED ([Property­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Property­Notes] ADD CONSTRAINT [FK\_­Property­Notes\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Property­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Property­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Property­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes)

|  |
| --- |
| [dbo].[Property­Sample­Results] |

MS\_­Description

collection of property test results

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 4 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 12:21:37 PM Saturday, February 21, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Property­Sample­Results­ID unique identifier for property test results | int | 4 | False | 1 - 1 |  |
|  | is­Baseline is this a baseline test result for the property | bit | 1 | False |  | ((0)) |
|  | Property­ID id of the property to which the test results apply | int | 4 | False |  |  |
|  | Lab­Submission­Date date the proeprty test samples were submitted to the lab | date | 3 | True |  |  |
|  | Lab­ID id of the lab to which the property samples were submitted | int | 4 | True |  |  |
|  | Sample­Type­ID id of the sample type | tinyint | 1 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Property­Samplet­Results | Property­Sample­Results­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Property­Sample­Results | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Property­Sample­Results\_­Lab­Submission­Date | Lab­Submission­Date | ([dbo].[udf\_­Date­In­The­Past]([Lab­Submission­Date])=(1)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Property­Samplet­Results\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |
| FK\_­Property­Sample­Results\_­Sample­Type | Sample­Type­ID->[[dbo].[Sample­Type].[Sample­Type­ID]](#(local)/User_databases/LCCHPDev/Tables/SampleType) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Property­Sample­Results]  (  [Property­Sample­Results­ID] [int] NOT NULL IDENTITY(1, 1),  [is­Baseline] [bit] NOT NULL CONSTRAINT [DF\_­Property­Test­Results\_is­Baseline] DEFAULT ((0)),  [Property­ID] [int] NOT NULL,  [Lab­Submission­Date] [date] NULL,  [Lab­ID] [int] NULL,  [Sample­Type­ID] [tinyint] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Property­Sample­Results\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Property­Sample­Results] on [dbo].[Property­Sample­Results] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Property­Sample­Results set Modified­Date = getdate() where Property­Sample­Results­ID in (select Property­Sample­Results­ID from inserted)  end  GO  ALTER TABLE [dbo].[Property­Sample­Results] ADD CONSTRAINT [ck\_­Property­Sample­Results\_­Lab­Submission­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Lab­Submission­Date])=(1)))  GO  ALTER TABLE [dbo].[Property­Sample­Results] ADD CONSTRAINT [PK\_­Property­Samplet­Results] PRIMARY KEY CLUSTERED ([Property­Sample­Results­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Property­Sample­Results] ADD CONSTRAINT [FK\_­Property­Samplet­Results\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  ALTER TABLE [dbo].[Property­Sample­Results] ADD CONSTRAINT [FK\_­Property­Sample­Results\_­Sample­Type] FOREIGN KEY ([Sample­Type­ID]) REFERENCES [dbo].[Sample­Type] ([Sample­Type­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of property test results', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'is this a baseline test result for the property', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'is­Baseline'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the lab to which the property samples were submitted', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'Lab­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the proeprty test samples were submitted to the lab', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'Lab­Submission­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the property to which the test results apply', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'Property­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for property test results', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'Property­Sample­Results­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the sample type', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results', 'COLUMN', N'Sample­Type­ID'  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[Sample­Type]](#(local)/User_databases/LCCHPDev/Tables/SampleType)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResultsNotes)

[[dbo].[usp\_­Insert­Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResults)

|  |
| --- |
| [dbo].[Property­Sample­Results­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 12:34:16 AM Tuesday, February 17, 2015 |
| Last Modified | 12:37:34 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Property­Sample­Results­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Property­Sample­Results­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Property­Sample­Results­Notes | Property­Sample­Results­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Property­Sample­Results­Notes\_­Property­Sample­Results | Property­Sample­Results­ID->[[dbo].[Property­Sample­Results].[Property­Sample­Results­ID]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Property­Sample­Results­Notes]  (  [Property­Sample­Results­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Property­Sample­Results­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Property­Sample­Results­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Property­Sample­Results­Notes] ADD CONSTRAINT [PK\_­Property­Sample­Results­Notes] PRIMARY KEY CLUSTERED ([Property­Sample­Results­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Property­Sample­Results­Notes] ADD CONSTRAINT [FK\_­Property­Sample­Results­Notes\_­Property­Sample­Results] FOREIGN KEY ([Property­Sample­Results­ID]) REFERENCES [dbo].[Property­Sample­Results] ([Property­Sample­Results­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Property­Sample­Results­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults)

Used By

[[dbo].[usp\_­Insert­Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResultsNotes)

|  |
| --- |
| [dbo].[Propertyto­Cleanup­Status] |

MS\_­Description

linking table for property and cleanup status

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 7 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:11:32 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Property­ID | int | 4 | False |  |
|  | Cleanup­Status­ID | tinyint | 1 | False |  |
|  | Cleanup­Status­Date date of the cleanup status | date | 3 | False | (getdate()) |
|  | Costof­Cleanup cost of the cleanup | money | 8 | True |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Propertyto­Cleanup­Status | Property­ID, Cleanup­Status­ID, Cleanup­Status­Date | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Propertyto­Cleanup­Status\_­Cleanup­Status | Cleanup­Status­ID->[[dbo].[Cleanup­Status].[Cleanup­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus) |
| FK\_­Propertyto­Cleanup­Status\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Propertyto­Cleanup­Status]  (  [Property­ID] [int] NOT NULL,  [Cleanup­Status­ID] [tinyint] NOT NULL,  [Cleanup­Status­Date] [date] NOT NULL CONSTRAINT [DF\_­Propertyto­Cleanup­Status\_­Cleanup­Status­Date] DEFAULT (getdate()),  [Costof­Cleanup] [money] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Propertyto­Cleanup­Status\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Cleanup­Status] ADD CONSTRAINT [PK\_­Propertyto­Cleanup­Status] PRIMARY KEY CLUSTERED ([Property­ID], [Cleanup­Status­ID], [Cleanup­Status­Date]) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Cleanup­Status] ADD CONSTRAINT [FK\_­Propertyto­Cleanup­Status\_­Cleanup­Status] FOREIGN KEY ([Cleanup­Status­ID]) REFERENCES [dbo].[Cleanup­Status] ([Cleanup­Status­ID])  GO  ALTER TABLE [dbo].[Propertyto­Cleanup­Status] ADD CONSTRAINT [FK\_­Propertyto­Cleanup­Status\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for property and cleanup status', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Cleanup­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date of the cleanup status', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Cleanup­Status', 'COLUMN', N'Cleanup­Status­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'cost of the cleanup', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Cleanup­Status', 'COLUMN', N'Costof­Cleanup'  GO |

Uses

[[dbo].[Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoCleanupStatus)

|  |
| --- |
| [dbo].[Propertyto­Household­Sourcesof­Lead] |

MS\_­Description

linking table for property and household sources of lead

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:11:32 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Property­ID | int | 4 | False |  |
|  | Household­Sourcesof­Lead­ID | int | 4 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Propertyto­Household­Sourcesof­Lead | Property­ID, Household­Sourcesof­Lead­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Household­Sourcesof­Lead\_­Propertyto­Household­Sourcesof­Lead | Household­Sourcesof­Lead­ID->[[dbo].[Household­Sourcesof­Lead].[Household­Sourcesof­Lead­ID]](#(local)/User_databases/LCCHPDev/Tables/HouseholdSourcesofLead) |
| FK\_­Property\_­Propertyto­Household­Sourcesof­Lead | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Propertyto­Household­Sourcesof­Lead]  (  [Property­ID] [int] NOT NULL,  [Household­Sourcesof­Lead­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Propertyto­Household­Sourcesof­Lead\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Household­Sourcesof­Lead] ADD CONSTRAINT [PK\_­Propertyto­Household­Sourcesof­Lead] PRIMARY KEY CLUSTERED ([Property­ID], [Household­Sourcesof­Lead­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Household­Sourcesof­Lead] ADD CONSTRAINT [FK\_­Household­Sourcesof­Lead\_­Propertyto­Household­Sourcesof­Lead] FOREIGN KEY ([Household­Sourcesof­Lead­ID]) REFERENCES [dbo].[Household­Sourcesof­Lead] ([Household­Sourcesof­Lead­ID])  GO  ALTER TABLE [dbo].[Propertyto­Household­Sourcesof­Lead] ADD CONSTRAINT [FK\_­Property\_­Propertyto­Household­Sourcesof­Lead] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for property and household sources of lead', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Household­Sourcesof­Lead', NULL, NULL  GO |

Uses

[[dbo].[Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Tables/HouseholdSourcesofLead)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoHouseholdSourcesofLead)

|  |
| --- |
| [dbo].[Propertyto­Medium] |

MS\_­Description

linking table for property and media

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 2:11:32 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Default |
|  | Property­ID | int | 4 | False |  |
|  | Medium­ID | int | 4 | False |  |
|  | Medium­Tested 0 - yes; 1 - no. Has the medium been tested. | bit | 1 | False |  |
|  | Created­Date | datetime | 8 | True | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Propertyto­Medium | Property­ID, Medium­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Propertyto­Medium\_­Medium | Medium­ID->[[dbo].[Medium].[Medium­ID]](#(local)/User_databases/LCCHPDev/Tables/Medium) |
| FK\_­Propertyto­Medium\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Propertyto­Medium]  (  [Property­ID] [int] NOT NULL,  [Medium­ID] [int] NOT NULL,  [Medium­Tested] [bit] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Propertyto­Medium\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Medium] ADD CONSTRAINT [PK\_­Propertyto­Medium] PRIMARY KEY CLUSTERED ([Property­ID], [Medium­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Propertyto­Medium] ADD CONSTRAINT [FK\_­Propertyto­Medium\_­Medium] FOREIGN KEY ([Medium­ID]) REFERENCES [dbo].[Medium] ([Medium­ID])  GO  ALTER TABLE [dbo].[Propertyto­Medium] ADD CONSTRAINT [FK\_­Propertyto­Medium\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for property and media', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Medium', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 - yes; 1 - no. Has the medium been tested.', 'SCHEMA', N'dbo', 'TABLE', N'Propertyto­Medium', 'COLUMN', N'Medium­Tested'  GO |

Uses

[[dbo].[Medium]](#(local)/User_databases/LCCHPDev/Tables/Medium)

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoMedium)

|  |
| --- |
| [dbo].[Questionnaire] |

MS\_­Description

collection of questionnaire questions and answers, typically only completed by flagged patients

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 6156 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 1:05:24 PM Thursday, June 18, 2015 |

Columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Computed | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Questionnaire­ID unique identifier for the questionnaire object | int |  | 4 | False | 1 - 1 |  |
| (2) | Person­ID id of the patient the questionnaire is referring to | int |  | 4 | False |  |  |
| (2) | Questionnaire­Date Date the questionnaire was completed | date |  | 3 | True |  |  |
|  | Questionnaire­Data­Source­ID id of the person completing the questionnaire | int |  | 4 | True |  |  |
|  | Visit­Remodeled­Property 0 = no; 1 = yes. does the patient frequently visited remodeled properties | bit |  | 1 | True |  |  |
|  | is­Exposedto­Peeling­Paint 0 = no; 1 = yes. has the patient been exposed to peeling paint | bit |  | 1 | True |  |  |
|  | is­Taking­Vitamins 0 = no; 1 = yes. Is the patient taking vitamins regularly | bit |  | 1 | True |  | ((0)) |
|  | Nursing­Mother 0 = no; 1 = yes. is the patient a mother nursing a child | bit |  | 1 | True |  | ((0)) |
|  | is­Using­Pacifier 0 = no; 1 = yes. is the patient using a pacifier | bit |  | 1 | True |  | ((0)) |
|  | is­Using­Bottle 0 = no; 1 = yes. is the patient using a bottle | bit |  | 1 | True |  | ((0)) |
|  | Bites­Nails 0 = no; 1 = yes. does the patient bite nails | bit |  | 1 | True |  | ((0)) |
|  | Non­Food­Eating 0 = no; 1 = yes. does the patient consume non food products | bit |  | 1 | True |  | ((0)) |
|  | Non­Foodin­Mouth 0 = no; 1 = yes. does the patient put non food items in mouth? | bit |  | 1 | True |  | ((0)) |
|  | Eat­Outside 0 = no; 1 = yes. does the patient eat outside? | bit |  | 1 | True |  | ((0)) |
|  | Suckling 0 = no; 1 = yes. does the patient suck his/her thumb or suckle | bit |  | 1 | True |  | ((0)) |
|  | Frequent­Hand­Washing 0 = no; 1 = yes. does the patient frequently wash hands througout the day | bit |  | 1 | True |  | ((0)) |
|  | Daycare­ID id of the daycare the patient attends | int |  | 4 | True |  |  |
|  | Created­Date Date the record was created | datetime |  | 8 | True |  | (getdate()) |
|  | Modified­Date date the record was last modified | datetime |  | 8 | True |  |  |
|  | Review­Status­ID Review Status ID | tinyint |  | 1 | True |  |  |
|  | Mouthing 0 = no; 1 = yes. does the client mouth things frequently | bit |  | 1 | True |  |  |
|  | Visits­Old­Homes 0 = no; 1 = yes. does the patient visit older homes | bit |  | 1 | True |  |  |
|  | Nursing­Infant 0 = no; 1 = yes. is the patient a nursing infant | bit |  | 1 | True |  |  |
|  | Pregnant 0 = no; 1 = yes. is the patient pregnant | bit |  | 1 | True |  |  |
|  | Paint­Date | date |  | 3 | True |  |  |
|  | Remodel­Property­Date | date |  | 3 | True |  |  |
|  | Paint­Age | int | True | 4 | True |  |  |
|  | Remodel­Property­Age | int | True | 4 | True |  |  |

Computed columns

|  |  |
| --- | --- |
| Name | Column definition |
| Paint­Age | ([dbo].[udf\_­Calculate­Age]([Paint­Date],getdate())) |
| Remodel­Property­Age | ([dbo].[udf\_­Calculate­Age]([Remodel­Property­Date],getdate())) |

Indexes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | Fill Factor | File Group |
|  | PK\_­Questionnaire | Questionnaire­ID | True |  | UData |
|  | IDX\_­Questionnaire­Date­IDPerson­IDNursing­Mother­Pregnant | Questionnaire­ID, Person­ID, Nursing­Mother, Pregnant, Questionnaire­Date |  |  | UData |
|  | Non­Clustered­Index-Person­IDQuestionnaire­Date | Person­ID, Questionnaire­Date |  | 90 | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Questionnaire | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Questionnaire\_­Paint­Date | Paint­Date | ([dbo].[udf\_­Date­In­The­Past]([Paint­Date])=(1) OR [Paint­Date] IS NULL) |
| ck\_­Questionnaire\_­Questionnaire­Date | Questionnaire­Date | ([dbo].[udf\_­Date­In­The­Past]([Questionnaire­Date])=(1)) |
| ck\_­Questionnaire\_­Remodel­Property­Date | Remodel­Property­Date | ([dbo].[udf\_­Date­In­The­Past]([Remodel­Property­Date])=(1) OR [Remodel­Property­Date] IS NULL) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Questionnaire\_­Daycare | Daycare­ID->[[dbo].[Daycare].[Daycare­ID]](#(local)/User_databases/LCCHPDev/Tables/Daycare) |
| FK\_­Questionnaire\_­Person | Person­ID->[[dbo].[Person].[Person­ID]](#(local)/User_databases/LCCHPDev/Tables/Person) |
| FK\_­Questionnaire\_­Review­Status | Review­Status­ID->[[dbo].[Review­Status].[Review­Status­ID]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Questionnaire]  (  [Questionnaire­ID] [int] NOT NULL IDENTITY(1, 1),  [Person­ID] [int] NOT NULL,  [Questionnaire­Date] [date] NULL,  [Questionnaire­Data­Source­ID] [int] NULL,  [Visit­Remodeled­Property] [bit] NULL,  [is­Exposedto­Peeling­Paint] [bit] NULL,  [is­Taking­Vitamins] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_is­Taking­Vitamins] DEFAULT ((0)),  [Nursing­Mother] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_is­Nursing] DEFAULT ((0)),  [is­Using­Pacifier] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_is­Using­Pacifier] DEFAULT ((0)),  [is­Using­Bottle] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_is­Using­Bottle] DEFAULT ((0)),  [Bites­Nails] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Bitesnails] DEFAULT ((0)),  [Non­Food­Eating] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Non­Food­Eating] DEFAULT ((0)),  [Non­Foodin­Mouth] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Non­Foodin­Mouth] DEFAULT ((0)),  [Eat­Outside] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Eat­Outside] DEFAULT ((0)),  [Suckling] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Suckling] DEFAULT ((0)),  [Frequent­Hand­Washing] [bit] NULL CONSTRAINT [DF\_­Questionnaire\_­Frequent­Hand­Washing] DEFAULT ((0)),  [Daycare­ID] [int] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Questionnaire\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL,  [Review­Status­ID] [tinyint] NULL,  [Mouthing] [bit] NULL,  [Visits­Old­Homes] [bit] NULL,  [Nursing­Infant] [bit] NULL,  [Pregnant] [bit] NULL,  [Paint­Date] [date] NULL,  [Remodel­Property­Date] [date] NULL,  [Paint­Age] AS ([dbo].[udf\_­Calculate­Age]([Paint­Date],getdate())),  [Remodel­Property­Age] AS ([dbo].[udf\_­Calculate­Age]([Remodel­Property­Date],getdate()))  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Questionnaire] on [dbo].[Questionnaire] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Questionnaire set Modified­Date = getdate() where Questionnaire­ID in (select Questionnaire­ID from inserted)  end  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [ck\_­Questionnaire\_­Paint­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Paint­Date])=(1) OR [Paint­Date] IS NULL))  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [ck\_­Questionnaire\_­Questionnaire­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Questionnaire­Date])=(1)))  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [ck\_­Questionnaire\_­Remodel­Property­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Remodel­Property­Date])=(1) OR [Remodel­Property­Date] IS NULL))  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [PK\_­Questionnaire] PRIMARY KEY CLUSTERED ([Questionnaire­ID]) ON [UData]  GO  CREATE NONCLUSTERED INDEX [Non­Clustered­Index-Person­IDQuestionnaire­Date] ON [dbo].[Questionnaire] ([Person­ID], [Questionnaire­Date]) ON [UData]  GO  CREATE NONCLUSTERED INDEX [IDX\_­Questionnaire­Date­IDPerson­IDNursing­Mother­Pregnant] ON [dbo].[Questionnaire] ([Questionnaire­Date]) INCLUDE ([Nursing­Mother], [Person­ID], [Pregnant], [Questionnaire­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [FK\_­Questionnaire\_­Daycare] FOREIGN KEY ([Daycare­ID]) REFERENCES [dbo].[Daycare] ([Daycare­ID])  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [FK\_­Questionnaire\_­Person] FOREIGN KEY ([Person­ID]) REFERENCES [dbo].[Person] ([Person­ID])  GO  ALTER TABLE [dbo].[Questionnaire] ADD CONSTRAINT [FK\_­Questionnaire\_­Review­Status] FOREIGN KEY ([Review­Status­ID]) REFERENCES [dbo].[Review­Status] ([Review­Status­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of questionnaire questions and answers, typically only completed by flagged patients', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient bite nails', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Bites­Nails'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the daycare the patient attends', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Daycare­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient eat outside?', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Eat­Outside'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient frequently wash hands througout the day', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Frequent­Hand­Washing'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. has the patient been exposed to peeling paint', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'is­Exposedto­Peeling­Paint'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. Is the patient taking vitamins regularly', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'is­Taking­Vitamins'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. is the patient using a bottle', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'is­Using­Bottle'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. is the patient using a pacifier', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'is­Using­Pacifier'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was last modified', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the client mouth things frequently', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Mouthing'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient consume non food products', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Non­Food­Eating'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient put non food items in mouth?', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Non­Foodin­Mouth'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. is the patient a nursing infant', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Nursing­Infant'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. is the patient a mother nursing a child', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Nursing­Mother'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the patient the questionnaire is referring to', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Person­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. is the patient pregnant', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Pregnant'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'id of the person completing the questionnaire', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Questionnaire­Data­Source­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Date the questionnaire was completed', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Questionnaire­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the questionnaire object', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Questionnaire­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Review Status ID', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Review­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient suck his/her thumb or suckle', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Suckling'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient frequently visited remodeled properties', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Visit­Remodeled­Property'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0 = no; 1 = yes. does the patient visit older homes', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire', 'COLUMN', N'Visits­Old­Homes'  GO |

Uses

[[dbo].[Daycare]](#(local)/User_databases/LCCHPDev/Tables/Daycare)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Review­Status]](#(local)/User_databases/LCCHPDev/Tables/ReviewStatus)

[[dbo].[udf\_­Calculate­Age]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_CalculateAge)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Tables/QuestionnaireNotes)

[[dbo].[v­Most­Recent­Questionnaires]](#(local)/User_databases/LCCHPDev/Views/vMostRecentQuestionnaires)

[[dbo].[usp\_­Insert­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

[[dbo].[usp\_up­Client­Flag]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag)

[[dbo].[usp\_up­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire)

|  |
| --- |
| [dbo].[Questionnaire­Data­Source] |

MS\_­Description

source of the data (Environmental group or Blood Lead)

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 7:58:38 PM Sunday, April 19, 2015 |
| Last Modified | 8:01:47 PM Sunday, April 19, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity |
|  | Questionnaire­Data­Source­ID unique identifier for the questionnaire data source | int | 4 | False | 1 - 1 |
|  | Questionnaire­Data­Source­Name Source of the questionnaire data - enviornmental or blood lead | varchar(50) | 50 | False |  |
|  | Questionnaire­Data­Source­Description More details about the source of the questionnaire data - enviornmental or blood lead | varchar(253) | 253 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Questionnaire­Data­Source | Questionnaire­Data­Source­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Questionnaire­Data­Source]  (  [Questionnaire­Data­Source­ID] [int] NOT NULL IDENTITY(1, 1),  [Questionnaire­Data­Source­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Questionnaire­Data­Source­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Questionnaire­Data­Source] ADD CONSTRAINT [PK\_­Questionnaire­Data­Source] PRIMARY KEY CLUSTERED ([Questionnaire­Data­Source­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'source of the data (Environmental group or Blood Lead)', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Data­Source', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'More details about the source of the questionnaire data - enviornmental or blood lead', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Data­Source', 'COLUMN', N'Questionnaire­Data­Source­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the questionnaire data source', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Data­Source', 'COLUMN', N'Questionnaire­Data­Source­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Source of the questionnaire data - enviornmental or blood lead', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Data­Source', 'COLUMN', N'Questionnaire­Data­Source­Name'  GO |

|  |
| --- |
| [dbo].[Questionnaire­Notes] |

MS\_­Description

linking table for access agreement and access agreement notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1457 |
| Created | 12:36:12 AM Tuesday, February 17, 2015 |
| Last Modified | 12:36:12 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Questionnaire­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Questionnaire­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Questionnaire­Notes | Questionnaire­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Questionnaire­Notes\_­Questionnaire | Questionnaire­ID->[[dbo].[Questionnaire].[Questionnaire­ID]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Questionnaire­Notes]  (  [Questionnaire­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Questionnaire­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Questionnaire­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Questionnaire­Notes] ADD CONSTRAINT [PK\_­Questionnaire­Notes] PRIMARY KEY CLUSTERED ([Questionnaire­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Questionnaire­Notes] ADD CONSTRAINT [FK\_­Questionnaire­Notes\_­Questionnaire] FOREIGN KEY ([Questionnaire­ID]) REFERENCES [dbo].[Questionnaire] ([Questionnaire­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'linking table for access agreement and access agreement notes', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Questionnaire­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

Used By

[[dbo].[usp\_­Insert­Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes)

|  |
| --- |
| [dbo].[Relationship­Type] |

MS\_­Description

collection of Relationship­Type names and basic attributes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 24 |
| Created | 5:38:08 PM Saturday, January 3, 2015 |
| Last Modified | 10:11:41 PM Saturday, January 3, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Relationship­Type­ID unique identifier for the Relationship­Type object | int | 4 | False | 1 - 1 |  |
|  | Relationship­Type­Name | varchar(50) | 50 | True |  |  |
|  | Relationship­Type­Description | varchar(253) | 253 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Modified­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Relationship­Type | Relationship­Type­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Relationship­Type | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Relationship­Type]  (  [Relationship­Type­ID] [int] NOT NULL IDENTITY(1, 1),  [Relationship­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Relationship­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Relationship­Type\_­Created­Date] DEFAULT (getdate()),  [Modified­Date] [datetime] NULL CONSTRAINT [DF\_­Relationship­Type\_­Modified­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Relationship­Type] on [dbo].[Relationship­Type] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Relationship­Type set Modified­Date = getdate() where Relationship­Type­ID in (select Relationship­Type­ID from inserted)  end  GO  ALTER TABLE [dbo].[Relationship­Type] ADD CONSTRAINT [PK\_­Relationship­Type] PRIMARY KEY CLUSTERED ([Relationship­Type­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of Relationship­Type names and basic attributes', 'SCHEMA', N'dbo', 'TABLE', N'Relationship­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for the Relationship­Type object', 'SCHEMA', N'dbo', 'TABLE', N'Relationship­Type', 'COLUMN', N'Relationship­Type­ID'  GO |

Used By

[[dbo].[Personto­Person]](#(local)/User_databases/LCCHPDev/Tables/PersontoPerson)

|  |
| --- |
| [dbo].[Release­Status] |

MS\_­Description

Collection of Release Status

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 4:56:45 PM Saturday, April 11, 2015 |
| Last Modified | 2:17:22 PM Sunday, April 19, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Release­Status­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Release­Status­Description Detailed description of the Release status | varchar(253) | 253 | True |  |  |
|  | Release­Status­Name short name for the Release status | varchar(50) | 50 | True |  |  |
|  | Historic­Release­Status­ID historic Release­Status ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Release­Status | Release­Status­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Release­Status]  (  [Release­Status­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Release­Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Release­Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Release­Status­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Release­Status\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Release­Status] ADD CONSTRAINT [PK\_­Release­Status] PRIMARY KEY CLUSTERED ([Release­Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of Release Status', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic Release­Status ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', 'COLUMN', N'Historic­Release­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the Release status', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', 'COLUMN', N'Release­Status­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the Release status', 'SCHEMA', N'dbo', 'TABLE', N'Release­Status', 'COLUMN', N'Release­Status­Name'  GO |

Used By

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

|  |
| --- |
| [dbo].[Remediation] |

MS\_­Description

collection of remediation data

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 8 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 12:31:41 PM Saturday, February 21, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Remediation­ID | int | 4 | False | 1 - 1 |  |
|  | Remediation­Approval­Date | date | 3 | True |  |  |
|  | Remediation­Start­Date | date | 3 | True |  |  |
|  | Remediation­End­Date | date | 3 | True |  |  |
|  | Property­ID | int | 4 | True |  |  |
|  | Access­Agreement­ID | int | 4 | True |  |  |
|  | Final­Remediation­Report­File | varbinary(max) | max | True |  |  |
|  | Final­Remediation­Report­Date | date | 3 | True |  |  |
|  | Remediation­Cost | money | 8 | True |  |  |
|  | One­Year­Remediation­Complete­Date | date | 3 | True |  |  |
|  | One­Year­Remediation­Complete | bit | 1 | True |  |  |
|  | Remediation­Action­Plan­ID | int | 4 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Remediation | Remediation­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Remediation | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Remediation\_­Remediation­Approval­Date | Remediation­Approval­Date | ([dbo].[udf\_­Date­In­The­Past]([Remediation­Approval­Date])=(1)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Remediation\_­Property | Property­ID->[[dbo].[Property].[Property­ID]](#(local)/User_databases/LCCHPDev/Tables/Property) |
| FK\_­Remediation\_­Remediation­Action­Plan | Remediation­Action­Plan­ID->[[dbo].[Remediation­Action­Plan].[Remediation­Action­Plan­ID]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Remediation]  (  [Remediation­ID] [int] NOT NULL IDENTITY(1, 1),  [Remediation­Approval­Date] [date] NULL,  [Remediation­Start­Date] [date] NULL,  [Remediation­End­Date] [date] NULL,  [Property­ID] [int] NULL,  [Access­Agreement­ID] [int] NULL,  [Final­Remediation­Report­File] [varbinary] (max) NULL,  [Final­Remediation­Report­Date] [date] NULL,  [Remediation­Cost] [money] NULL,  [One­Year­Remediation­Complete­Date] [date] NULL,  [One­Year­Remediation­Complete] [bit] NULL,  [Remediation­Action­Plan­ID] [int] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Remediation\_­Created­Date] DEFAULT (getdate())  ) ON [UData] TEXTIMAGE\_­ON [UData]  GO  create trigger [dbo].[tr­Update­Remediation] on [dbo].[Remediation] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Remediation set Modified­Date = getdate() where Remediation­ID in (select Remediation­ID from inserted)  end  GO  ALTER TABLE [dbo].[Remediation] ADD CONSTRAINT [ck\_­Remediation\_­Remediation­Approval­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Remediation­Approval­Date])=(1)))  GO  ALTER TABLE [dbo].[Remediation] ADD CONSTRAINT [PK\_­Remediation] PRIMARY KEY CLUSTERED ([Remediation­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Remediation] ADD CONSTRAINT [FK\_­Remediation\_­Property] FOREIGN KEY ([Property­ID]) REFERENCES [dbo].[Property] ([Property­ID])  GO  ALTER TABLE [dbo].[Remediation] ADD CONSTRAINT [FK\_­Remediation\_­Remediation­Action­Plan] FOREIGN KEY ([Remediation­Action­Plan­ID]) REFERENCES [dbo].[Remediation­Action­Plan] ([Remediation­Action­Plan­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of remediation data', 'SCHEMA', N'dbo', 'TABLE', N'Remediation', NULL, NULL  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediation)

[[dbo].[Remediation­Notes]](#(local)/User_databases/LCCHPDev/Tables/RemediationNotes)

[[dbo].[usp\_­Insert­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediation)

|  |
| --- |
| [dbo].[Remediation­Action­Plan] |

MS\_­Description

collection of sampling plans

Properties

|  |  |
| --- | --- |
| Property | Value |
| File Group | UData |
| Row Count (~) | 3 |
| Created | 1:12:50 AM Saturday, December 27, 2014 |
| Last Modified | 12:32:21 PM Saturday, February 21, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Remediation­Action­Plan­ID | int | 4 | False | 1 - 1 |  |
|  | Remediation­Action­Plan­Approval­Date | date | 3 | True |  |  |
|  | Home­Owner­Consultation­Date Meeting date between homeowner and workgroup to review the sampling plan | date | 3 | True |  |  |
|  | Contractor­Completed­Investigation­Date | date | 3 | True |  |  |
|  | Remediation­Action­Plan­Final­Report­Submission­Date | date | 3 | True |  |  |
|  | Remediation­Action­Plan­File | varbinary(max) | max | True |  |  |
|  | Property­ID | int | 4 | True |  |  |
|  | Environmental­Investigation­ID | int | 4 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Remediation­Action­Plan | Remediation­Action­Plan­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Remediation­Action­Plan | True | True | After Update |

Check Constraints

|  |  |  |
| --- | --- | --- |
| Name | On Column | Constraint |
| ck\_­Remediation­Action­Plan\_­Remediation­Action­Plan­Approval­Date | Remediation­Action­Plan­Approval­Date | ([dbo].[udf\_­Date­In­The­Past]([Remediation­Action­Plan­Approval­Date])=(1)) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Remediation­Action­Plan\_­Environmental­Investigation | Environmental­Investigation­ID->[[dbo].[Environmental­Investigation].[Environmental­Investigation­ID]](#(local)/User_databases/LCCHPDev/Tables/EnvironmentalInvestigation) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Remediation­Action­Plan]  (  [Remediation­Action­Plan­ID] [int] NOT NULL IDENTITY(1, 1),  [Remediation­Action­Plan­Approval­Date] [date] NULL,  [Home­Owner­Consultation­Date] [date] NULL,  [Contractor­Completed­Investigation­Date] [date] NULL,  [Remediation­Action­Plan­Final­Report­Submission­Date] [date] NULL,  [Remediation­Action­Plan­File] [varbinary] (max) NULL,  [Property­ID] [int] NULL,  [Environmental­Investigation­ID] [int] NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Remediation­Action­Plan\_­Created­Date] DEFAULT (getdate())  ) ON [UData] TEXTIMAGE\_­ON [UData]  GO  create trigger [dbo].[tr­Update­Remediation­Action­Plan] on [dbo].[Remediation­Action­Plan] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Remediation­Action­Plan set Modified­Date = getdate() where Remediation­Action­Plan­ID in (select Remediation­Action­Plan­ID from inserted)  end  GO  ALTER TABLE [dbo].[Remediation­Action­Plan] ADD CONSTRAINT [ck\_­Remediation­Action­Plan\_­Remediation­Action­Plan­Approval­Date] CHECK (([dbo].[udf\_­Date­In­The­Past]([Remediation­Action­Plan­Approval­Date])=(1)))  GO  ALTER TABLE [dbo].[Remediation­Action­Plan] ADD CONSTRAINT [PK\_­Remediation­Action­Plan] PRIMARY KEY CLUSTERED ([Remediation­Action­Plan­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Remediation­Action­Plan] ADD CONSTRAINT [FK\_­Remediation­Action­Plan\_­Environmental­Investigation] FOREIGN KEY ([Environmental­Investigation­ID]) REFERENCES [dbo].[Environmental­Investigation] ([Environmental­Investigation­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of sampling plans', 'SCHEMA', N'dbo', 'TABLE', N'Remediation­Action­Plan', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Meeting date between homeowner and workgroup to review the sampling plan', 'SCHEMA', N'dbo', 'TABLE', N'Remediation­Action­Plan', 'COLUMN', N'Home­Owner­Consultation­Date'  GO |

Uses

[[dbo].[Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Tables/EnvironmentalInvestigation)

[[dbo].[udf\_­Date­In­The­Past]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast)

Used By

[[dbo].[Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediationActionPlan)

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

[[dbo].[usp\_­Insert­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationActionPlan)

|  |
| --- |
| [dbo].[Remediation­Notes] |

MS\_­Description

table for remediation notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 12:41:02 AM Tuesday, February 17, 2015 |
| Last Modified | 12:41:02 AM Tuesday, February 17, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Remediation­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Remediation­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Remediation­Notes | Remediation­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Remediation­Notes\_­Remediation | Remediation­ID->[[dbo].[Remediation].[Remediation­ID]](#(local)/User_databases/LCCHPDev/Tables/Remediation) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Remediation­Notes]  (  [Remediation­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Remediation­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Remediation­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Remediation­Notes] ADD CONSTRAINT [PK\_­Remediation­Notes] PRIMARY KEY CLUSTERED ([Remediation­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Remediation­Notes] ADD CONSTRAINT [FK\_­Remediation­Notes\_­Remediation] FOREIGN KEY ([Remediation­ID]) REFERENCES [dbo].[Remediation] ([Remediation­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'table for remediation notes', 'SCHEMA', N'dbo', 'TABLE', N'Remediation­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Remediation­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

Used By

[[dbo].[usp\_­Insert­Remediation­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationNotes)

|  |
| --- |
| [dbo].[Review­Status] |

MS\_­Description

Collection of potential status for Review

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 5:46:41 PM Saturday, April 11, 2015 |
| Last Modified | 8:53:57 PM Sunday, April 19, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Review­Status­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Review­Status­Description Detailed description of the Review status | varchar(253) | 253 | True |  |  |
|  | Review­Status­Name status for the Review | varchar(50) | 50 | True |  |  |
|  | Historic­Review­Status­ID historic status from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Review­Status | Review­Status­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Review­Status]  (  [Review­Status­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Review­Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Review­Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Review­Status­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Review­Status\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Review­Status] ADD CONSTRAINT [PK\_­Review­Status] PRIMARY KEY CLUSTERED ([Review­Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of potential status for Review', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic status from access database', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', 'COLUMN', N'Historic­Review­Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the Review status', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', 'COLUMN', N'Review­Status­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'status for the Review', 'SCHEMA', N'dbo', 'TABLE', N'Review­Status', 'COLUMN', N'Review­Status­Name'  GO |

Used By

[[dbo].[Familyto­Property]](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

|  |
| --- |
| [dbo].[Sample­Level­Category] |

MS\_­Description

collection of sample level categorizations

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 1 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 8:24:48 PM Wednesday, March 4, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Sample­Level­Category­ID unique identifier for sample level categorization | tinyint | 1 | False | 1 - 1 |  |
|  | Sample­Level­Category­Name description of sample level category | varchar(50) | 50 | True |  |  |
|  | Sample­Level­Category­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Sample­Level­Category | Sample­Level­Category­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Sample­Level­Category | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Sample­Level­Category]  (  [Sample­Level­Category­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Sample­Level­Category­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Sample­Level­Category­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Sample­Level­Category\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Sample­Level­Category] on [dbo].[Sample­Level­Category] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Sample­Level­Category set Modified­Date = getdate() where Sample­Level­Category­ID in (select Sample­Level­Category­ID from inserted)  end  GO  ALTER TABLE [dbo].[Sample­Level­Category] ADD CONSTRAINT [PK\_­Sample­Level­Category] PRIMARY KEY CLUSTERED ([Sample­Level­Category­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of sample level categorizations', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Level­Category', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier for sample level categorization', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Level­Category', 'COLUMN', N'Sample­Level­Category­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'description of sample level category', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Level­Category', 'COLUMN', N'Sample­Level­Category­Name'  GO |

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[usp\_­Insert­Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleLevelCategory)

|  |
| --- |
| [dbo].[Sample­Purpose] |

MS\_­Description

Collection of sample purposes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 0 |
| Created | 5:07:52 PM Saturday, April 11, 2015 |
| Last Modified | 5:07:52 PM Saturday, April 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Sample­Purpose­ID | tinyint | 1 | False | 1 - 1 |  |
|  | Sample­Purpose­Description Detailed description of the sample purpose | varchar(253) | 253 | True |  |  |
|  | Sample­Purpose­Name short name for the sample purpose | varchar(50) | 50 | True |  |  |
|  | Historic­Sample­Purpose­ID historic Sample­Purpose ID from access database | char(1) | 1 | True |  |  |
|  | Modified­Date last modified date for the record | datetime | 8 | True |  |  |
|  | Created­Date date the record was created | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Sample­Purpose | Sample­Purpose­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Sample­Purpose]  (  [Sample­Purpose­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Sample­Purpose­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Sample­Purpose­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Historic­Sample­Purpose­ID] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Sample­Purpose\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  ALTER TABLE [dbo].[Sample­Purpose] ADD CONSTRAINT [PK\_­Sample­Purpose] PRIMARY KEY CLUSTERED ([Sample­Purpose­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of sample purposes', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the record was created', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', 'COLUMN', N'Created­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'historic Sample­Purpose ID from access database', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', 'COLUMN', N'Historic­Sample­Purpose­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'last modified date for the record', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', 'COLUMN', N'Modified­Date'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Detailed description of the sample purpose', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', 'COLUMN', N'Sample­Purpose­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'short name for the sample purpose', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Purpose', 'COLUMN', N'Sample­Purpose­Name'  GO |

|  |
| --- |
| [dbo].[Sample­Type] |

MS\_­Description

collection of sample types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 11 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 8:24:48 PM Wednesday, March 4, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Sample­Type­ID unique identifier of sample type | tinyint | 1 | False | 1 - 1 |  |
|  | Sample­Type­Name friendly name for the sample type | varchar(50) | 50 | True |  |  |
|  | Sample­Type­Description extended description of the sample type | varchar(253) | 253 | True |  |  |
|  | historic­Sample­Type | char(1) | 1 | True |  |  |
|  | Sample­Target | varchar(50) | 50 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Sample­Type | Sample­Type­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Sample­Type | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Sample­Type]  (  [Sample­Type­ID] [tinyint] NOT NULL IDENTITY(1, 1),  [Sample­Type­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Sample­Type­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [historic­Sample­Type] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Sample­Target] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Sample­Type\_­Created­Date] DEFAULT (getdate())  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Sample­Type] on [dbo].[Sample­Type] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Sample­Type set Modified­Date = getdate() where Sample­Type­ID in (select Sample­Type­ID from inserted)  end  GO  ALTER TABLE [dbo].[Sample­Type] ADD CONSTRAINT [PK\_­Sample­Type] PRIMARY KEY CLUSTERED ([Sample­Type­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of sample types', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Type', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'extended description of the sample type', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Type', 'COLUMN', N'Sample­Type­Description'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of sample type', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Type', 'COLUMN', N'Sample­Type­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'friendly name for the sample type', 'SCHEMA', N'dbo', 'TABLE', N'Sample­Type', 'COLUMN', N'Sample­Type­Name'  GO |

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults)

[[dbo].[usp\_­Insert­Sample­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleType)

|  |
| --- |
| [dbo].[Source] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 2 |
| Created | 8:27:57 PM Tuesday, January 27, 2015 |
| Last Modified | 8:27:57 PM Tuesday, January 27, 2015 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity |
|  | Source­ID | int | 4 | False | 1 - 1 |
|  | Source­Name | varchar(50) | 50 | False |  |
|  | Source­Description | varchar(253) | 253 | True |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Source | Source­ID | True | UData |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Source]  (  [Source­ID] [int] NOT NULL IDENTITY(1, 1),  [Source­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Source­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Source] ADD CONSTRAINT [PK\_­Source] PRIMARY KEY CLUSTERED ([Source­ID]) ON [UData]  GO |

|  |
| --- |
| [dbo].[Target­Status] |

MS\_­Description

collection of status objects

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 22 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 5:31:42 PM Thursday, June 11, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Status­ID unique identifier of status objects | smallint | 2 | False | 1 - 1 |  |
|  | Status­Name friendly name/description of status object | varchar(50) | 50 | True |  |  |
|  | Status­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Target­Type | varchar(50) | 50 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Status | Status­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Status | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Target­Status]  (  [Status­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Status­Name] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Status­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Status\_­Created­Date] DEFAULT (getdate()),  [Target­Type] [varchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  CREATE trigger [dbo].[tr­Update­Status] on [dbo].[Target­Status] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Target­Status set Modified­Date = getdate() where Status­ID in (select Status­ID from inserted)  end  GO  ALTER TABLE [dbo].[Target­Status] ADD CONSTRAINT [PK\_­Status] PRIMARY KEY CLUSTERED ([Status­ID]) ON [UData]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'collection of status objects', 'SCHEMA', N'dbo', 'TABLE', N'Target­Status', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'unique identifier of status objects', 'SCHEMA', N'dbo', 'TABLE', N'Target­Status', 'COLUMN', N'Status­ID'  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'friendly name/description of status object', 'SCHEMA', N'dbo', 'TABLE', N'Target­Status', 'COLUMN', N'Status­Name'  GO |

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp\_­Sl­Child­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlChildStatus)

[[dbo].[usp\_­Sl­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlStatus)

|  |
| --- |
| [dbo].[Travel­Notes] |

MS\_­Description

Collection of family and travel notes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 12 |
| Created | 3:31:08 PM Thursday, March 19, 2015 |
| Last Modified | 3:36:55 PM Thursday, March 19, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Travel­Notes­ID | int | 4 | False | 1 - 1 |  |
|  | Family­ID | int | 4 | False |  |  |
|  | Created­Date date the notes where added | datetime | 8 | True |  | (getdate()) |
|  | Notes | varchar(3000) | 3000 | False |  |  |
|  | Start­Date | date | 3 | True |  |  |
|  | End­Date | date | 3 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Travel­Notes | Travel­Notes­ID | True | UData |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Travel­Notes\_­Family | Family­ID->[[dbo].[Family].[Family­ID]](#(local)/User_databases/LCCHPDev/Tables/Family) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Travel­Notes]  (  [Travel­Notes­ID] [int] NOT NULL IDENTITY(1, 1),  [Family­ID] [int] NOT NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Travel­Notes\_­Created­Date] DEFAULT (getdate()),  [Notes] [varchar] (3000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Start­Date] [date] NULL,  [End­Date] [date] NULL  ) ON [UData]  GO  ALTER TABLE [dbo].[Travel­Notes] ADD CONSTRAINT [PK\_­Travel­Notes] PRIMARY KEY CLUSTERED ([Travel­Notes­ID]) ON [UData]  GO  ALTER TABLE [dbo].[Travel­Notes] ADD CONSTRAINT [FK\_­Travel­Notes\_­Family] FOREIGN KEY ([Family­ID]) REFERENCES [dbo].[Family] ([Family­ID])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Collection of family and travel notes', 'SCHEMA', N'dbo', 'TABLE', N'Travel­Notes', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'date the notes where added', 'SCHEMA', N'dbo', 'TABLE', N'Travel­Notes', 'COLUMN', N'Created­Date'  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

Used By

[[dbo].[usp\_­Insert­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertTravelNotes)

|  |
| --- |
| [dbo].[Units] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| File Group | UData |
| Row Count (~) | 7 |
| Created | 7:49:31 PM Friday, August 29, 2014 |
| Last Modified | 9:16:43 PM Thursday, April 9, 2015 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Units­ID | smallint | 2 | False | 1 - 1 |  |
|  | Units | varchar(20) | 20 | False |  |  |
|  | Units­Description | varchar(253) | 253 | True |  |  |
|  | Modified­Date | datetime | 8 | True |  |  |
|  | Created­Date | datetime | 8 | True |  | (getdate()) |
|  | Historic­Units­Code | char(1) | 1 | True |  |  |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Columns | Unique | File Group |
|  | PK\_­Units | Units­ID | True | UData |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| tr­Update­Units | True | True | After Update |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[Units]  (  [Units­ID] [smallint] NOT NULL IDENTITY(1, 1),  [Units] [varchar] (20) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [Units­Description] [varchar] (253) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Modified­Date] [datetime] NULL,  [Created­Date] [datetime] NULL CONSTRAINT [DF\_­Units\_­Created­Date] DEFAULT (getdate()),  [Historic­Units­Code] [char] (1) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [UData]  GO  create trigger [dbo].[tr­Update­Units] on [dbo].[Units] AFTER UPDATE  as  begin  if @@rowcount = 0  return  if not update(Modified­Date) update Units set Modified­Date = getdate() where Units­ID in (select Units­ID from inserted)  end  GO  ALTER TABLE [dbo].[Units] ADD CONSTRAINT [PK\_­Units] PRIMARY KEY CLUSTERED ([Units­ID]) ON [UData]  GO |

Used By

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

|  |
| --- |
| Views |

Objects

|  |
| --- |
| Name |
| [dbo.v­Adults](#(local)/User_databases/LCCHPDev/Views/vAdults) |
| [dbo.v­Most­Recent­Blood­Test­Results](#(local)/User_databases/LCCHPDev/Views/vMostRecentBloodTestResults) |
| [dbo.v­Most­Recent­Questionnaires](#(local)/User_databases/LCCHPDev/Views/vMostRecentQuestionnaires) |
| [dbo.v­Nursing­Infants](#(local)/User_databases/LCCHPDev/Views/vNursingInfants) |
| [dbo.v­Nursing­Mothers](#(local)/User_databases/LCCHPDev/Views/vNursingMothers) |
| [dbo.v­Pregnant](#(local)/User_databases/LCCHPDev/Views/vPregnant) |

|  |
| --- |
| [dbo].[v­Adults] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 8:02:42 PM Friday, May 29, 2015 |
| Last Modified | 6:50:45 PM Thursday, June 4, 2015 |

Columns

|  |
| --- |
| Name |
| Person­ID |
| Last­Name |
| First­Name |
| Age |
| Gender |
| Birth­Date |
| Pregnant |
| Nursing­Mother |

SQL Script

|  |
| --- |
| CREATE VIEW [dbo].[v­Adults]  AS  SELECT P.Person­ID, P.Last­Name, P.First­Name, P.Age, P.Gender,P.Birth­Date, P.Pregnant, P.Nursing­Mother  FROM dbo.Person AS P  WHERE (P.Age > 17) and P.is­Client = 1 AND P.Pregnant = 0 AND P.Nursing­Mother = 0  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[v­Most­Recent­Blood­Test­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 5:50:00 PM Friday, May 29, 2015 |
| Last Modified | 5:26:41 PM Thursday, June 11, 2015 |

Columns

|  |
| --- |
| Name |
| Last­Name |
| First­Name |
| Person­ID |
| Blood­Test­Results­ID |
| is­Baseline |
| Sample­Date |
| Lab­Submission­Date |
| Lead­Value |
| Lead­Value­Category­ID |
| Hemoglobin­Value |
| Hemoglobin­Value­Category­ID |
| Hematocrit­Value­Category­ID |
| Lab­ID |
| Blood­Test­Costs |
| Sample­Type­ID |
| Taken­After­Property­Remediation­Completed |
| Modified­Date |
| Created­Date |
| Hematocrit­Value |
| Exclude­Result |
| Client­Status­ID |
| Historic­Blood­Test­Results­ID |
| Historic­Lab­Results­ID |

SQL Script

|  |
| --- |
| CREATE View [dbo].[v­Most­Recent­Blood­Test­Results]  AS  Select [P].[Last­Name],[P].[First­Name],[P].[Person­ID],[BTR].[Blood­Test­Results­ID]  ,[BTR].[is­Baseline]  ,[BTR].[Sample­Date]  ,[BTR].[Lab­Submission­Date]  ,[BTR].[Lead­Value]  ,[BTR].[Lead­Value­Category­ID]  ,[BTR].[Hemoglobin­Value]  ,[BTR].[Hemoglobin­Value­Category­ID]  ,[BTR].[Hematocrit­Value­Category­ID]  ,[BTR].[Lab­ID]  ,[BTR].[Blood­Test­Costs]  ,[BTR].[Sample­Type­ID]  ,[BTR].[Taken­After­Property­Remediation­Completed]  ,[BTR].[Modified­Date]  ,[BTR].[Created­Date]  ,[BTR].[Hematocrit­Value]  ,[BTR].[Exclude­Result]  ,[BTR].[Client­Status­ID]  ,[BTR].[Historic­Blood­Test­Results­ID]  ,[BTR].[Historic­Lab­Results­ID] from [Person] AS [P]  JOIN [Blood­Test­Results] AS [BTR] on [BTR].[Blood­Test­Results­ID] = (  select top 1 [Blood­Test­Results­ID] from [Blood­Test­Results]  where [Blood­Test­Results].[Person­ID] = [P].[Person­ID]  -- AND [Lead­Value] > @Min­Lead­Value uncomment to list most recent tests with BLL above minimum  )  WHERE [P].[is­Client] = 1  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[v­Most­Recent­Questionnaires] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 5:43:01 PM Friday, May 29, 2015 |
| Last Modified | 7:41:49 AM Sunday, June 28, 2015 |

Columns

|  |
| --- |
| Name |
| Last­Name |
| First­Name |
| Person­ID |
| Questionnaire­ID |
| Questionnaire­Date |
| Questionnaire­Data­Source­ID |
| Visit­Remodeled­Property |
| is­Exposedto­Peeling­Paint |
| is­Taking­Vitamins |
| Nursing­Mother |
| is­Using­Pacifier |
| is­Using­Bottle |
| Bites­Nails |
| Non­Food­Eating |
| Non­Foodin­Mouth |
| Eat­Outside |
| Suckling |
| Frequent­Hand­Washing |
| Daycare­ID |
| Created­Date |
| Modified­Date |
| Remodel­Property­Date |
| Remodel­Property­Age |
| Paint­Date |
| Paint­Age |
| Review­Status­ID |
| Mouthing |
| Visits­Old­Homes |
| Nursing­Infant |
| Pregnant |

SQL Script

|  |
| --- |
| CREATE View [dbo].[v­Most­Recent­Questionnaires]  AS  Select [P].[Last­Name],[P].[First­Name],[P].[Person­ID]  ,[Q].[Questionnaire­ID]  ,[Q].[Questionnaire­Date]  ,[Q].[Questionnaire­Data­Source­ID]  ,[Q].[Visit­Remodeled­Property]  ,[Q].[is­Exposedto­Peeling­Paint]  ,[Q].[is­Taking­Vitamins]  ,[Q].[Nursing­Mother]  ,[Q].[is­Using­Pacifier]  ,[Q].[is­Using­Bottle]  ,[Q].[Bites­Nails]  ,[Q].[Non­Food­Eating]  ,[Q].[Non­Foodin­Mouth]  ,[Q].[Eat­Outside]  ,[Q].[Suckling]  ,[Q].[Frequent­Hand­Washing]  ,[Q].[Daycare­ID]  ,[Q].[Created­Date]  ,[Q].[Modified­Date]  ,[Q].[Remodel­Property­Date]  ,[Q].[Remodel­Property­Age]  ,[Q].[Paint­Date]  ,[Q].[Paint­Age]  ,[Q].[Review­Status­ID]  ,[Q].[Mouthing]  ,[Q].[Visits­Old­Homes]  ,[Q].[Nursing­Infant]  ,[Q].[Pregnant]  from [Person] AS [P]  JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select top 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [P].[Person­ID]  -- AND [Lead­Value] > @Min­Lead­Value uncomment to list most recent tests with BLL above minimum  )  WHERE [P].[is­Client] = 1  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

|  |
| --- |
| [dbo].[v­Nursing­Infants] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 6:38:16 PM Friday, May 29, 2015 |
| Last Modified | 8:31:45 PM Friday, May 29, 2015 |

Columns

|  |
| --- |
| Name |
| Person­ID |
| Last­Name |
| First­Name |
| Age |
| Gender |
| Nursing­Infant |

SQL Script

|  |
| --- |
| CREATE VIEW [dbo].[v­Nursing­Infants]  AS  SELECT P.Person­ID, P.Last­Name, P.First­Name, P.Age, P.Gender,P.Nursing­Infant  FROM dbo.Person AS P  WHERE (P.Nursing­Infant = 1)  AND [P].[is­Client] = 1  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[v­Nursing­Mothers] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 6:37:20 PM Friday, May 29, 2015 |
| Last Modified | 8:35:27 PM Friday, May 29, 2015 |

Columns

|  |
| --- |
| Name |
| Person­ID |
| Last­Name |
| First­Name |
| Age |
| Gender |
| Nursing­Mother |

SQL Script

|  |
| --- |
| CREATE VIEW [dbo].[v­Nursing­Mothers]  AS  SELECT P.Person­ID, P.Last­Name, P.First­Name, P.Age, P.Gender,P.Nursing­Mother  FROM dbo.Person AS P  WHERE (P.Nursing­Mother = 1)  AND P.is­Client = 1  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[v­Pregnant] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Created | 6:38:50 PM Friday, May 29, 2015 |
| Last Modified | 8:36:03 PM Friday, May 29, 2015 |

Columns

|  |
| --- |
| Name |
| Person­ID |
| Last­Name |
| First­Name |
| Age |
| Gender |
| Pregnant |

SQL Script

|  |
| --- |
| CREATE VIEW [dbo].[v­Pregnant]  AS  SELECT P.Person­ID, P.Last­Name, P.First­Name, P.Age, P.Gender,P.Pregnant  FROM dbo.Person AS P  WHERE (P.Pregnant = 1)  AND P.is­Client = 1  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| Stored Procedures |

Objects

|  |
| --- |
| Name |
| [dbo.DELETE\_usp\_­Insert­Personto­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_InsertPersontoStatus) |
| [dbo.DELETE\_usp\_­Sl­Count­Clients](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_SlCountClients) |
| [dbo.Trans­Proc](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/TransProc) |
| [dbo.usp\_­Insert­Access­Agreement](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessAgreement) |
| [dbo.usp\_­Insert­Access­Purpose](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessPurpose) |
| [dbo.usp\_­Insert­Area](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertArea) |
| [dbo.usp\_­Insert­Blood­Test­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults) |
| [dbo.usp\_­Insert­Blood­Test­Results­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes) |
| [dbo.usp\_­Insert­Cleanup­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCleanupStatus) |
| [dbo.usp\_­Insert­Construction­Type](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertConstructionType) |
| [dbo.usp\_­Insert­Contractor](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractor) |
| [dbo.usp\_­Insert­Contractorto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoProperty) |
| [dbo.usp\_­Insert­Contractorto­Remediation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediation) |
| [dbo.usp\_­Insert­Contractorto­Remediation­Action­Plan](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediationActionPlan) |
| [dbo.usp\_­Insert­Country](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCountry) |
| [dbo.usp\_­Insert­Daycare](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycare) |
| [dbo.usp\_­Insert­Daycare­Primary­Contact](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycarePrimaryContact) |
| [dbo.usp\_­Insert­Daycareto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycaretoProperty) |
| [dbo.usp\_­Insert­Employer](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployer) |
| [dbo.usp\_­Insert­Employerto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployertoProperty) |
| [dbo.usp\_­Insert­Environmental­Investigation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEnvironmentalInvestigation) |
| [dbo.usp\_­Insert­Ethnicity](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEthnicity) |
| [dbo.usp\_­Insert­Family](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily) |
| [dbo.usp\_­Insert­Family­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes) |
| [dbo.usp\_­Insert­Familyto­Phone­Number](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber) |
| [dbo.usp\_­Insert­Familyto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty) |
| [dbo.usp\_­Insert­Foreign­Food](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFood) |
| [dbo.usp\_­Insert­Foreign­Foodto­Country](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFoodtoCountry) |
| [dbo.usp\_­Insert­Gift­Card](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertGiftCard) |
| [dbo.usp\_­Insert­Hobby](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHobby) |
| [dbo.usp\_­Insert­Home­Remedies](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHomeRemedies) |
| [dbo.usp\_­Insert­Household­Sourcesof­Lead](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHouseholdSourcesofLead) |
| [dbo.usp\_­Insert­Insurance­Provider](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertInsuranceProvider) |
| [dbo.usp\_­Insert­Lab](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLab) |
| [dbo.usp\_­Insert­Lab­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLabNotes) |
| [dbo.usp\_­Insert­Language](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLanguage) |
| [dbo.usp\_­Insert­Medium](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMedium) |
| [dbo.usp\_­Insert­Medium­Sample­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults) |
| [dbo.usp\_­Insert­Medium­Sample­Results­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResultsNotes) |
| [dbo.usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewBloodLeadTestResultsWebScreen) |
| [dbo.usp\_­Insert­New­Client­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen) |
| [dbo.usp\_­Insert­New­Family­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen) |
| [dbo.usp\_­Insert­New­Questionnaire­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen) |
| [dbo.usp\_­Insert­Occupation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertOccupation) |
| [dbo.usp\_­Insert­Person](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson) |
| [dbo.usp\_­Insert­Person­Hobby­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes) |
| [dbo.usp\_­Insert­Person­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes) |
| [dbo.usp\_­Insert­Person­Release­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes) |
| [dbo.usp\_­Insert­Personto­Access­Agreement](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoAccessAgreement) |
| [dbo.usp\_­Insert­Personto­Daycare](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoDaycare) |
| [dbo.usp\_­Insert­Personto­Employer](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEmployer) |
| [dbo.usp\_­Insert­Personto­Ethnicity](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity) |
| [dbo.usp\_­Insert­Personto­Family](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily) |
| [dbo.usp\_­Insert­Personto­Foreign­Food](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoForeignFood) |
| [dbo.usp\_­Insert­Personto­Hobby](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby) |
| [dbo.usp\_­Insert­Personto­Home­Remedy](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHomeRemedy) |
| [dbo.usp\_­Insert­Personto­Insurance](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoInsurance) |
| [dbo.usp\_­Insert­Personto­Language](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage) |
| [dbo.usp\_­Insert­Personto­Occupation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation) |
| [dbo.usp\_­Insert­Personto­Person](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPerson) |
| [dbo.usp\_­Insert­Personto­Phone­Number](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPhoneNumber) |
| [dbo.usp\_­Insert­Personto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoProperty) |
| [dbo.usp\_­Insert­Personto­Travel­Country](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoTravelCountry) |
| [dbo.usp\_­Insert­Person­Travel­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes) |
| [dbo.usp\_­Insert­Phone­Number](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber) |
| [dbo.usp\_­Insert­Phone­Number­Type](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumberType) |
| [dbo.usp\_­Insert­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty) |
| [dbo.usp\_­Insert­Property­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes) |
| [dbo.usp\_­Insert­Property­Sample­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResults) |
| [dbo.usp\_­Insert­Property­Sample­Results­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResultsNotes) |
| [dbo.usp\_­Insert­Propertyto­Cleanup­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoCleanupStatus) |
| [dbo.usp\_­Insert­Propertyto­Household­Sourcesof­Lead](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoHouseholdSourcesofLead) |
| [dbo.usp\_­Insert­Propertyto­Medium](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoMedium) |
| [dbo.usp\_­Insert­Questionnaire](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire) |
| [dbo.usp\_­Insert­Questionnaire­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes) |
| [dbo.usp\_­Insert­Remediation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediation) |
| [dbo.usp\_­Insert­Remediation­Action­Plan](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationActionPlan) |
| [dbo.usp\_­Insert­Remediation­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationNotes) |
| [dbo.usp\_­Insert­Sample­Level­Category](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleLevelCategory) |
| [dbo.usp\_­Insert­Sample­Type](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleType) |
| [dbo.usp\_­Insert­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertStatus) |
| [dbo.usp\_­Insert­Travel­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertTravelNotes) |
| [dbo.usp\_­SLAll­Blood­Test­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults) |
| [dbo.usp\_­SLAll­Blood­Test­Results2](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults2) |
| [dbo.usp\_­SLAll­Blood­Test­Results­Meta­Data](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResultsMetaData) |
| [dbo.usp\_­Sl­Child­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlChildStatus) |
| [dbo.usp\_­Sl­Client­Follow­Up](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlClientFollowUp) |
| [dbo.usp\_­Sl­Column­Details](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlColumnDetails) |
| [dbo.usp\_­Sl­Count­Adults](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults) |
| [dbo.usp\_­Sl­Count­Blood­Lead­Levels](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodLeadLevels) |
| [dbo.usp\_­Sl­Count­Blood­Tests](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodTests) |
| [dbo.usp\_­Sl­Count­Clients](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountClients) |
| [dbo.usp\_­Sl­Count­Family­Members](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountFamilyMembers) |
| [dbo.usp\_­Sl­Count­Home­Visit­Soil­Sample](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountHomeVisitSoilSample) |
| [dbo.usp\_­Sl­Count­New­Clients](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewClients) |
| [dbo.usp\_­Sl­Count­New­People](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewPeople) |
| [dbo.usp\_­Sl­Count­Nursing­Infants](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingInfants) |
| [dbo.usp\_­Sl­Count­Nursing­Mothers](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingMothers) |
| [dbo.usp\_­Sl­Count­People](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeople) |
| [dbo.usp\_­Sl­Count­People­By­Age](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByAge) |
| [dbo.usp\_­Sl­Count­People­By­Age­Group](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByAgeGroup) |
| [dbo.usp\_­Sl­Count­People­By­Last­Name](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByLastName) |
| [dbo.usp\_­Sl­Count­Pregnant­Women](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPregnantWomen) |
| [dbo.usp\_­Sl­Daycare](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlDaycare) |
| [dbo.usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditBloodTestResultsWebScreenInformation) |
| [dbo.usp\_­Sl­Edit­Client­Info­Web­Screen­Information](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditClientInfoWebScreenInformation) |
| [dbo.usp\_­Sl­Edit­Family­Web­Screen­Information](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditFamilyWebScreenInformation) |
| [dbo.usp\_­Sl­Edit­Property­Web­Screen­Information](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditPropertyWebScreenInformation) |
| [dbo.usp\_­Sl­Edit­Questionnaire­Web­Screen­Information](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditQuestionnaireWebScreenInformation) |
| [dbo.usp\_­Sl­Family­Members](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyMembers) |
| [dbo.usp\_­Sl­Family­Nameto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyNametoProperty) |
| [dbo.usp\_­Sl­Hobby](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlHobby) |
| [dbo.usp\_­SLInserted­Data](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedData) |
| [dbo.usp\_­SLInserted­Data­Simplified](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified) |
| [dbo.usp\_­Sl­Lab­Name](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlLabName) |
| [dbo.usp\_­SLList­All­Family­Members](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListAllFamilyMembers) |
| [dbo.usp\_­Sl­List­Clients­By­Createdate](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByCreatedate) |
| [dbo.usp\_­Sl­List­Clients­By­Modifieddate](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByModifieddate) |
| [dbo.usp\_­Sl­List­Families](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilies) |
| [dbo.usp\_­Sl­List­Family­Members](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilyMembers) |
| [dbo.usp\_­Sl­List­Nursing­Womenby­Create­Date­Range](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListNursingWomenbyCreateDateRange) |
| [dbo.usp\_­Sl­List­Peopleby­Create­Date­Range](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPeoplebyCreateDateRange) |
| [dbo.usp\_­SLList­Potential­Duplicate­People](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicatePeople) |
| [dbo.usp\_­SLList­Potential­Duplicate­Properties](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicateProperties) |
| [dbo.usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPregnantWomenbyCreateDateRange) |
| [dbo.usp\_­SLMost­Recent­Blood­Test­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLMostRecentBloodTestResults) |
| [dbo.usp\_­Sl­Person­Notes](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersonNotes) |
| [dbo.usp\_­Sl­Personto­Ethnicity](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoEthnicity) |
| [dbo.usp\_­Sl­Personto­Language](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoLanguage) |
| [dbo.usp\_­Sl­Relation­Ship­Types](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlRelationShipTypes) |
| [dbo.usp\_­Sl­Status](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlStatus) |
| [dbo.usp\_­Sl­Summary­Report](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport) |
| [dbo.usp\_­Sl­Summary­Report\_­Meta­Data](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport_MetaData) |
| [dbo.usp\_­Sl­Target­Sample­Type](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlTargetSampleType) |
| [dbo.usp\_up­Blood­Test­Results](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults) |
| [dbo.usp\_up­Blood­Test­Results­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen) |
| [dbo.usp\_up­Client­Flag](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag) |
| [dbo.usp\_up­Client­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen) |
| [dbo.usp\_up­Family](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily) |
| [dbo.usp\_up­Familyto­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilytoProperty) |
| [dbo.usp\_up­Family­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen) |
| [dbo.usp\_up­Occupation](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upOccupation) |
| [dbo.usp\_up­Person](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson) |
| [dbo.usp\_up­Property](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty) |
| [dbo.usp\_up­Questionnaire](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire) |
| [dbo.usp\_up­Questionnaire­Web­Screen](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaireWebScreen) |
| [dbo.usp­Log­Error](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError) |
| [dbo.usp­Print­Error](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError) |

|  |
| --- |
| [dbo].[DELETE\_usp\_­Insert­Personto­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Status­ID | int | 4 |
| @Status­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Status records  -- =============================================  CREATE PROCEDURE [dbo].[DELETE\_usp\_­Insert­Personto­Status] -- usp\_­Insert­Personto­Status  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Status­ID int = NULL,  @Status­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Status( Person­ID, Status­ID, Status­Date )  Values ( @Person­ID, @Status­ID, @Status­Date )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[DELETE\_usp\_­Sl­Count­Clients] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Max\_­Age | int | 4 |
| @Start\_­Date | datetime | 8 |
| @End\_­Date | datetime | 8 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150604  -- Description: procedure returns the number of  -- entries in the persons table where  -- is­Client = 1 filtered by age  -- and Report Dates  -- =============================================  CREATE PROCEDURE [dbo].[DELETE\_usp\_­Sl­Count­Clients]  -- Add the parameters for the stored procedure here  @Max\_­Age int = NULL,  @Start\_­Date datetime = '18000101',  @End\_­Date datetime = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  IF (@End\_­Date IS NULL)  SELECT @End\_­Date = Get­Date();  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int, @Max­Age int;    BEGIN TRY  SELECT @spexecutesql­Str = 'SELECT Total­Clients = count([Person­Id]) from [person] WHERE is­Client = 1'  IF (@Max\_­Age IS NOT NULL)  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [Age] <= @Max­Age';  END  IF (@Start\_­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND ( Created­Date >= @Start­Date OR Modified­Date >= @Start­Date ) '  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Max­Age VARCHAR(50), @Start­Date datetime'  , @Max­Age = @Max\_­Age  , @Start­Date = @Start\_­Date  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[Trans­Proc] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | False |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Pri­Key | int | 4 |
| @Char­Col | char(3) | 3 |

SQL Script

|  |
| --- |
| SET QUOTED\_­IDENTIFIER OFF  GO  CREATE PROCEDURE [dbo].[Trans­Proc] @Pri­Key INT, @Char­Col CHAR(3) AS  BEGIN TRANSACTION In­Proc  INSERT INTO Test­Trans VALUES (@Pri­Key, @Char­Col)  INSERT INTO Test­Trans VALUES (@Pri­Key + 1, @Char­Col)  COMMIT TRANSACTION In­Proc;  GO |

|  |
| --- |
| [dbo].[usp\_­Insert­Access­Agreement] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Access­Purpose­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Access­Agreement­File | varbinary(max) | max |  |
| @Property­ID | int | 4 |  |
| @Inserted­Access­Agreement­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new  -- Access­Agreement records  -- =============================================  -- HISTORY  -- 12/13/2014 modified procedure to accept OUTPUT parameters  CREATE PROCEDURE [dbo].[usp\_­Insert­Access­Agreement] -- usp\_­Insert­Access­Agreement  -- Add the parameters for the stored procedure here  @Access­Purpose­ID int = NULL,  @Notes varchar(3000) = NULL,  @Access­Agreement­File varbinary(max) = NULL,  @Property­ID int = NULL,  @Inserted­Access­Agreement­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Access­Agreement (Access­Purpose­ID, Access­Agreement­File, Property­ID)  Values ( @Access­Purpose­ID, @Access­Agreement­File, @Property­ID);  SELECT @Inserted­Access­Agreement­ID = SCOPE\_­IDENTITY();  IF (@NOTES IS NOT NULL)  BEGIN TRY  INSERT into Access­Agreement­Notes (Access­Agreement­ID, Notes)  Values (@Inserted­Access­Agreement­ID, @Notes)  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  END CATCH;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreement)

[[dbo].[Access­Agreement­Notes]](#(local)/User_databases/LCCHPDev/Tables/AccessAgreementNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Access­Purpose] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Access­Purpose­Name | varchar(50) | 50 |  |
| @Access­Purpose­Description | varchar(250) | 250 |  |
| @Access­Purpose­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Access­Purpose records  -- =============================================  -- HISTORY  -- 12/13/2014 modified procedure to accept OUTPUT parameters  CREATE PROCEDURE [dbo].[usp\_­Insert­Access­Purpose] -- usp\_­Insert­Access­Purpose  -- Add the parameters for the stored procedure here  @Access­Purpose­Name varchar(50) = NULL,  @Access­Purpose­Description varchar(250) = NULL,  @Access­Purpose­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Access­Purpose ( Access­Purpose­Name, Access­Purpose­Description)  Values ( @Access­Purpose­Name, @Access­Purpose­Description);  SELECT @Access­Purpose­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Access­Purpose]](#(local)/User_databases/LCCHPDev/Tables/AccessPurpose)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Area] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Area­Description | varchar(250) | 250 |  |
| @Area­Name | varchar(50) | 50 |  |
| @New­Area­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Area records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Area] -- usp\_­Insert­Area  -- Add the parameters for the stored procedure here  @Area­Description varchar(250) = NULL,  @Area­Name varchar(50) = NULL,  @New­Area­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Area ( Area­Description, Historic­Area­ID)  Values ( @Area­Description, @Area­Name);  SELECT @New­Area­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Area]](#(local)/User_databases/LCCHPDev/Tables/Area)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Blood­Test­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @is­Baseline | bit | 1 |  |
| @Person­ID | int | 4 |  |
| @Sample­Date | date | 3 |  |
| @Lab­Submission­Date | date | 3 |  |
| @Lead­Value | numeric(4,1) | 5 |  |
| @Lead­Value­Category­ID | tinyint | 1 |  |
| @Hemoglobin­Value | numeric(4,1) | 5 |  |
| @Hemoglobin­Value­Category­ID | tinyint | 1 |  |
| @Hematocrit­Value­Category­ID | tinyint | 1 |  |
| @Lab­ID | int | 4 |  |
| @Client­Status­ID | smallint | 2 |  |
| @Blood­Test­Costs | money | 8 |  |
| @sample­Type­ID | tinyint | 1 |  |
| @New\_­Notes | varchar(3000) | 3000 |  |
| @Taken­After­Property­Remediation­Completed | bit | 1 |  |
| @Blood­Test­Result­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Blood­Test­Results records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Blood­Test­Results] -- usp\_­Insert­Blood­Test­Results  -- Add the parameters for the stored procedure here  @is­Baseline bit = NULL,  @Person­ID int = NULL,  @Sample­Date date = NULL,  @Lab­Submission­Date date = NULL,  @Lead­Value numeric(4,1) = NULL,  @Lead­Value­Category­ID tinyint = NULL,  @Hemoglobin­Value numeric(4,1) = NULL,  @Hemoglobin­Value­Category­ID tinyint = NULL, -- lookup in the database  @Hematocrit­Value­Category­ID tinyint = NULL, -- lookup in the database  @Lab­ID int = NULL,  @Client­Status­ID smallint = NULL,  @Blood­Test­Costs money = NULL,  @sample­Type­ID tinyint = NULL,  @New\_­Notes varchar(3000) = NULL,  @Taken­After­Property­Remediation­Completed bit = NULL,  @Blood­Test­Result­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Exists­Person­ID int -- does the person have a record in Blood­Test­Results table  , @Error­Log­ID int, @Notes­ID int;  -- Handle Null sample­Date?  -- Handle Null Lab­Submission­Date?  -- check if the person exists  IF NOT EXISTS (select Person­ID from Person where Person­ID = @Person­ID)  BEGIN  RAISERROR ('Person does not exist. Cannot create a Bloodtest­Result record', 11, -1);  RETURN;  END  -- check if the person has a record in Blood­Test­Results Table  select @Exists­Person­ID = Person­ID from Blood­Test­Results  -- Insert statements for procedure here  BEGIN TRY  -- Determine if this person already has an entry in Blood­Test­Results and set is­Baseline appropriately.  IF ( @is­Baseline is NULL ) -- nothing passed in for baseline  BEGIN  IF ( @Exists­Person­ID is not NULL )  BEGIN  SET @is­Baseline = 0;  END  ELSE -- the person has no entry in Blood­Test­Results, this is a baseline entry  BEGIN  SET @is­Baseline = 1;  END  END  ELSE IF ( @is­Baseline = 0 ) -- this should not be a baseline entry according to passed in argument  BEGIN  IF (@Exists­Person­ID is NULL) -- the person does not have an entry in Blood­Test­Results, this is a baseline entry  BEGIN  Set @is­Baseline = 1;  END  END  ELSE IF ( @is­Baseline = 1 ) -- this should be a baseline entry according to passed in argument  BEGIN  IF (@Exists­Person­ID is not NULL) -- the person already has an entry in Blood­Test­Results, this isn't a baseline entry  BEGIN  Set @is­Baseline = 0;  END  END  INSERT into Blood­Test­Results ( is­Baseline, Person­ID, Sample­Date, Lab­Submission­Date, Lead­Value, Lead­Value­Category­ID,  Hemoglobin­Value, Hemoglobin­Value­Category­ID, Hematocrit­Value­Category­ID, Lab­ID, Client­Status­ID,  Blood­Test­Costs, Sample­Type­ID, Taken­After­Property­Remediation­Completed)  Values ( @is­Baseline, @Person­ID, @Sample­Date, @Lab­Submission­Date, @Lead­Value, @Lead­Value­Category­ID,  @Hemoglobin­Value, @Hemoglobin­Value­Category­ID, @Hematocrit­Value­Category­ID, @Lab­ID, @Client­Status­ID,  @Blood­Test­Costs, @Sample­Type­ID, @Taken­After­Property­Remediation­Completed);  SELECT @Blood­Test­Result­ID = SCOPE\_­IDENTITY();  IF (@New\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Blood­Test­Results­Notes]  @Bloodtest­Results\_­ID = @Blood­Test­Result­ID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp\_­Insert­Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewBloodLeadTestResultsWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Blood­Test­Results­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Blood­Test­Results\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Blood­Test­Results notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Blood­Test­Results­Notes]  -- Add the parameters for the stored procedure here  @Blood­Test­Results\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Blood­Test­Results information  INSERT INTO Blood­Test­Results­Notes (Blood­Test­Results­ID, Notes)  values (@Blood­Test­Results\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_up­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults)

|  |
| --- |
| [dbo].[usp\_­Insert­Cleanup­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Cleanup­Status­Description | varchar(200) | 200 |  |
| @Cleanup­Status­Name | varchar(25) | 25 |  |
| @New­Cleanup­Status­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Cleanup­Status records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Cleanup­Status] -- usp\_­Insert­Cleanup­Status  -- Add the parameters for the stored procedure here  @Cleanup­Status­Description varchar(200) = NULL,  @Cleanup­Status­Name varchar(25) = NULL,  @New­Cleanup­Status­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Cleanup­Status ( Cleanup­Status­Description, Cleanup­Status­Name)  Values ( @Cleanup­Status­Description, @Cleanup­Status­Name);  SELECT @New­Cleanup­Status­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/CleanupStatus)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Construction­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Construction­Type­Description | varchar(250) | 250 |  |
| @Construction­Type­Name | varchar(50) | 50 |  |
| @New­Construction­Type­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Construction­Type records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Construction­Type] -- usp\_­Insert­Construction­Type  -- Add the parameters for the stored procedure here  @Construction­Type­Description varchar(250) = NULL,  @Construction­Type­Name varchar(50) = NULL,  @New­Construction­Type­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Construction­Type ( Construction­Type­Description, Construction­Type­Name)  Values ( @Construction­Type­Description, @Construction­Type­Name);  SELECT @New­Construction­Type­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Construction­Type]](#(local)/User_databases/LCCHPDev/Tables/ConstructionType)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Contractor] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Contractor­Description | varchar(250) | 250 |  |
| @Contractor­Name | varchar(50) | 50 |  |
| @New­Contractor­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Contractor records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Contractor] -- usp\_­Insert­Contractor  -- Add the parameters for the stored procedure here  @Contractor­Description varchar(250) = NULL,  @Contractor­Name varchar(50) = NULL,  @New­Contractor­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Contractor ( Contractor­Description, Contractor­Name)  Values ( @Contractor­Description, @Contractor­Name);  SELECT @New­Contractor­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Contractor]](#(local)/User_databases/LCCHPDev/Tables/Contractor)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Contractorto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Contractor­ID | int | 4 |
| @Property­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Contractorto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Contractorto­Property] -- usp\_­Insert­Contractorto­Property  -- Add the parameters for the stored procedure here  @Contractor­ID int = NULL,  @Property­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Contractorto­Property ( Contractor­ID, Property­ID, Start­Date, End­Date)  Values ( @Contractor­ID, @Property­ID, @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Contractorto­Property]](#(local)/User_databases/LCCHPDev/Tables/ContractortoProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Contractorto­Remediation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Contractor­ID | int | 4 |
| @Remediation­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @is­Sub­Contractor | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Contractorto­Remediation records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Contractorto­Remediation] -- usp\_­Insert­Contractorto­Remediation  -- Add the parameters for the stored procedure here  @Contractor­ID int = NULL,  @Remediation­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @is­Sub­Contractor bit = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Contractorto­Remediation ( Contractor­ID, Remediation­ID, Start­Date, End­Date, is­Sub­Contractor)  Values ( @Contractor­ID, @Remediation­ID, @Start­Date, @End­Date, @is­Sub­Contractor);  SELECT SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Contractor­ID | int | 4 |
| @Remediation­Action­Plan­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @is­Sub­Contractor | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Contractorto­Remediation­Plan records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan] -- usp\_­Insert­Contractorto­Remediation­Plan  -- Add the parameters for the stored procedure here  @Contractor­ID int = NULL,  @Remediation­Action­Plan­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @is­Sub­Contractor bit = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Contractorto­Remediation­Action­Plan ( Contractor­ID, Remediation­Action­Plan­ID, Start­Date, End­Date, is­Sub­Contractor)  Values ( @Contractor­ID, @Remediation­Action­Plan­ID, @Start­Date, @End­Date, @is­Sub­Contractor);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/ContractortoRemediationActionPlan)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Country] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Country­Name | varchar(50) | 50 |  |
| @New­Country­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Country records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Country] -- usp\_­Insert­Country  -- Add the parameters for the stored procedure here  @Country­Name varchar(50) = NULL,  @New­Country­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Country ( Country­Name)  Values ( @Country­Name);  SELECT @New­Country­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Country]](#(local)/User_databases/LCCHPDev/Tables/Country)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Daycare] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Daycare­Name | varchar(50) | 50 |  |
| @Daycare­Description | varchar(200) | 200 |  |
| @new­Day­Care­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Daycare records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Daycare] -- usp\_­Insert­Daycare  -- Add the parameters for the stored procedure here  @Daycare­Name varchar(50) = NULL,  @Daycare­Description varchar(200) = NULL,  @new­Day­Care­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Daycare ( Daycare­Name, Daycare­Description )  Values ( @Daycare­Name, @Daycare­Description );  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Daycare]](#(local)/User_databases/LCCHPDev/Tables/Daycare)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Daycare­Primary­Contact] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Daycare­ID | int | 4 |
| @Person­ID | int | 4 |
| @Contact­Priority | tinyint | 1 |
| @Primary­Phone­Number­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Daycare­Primary­Contact records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Daycare­Primary­Contact] -- usp\_­Insert­Daycare­Primary­Contact  -- Add the parameters for the stored procedure here  @Daycare­ID int = NULL,  @Person­ID int = NULL,  @Contact­Priority tinyint = NULL,  @Primary­Phone­Number­ID int = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Daycare­Primary­Contact ( Day­Care­ID, Person­ID, Contact­Priority, Primary­Phone­Number­ID )  Values ( @Day­Care­ID, @Person­ID, @Contact­Priority, @Primary­Phone­Number­ID );  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Daycare­Primary­Contact]](#(local)/User_databases/LCCHPDev/Tables/DaycarePrimaryContact)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Daycareto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Daycare­ID | int | 4 |
| @Property­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Daycareto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Daycareto­Property] -- usp\_­Insert­Daycareto­Property  -- Add the parameters for the stored procedure here  @Daycare­ID int = NULL,  @Property­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Daycareto­Property ( Daycare­ID, Property­ID, Start­Date, End­Date)  Values ( @Daycare­ID, @Property­ID, @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Daycareto­Property]](#(local)/User_databases/LCCHPDev/Tables/DaycaretoProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Employer] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Employer­Name | varchar(50) | 50 |  |
| @New­Employer­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Employer records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Employer] -- usp\_­Insert­Employer  -- Add the parameters for the stored procedure here  @Employer­Name VARCHAR(50) = NULL,  @New­Employer­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Employer ( Employer­Name )  Values ( @Employer­Name );  SELECT @New­Employer­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Employer]](#(local)/User_databases/LCCHPDev/Tables/Employer)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Employerto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Employer­ID | int | 4 |
| @Property­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Employerto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Employerto­Property] -- usp\_­Insert­Employerto­Property  -- Add the parameters for the stored procedure here  @Employer­ID int = NULL,  @Property­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Employerto­Property ( Employer­ID, Property­ID, Start­Date, End­Date)  Values ( @Employer­ID, @Property­ID, @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Employerto­Property]](#(local)/User_databases/LCCHPDev/Tables/EmployertoProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Environmental­Investigation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Conduct­Environmental­Investigation | bit | 1 |  |
| @Conduct­Environmental­Investigation­Decision­Date | date | 3 |  |
| @Cost | money | 8 |  |
| @Environmental­Investigation­Date | date | 3 |  |
| @Property­ID | int | 4 |  |
| @Start­Date | date | 3 |  |
| @End­Date | date | 3 |  |
| @New­Environmental­Investigation | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Environmental­Investigation records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Environmental­Investigation] -- usp\_­Insert­Environmental­Investigation  -- Add the parameters for the stored procedure here  @Conduct­Environmental­Investigation bit = NULL,  @Conduct­Environmental­Investigation­Decision­Date date = NULL,  @Cost money = NULL,  @Environmental­Investigation­Date date = NULL,  @Property­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @New­Environmental­Investigation int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Environmental­Investigation ( Conduct­Environmental­Investigation, Conduct­Environmental­Investigation­Decision­Date,  Cost, Environmental­Investigation­Date, Property­ID, Start­Date, End­Date )  Values ( @Conduct­Environmental­Investigation, @Conduct­Environmental­Investigation­Decision­Date,  @Cost, @Environmental­Investigation­Date, @Property­ID, @Start­Date, @End­Date );  SELECT @New­Environmental­Investigation = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Tables/EnvironmentalInvestigation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Ethnicity] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Ethnicity | varchar(50) | 50 |  |
| @New­Ethnicity­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Ethnicity records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Ethnicity] -- usp\_­Insert­Ethnicity  -- Add the parameters for the stored procedure here  @Ethnicity varchar(50) = NULL,  @New­Ethnicity­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Ethnicity ( Ethnicity )  Values ( @Ethnicity );  SELECT @New­Ethnicity­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/Ethnicity)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Family] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Last­Name | varchar(50) | 50 |  |
| @Numberof­Smokers | tinyint | 1 |  |
| @Primary­Language­ID | tinyint | 1 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Foreign­Travel | bit | 1 |  |
| @New\_­Travel\_­Notes | varchar(3000) | 3000 |  |
| @Travel\_­Start\_­Date | date | 3 |  |
| @Travel\_­End\_­Date | date | 3 |  |
| @Pets | tinyint | 1 |  |
| @Petsinandout | bit | 1 |  |
| @Primary­Property­ID | int | 4 |  |
| @Frequently­Wash­Pets | bit | 1 |  |
| @FID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140205  -- Description: Stored Procedure to insert new Family information  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Family]  -- Add the parameters for the stored procedure here  @Last­Name varchar(50) = NULL,  @Numberof­Smokers tinyint = 0,  @Primary­Language­ID tinyint = 1,  @Notes varchar(3000) = NULL,  @Foreign­Travel bit = NULL,  @New\_­Travel\_­Notes varchar(3000) = NULL,  @Travel\_­Start\_­Date date = NULL,  @Travel\_­End\_­Date date = NULL,  @Pets tinyint = NULL,  @Petsinandout bit = NULL,  @Primary­Property­ID int = NULL,  @Frequently­Wash­Pets bit = NULL,  @FID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @Error­Log­ID int, @Family­Notes­Return­Value int, @Inserted­Family­Notes­ID int  , @Travel­Notes­Return­Value int, @Inserted­Travel­Notes­ID int;  BEGIN TRY -- insert Family  BEGIN TRANSACTION Insert­Family­Transaction  INSERT into Family ( Last­Name, Numberof­Smokers, Primary­Language­ID, Pets, Petsinandout  , Primary­Property­ID, Frequently­Wash­Pets, Foreign­Travel)  Values (@Last­Name, @Numberof­Smokers, @Primary­Language­ID, @Pets, @Petsinandout  , @Primary­Property­ID, @Frequently­Wash­Pets, @Foreign­Travel)  SET @FID = SCOPE\_­IDENTITY(); -- uncomment to return primary key of inserted values  IF (@Notes IS NOT NULL)  EXEC @Family­Notes­Return­Value = [dbo].[usp\_­Insert­Family­Notes]  @Family\_­ID = @FID,  @Notes = @Notes,  @Inserted­Notes­ID = @Inserted­Family­Notes­ID OUTPUT    IF (@New\_­Travel\_­Notes IS NOT NULL)  EXEC @Travel­Notes­Return­Value = [dbo].[usp\_­Insert­Travel­Notes]  @Family\_­ID = @FID,  @Travel\_­Notes = @New\_­Travel\_­Notes,  @Start\_­Date = @Travel\_­Start\_­Date,  @End\_­Date = @Travel\_­End\_­Date,  @Inserted­Notes­ID = @Inserted­Travel­Notes­ID OUTPUT  COMMIT TRANSACTION Insert­Family­Transaction  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER();  END CATCH;  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[usp\_­Insert­Family­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes)

[[dbo].[usp\_­Insert­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertTravelNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Family­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Family\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150214  -- Description: stored procedure to insert family notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Family­Notes]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Family information  INSERT INTO Family­Notes (Family­ID, Notes)  values (@Family\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER();  END CATCH;  END  GO |

Uses

[[dbo].[Family­Notes]](#(local)/User_databases/LCCHPDev/Tables/FamilyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

[[dbo].[usp\_up­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily)

|  |
| --- |
| [dbo].[usp\_­Insert­Familyto­Phone­Number] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family­ID | int | 4 |
| @Phone­Number­ID | int | 4 |
| @Number­Priority | tinyint | 1 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150404  -- Description: Stored Procedure to insert new  -- Familyto­Phone­Number records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Familyto­Phone­Number]  -- Add the parameters for the stored procedure here  @Family­ID int = NULL,  @Phone­Number­ID int = NULL,  @Number­Priority tinyint = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Existing­Number­Priority tinyint, @Familyto­Phone­Number­Exists bit = 0  , @Family­Phone­Priority­Exists int = 0, @Swap­Priority tinyint;  -- Insert statements for procedure here  BEGIN TRY  -- see if family already has that number  SELECT @Familyto­Phone­Number­Exists = 1, @Swap­Priority = Number­Priority from Familyto­Phone­Number where Family­ID = @Family­ID and Phone­Number­ID = @Phone­Number­ID  -- see if the family already has a number with same priority get the PHone­Number­ID  SELECT @Family­Phone­Priority­Exists = Phone­Number­ID from Familyto­Phone­Number where Family­ID = @Family­ID and Number­Priority = @Number­Priority  -- If the family is already associated with the phone number  IF (@Familyto­Phone­Number­Exists = 1)  BEGIN  -- if the priority is the same do nothing  IF (@Swap­Priority = @Number­Priority)  RETURN;  ELSE IF (@Family­Phone­Priority­Exists = 0) -- there are no numbers for that family with that priority, set the New number priority for the specified number  update Familyto­Phone­Number set Number­Priority = @Number­Priority where Family­ID = @Family­ID and Phone­Number­ID = @Phone­Number­ID  ELSE -- there is another number for that family with the desired priority, swap priorities  BEGIN  -- Set the New number priority for the specified number  update Familyto­Phone­Number set Number­Priority = @Number­Priority where Family­ID = @Family­ID and Phone­Number­ID = @Phone­Number­ID  -- Set number priority for number that had the existing @Number­Priority to the previous priority from the passed in phone number  update Familyto­Phone­Number set Number­Priority = @Swap­Priority where Family­ID = @Family­ID and Phone­Number­ID = @Family­Phone­Priority­Exists  END  END  ELSE -- the family is not associated with that phone number  BEGIN  -- there are no numbers for that family with that priority  IF (@Family­Phone­Priority­Exists = 0)  BEGIN  INSERT into Familyto­Phone­Number( Family­ID, Phone­Number­ID, Number­Priority)  Values ( @Family­ID, @Phone­Number­ID, @Number­Priority )  END  ELSE  BEGIN  -- Insert the New number and priority  INSERT into Familyto­Phone­Number( Family­ID, Phone­Number­ID, Number­Priority)  Values ( @Family­ID, @Phone­Number­ID, @Number­Priority )  -- determine next priority  select @Swap­Priority = max(Number­Priority)+1 from Familyto­Phone­Number where Family­ID = @Family­ID  -- Set number priority for number that had the existing @Number­Priority to the lowest priority  update Familyto­Phone­Number set Number­Priority = @Swap­Priority where Family­ID = @Family­ID and Phone­Number­ID = @Family­Phone­Priority­Exists  END  END  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  THROW  -- RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/FamilytoPhoneNumber)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Familyto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Family­ID | int | 4 |  |
| @Property­ID | int | 4 |  |
| @Property­Link­Type­ID | int | 4 |  |
| @Start­Date | date | 3 |  |
| @End­Date | date | 3 |  |
| @is­Primary­Residence | bit | 1 |  |
| @DEBUG | bit | 1 |  |
| @New­Familyto­Property­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 201504817  -- Description: Stored Procedure to insert new Familyto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Familyto­Property] -- usp\_­Insert­Familyto­Property  -- Add the parameters for the stored procedure here  @Family­ID int = NULL,  @Property­ID int = NULL,  @Property­Link­Type­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @is­Primary­Residence bit = NULL,  @DEBUG bit = NULL,  @New­Familyto­Property­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Familyto­Property­ID int, @update\_return\_value int;  -- Insert statements for procedure here  BEGIN TRY  select @Familyto­Property­ID = Familyto­Property­ID from Familyto­Property where Family­Id = @Family­ID and Property­ID = @Property­ID  IF @Familyto­Property­ID IS NOT NULL  BEGIN  EXEC @update\_return\_value = usp\_up­Familyto­Property  @Familyto­Property­ID = @Familyto­Property­ID,  @Property­Link­Type­ID = @Property­Link­Type­ID,  @Start­Date = @Start­Date,  @End­Date = @End­Date,  @is­Primary­Residence = @is­Primary­Residence,  @DEBUG = @DEBUG  END  ELSE  INSERT into Familyto­Property( Family­ID, Property­ID, Property­Link­Type­ID, Start­Date, End­Date, is­Primary­Residence)  Values ( @Family­ID, @Property­ID, @Property­Link­Type­ID, @Start­Date, @End­Date, @is­Primary­Residence )  --SELECT @New­Familyto­Property­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Familyto­Property]](#(local)/User_databases/LCCHPDev/Tables/FamilytoProperty)

[[dbo].[usp\_up­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilytoProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Foreign­Food] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Foreign­Food­Name | varchar(50) | 50 |  |
| @Foreign­Food­Description | varchar(256) | 256 |  |
| @New­Foreign­Food­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Foreign­Food records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Foreign­Food] -- usp\_­Insert­Foreign­Food  -- Add the parameters for the stored procedure here  @Foreign­Food­Name varchar(50) = NULL,  @Foreign­Food­Description varchar(256) = NULL,  @New­Foreign­Food­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Foreign­Food ( Foreign­Food­Name, Foreign­Food­Description )  Values ( @Foreign­Food­Name, @Foreign­Food­Description );  SELECT @New­Foreign­Food­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER();  END CATCH;  END  GO |

Uses

[[dbo].[Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/ForeignFood)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Foreign­Foodto­Country] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Foreign­Food­ID | int | 4 |
| @Country­ID | tinyint | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Foreign­Foodto­Country records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Foreign­Foodto­Country] -- usp\_­Insert­Foreign­Foodto­Country  -- Add the parameters for the stored procedure here  @Foreign­Food­ID int = NULL,  @Country­ID tinyint = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Foreign­Foodto­Country ( Foreign­Food­ID, Country­ID ) --, Start­Date, End­Date)  Values ( @Foreign­Food­ID, @Country­ID ) -- , @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Tables/ForeignFoodtoCountry)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Gift­Card] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Gift­Card­Value | money | 8 |  |
| @Issue­Date | date | 3 |  |
| @Person­ID | int | 4 |  |
| @New­Gift­Card­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Gift­Card records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Gift­Card] -- usp\_­Insert­Gift­Card  -- Add the parameters for the stored procedure here  @Gift­Card­Value money = NULL,  @Issue­Date date = NULL,  @Person­ID int = NULL,  @New­Gift­Card­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  IF EXISTS (SELECT Person­ID from Person where Person­ID = @Person­ID) print 'Person exists'  INSERT into Gift­Card ( Gift­Card­Value, Issue­Date, Person­ID )  Values ( @Gift­Card­Value, @Issue­Date, @Person­ID );  SELECT @New­Gift­Card­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Gift­Card]](#(local)/User_databases/LCCHPDev/Tables/GiftCard)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Hobby] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Hobby­Name | varchar(50) | 50 |  |
| @Hobby­Description | varchar(256) | 256 |  |
| @Lead­Exposure | bit | 1 |  |
| @New­Hobby­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Hobby records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Hobby] -- usp\_­Insert­Hobby  -- Add the parameters for the stored procedure here  @Hobby­Name varchar(50) = NULL,  @Hobby­Description varchar(256) = NULL,  @Lead­Exposure bit = NULL,  @New­Hobby­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Hobby ( Hobby­Name, Hobby­Description, Lead­Exposure )  Values ( @Hobby­Name, @Hobby­Description, @Lead­Exposure );  SELECT @New­Hobby­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Hobby]](#(local)/User_databases/LCCHPDev/Tables/Hobby)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Home­Remedies] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Home­Remedy­Name | varchar(50) | 50 |  |
| @Home­Remedy­Description | varchar(256) | 256 |  |
| @New­Home­Remedy­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Home­Remedies records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Home­Remedies] -- usp\_­Insert­Home­Remedies  -- Add the parameters for the stored procedure here  @Home­Remedy­Name varchar(50) = NULL,  @Home­Remedy­Description varchar(256) = NULL,  @New­Home­Remedy­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Home­Remedy ( Home­Remedy­Name, Home­Remedy­Description )  Values ( @Home­Remedy­Name, @Home­Remedy­Description );  SELECT @New­Home­Remedy­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Home­Remedy]](#(local)/User_databases/LCCHPDev/Tables/HomeRemedy)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Household­Sourcesof­Lead] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Household­Item­Name | varchar(50) | 50 |  |
| @Household­Item­Description | varchar(512) | 512 |  |
| @New­Household­Item­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Household­Sourcesof­Lead records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Household­Sourcesof­Lead] -- usp\_­Insert­Household­Sourcesof­Lead  -- Add the parameters for the stored procedure here  @Household­Item­Name varchar(50) = NULL,  @Household­Item­Description varchar(512) = NULL,  @New­Household­Item­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Household­Sourcesof­Lead ( Household­Item­Name, Household­Item­Description )  Values ( @Household­Item­Name, @Household­Item­Description );  SELECT @New­Household­Item­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Tables/HouseholdSourcesofLead)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Insurance­Provider] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Insurance­Provider­Name | varchar(50) | 50 |  |
| @New­Insurance­Provider­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Insurance­Provider records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Insurance­Provider] -- usp\_­Insert­Insurance­Provider  -- Add the parameters for the stored procedure here  @Insurance­Provider­Name varchar(50) = NULL,  @New­Insurance­Provider­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Insurance­Provider ( Insurance­Provider­Name ) --, Household­Item­Description )  Values ( @Insurance­Provider­Name ) -- , @Household­Item­Description );  SELECT @New­Insurance­Provider­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Insurance­Provider]](#(local)/User_databases/LCCHPDev/Tables/InsuranceProvider)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Lab] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Lab­Name | varchar(50) | 50 |  |
| @Lab­Description | varchar(250) | 250 |  |
| @New\_­Lab\_­Notes | varchar(3000) | 3000 |  |
| @New­Lab­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Lab records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Lab] -- usp\_­Insert­Lab  -- Add the parameters for the stored procedure here  @Lab­Name varchar(50) = NULL,  @Lab­Description varchar(250) = NULL,  @New\_­Lab\_­Notes varchar(3000) = NULL,  @New­Lab­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Lab ( Lab­Name, Lab­Description )  Values ( @Lab­Name, @Lab­Description );  SELECT @New­Lab­ID = SCOPE\_­IDENTITY();  IF (@New\_­Lab\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Lab­Notes]  @Lab\_­ID = @New­Lab­ID,  @Notes = @New\_­Lab\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Lab]](#(local)/User_databases/LCCHPDev/Tables/Lab)

[[dbo].[usp\_­Insert­Lab­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLabNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Lab­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Lab\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Lab notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Lab­Notes]  -- Add the parameters for the stored procedure here  @Lab\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Lab information  INSERT INTO Lab­Notes (Lab­ID, Notes)  values (@Lab\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Lab­Notes]](#(local)/User_databases/LCCHPDev/Tables/LabNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Lab]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLab)

|  |
| --- |
| [dbo].[usp\_­Insert­Language] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Language­Name | varchar(50) | 50 |  |
| @LANGUAGEID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130506  -- Description: Stored Procedure to insert new Languages  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Language] -- usp\_­Insert­Language "Italian"  -- Add the parameters for the stored procedure here  @Language­Name varchar(50),  @LANGUAGEID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @DBNAME NVARCHAR(128), @Error­Log­ID int;  SET @DBNAME = DB\_­NAME();  BEGIN TRY  if Exists (select Language­Name from language where Language­Name = @Language­Name)  BEGIN  RAISERROR  (N'The language: %s already exists.',  11, -- Severity.  1, -- State.  @Language­Name);  END    INSERT into Language (Language­Name) Values (upper(@Language­Name))  SET @LANGUAGEID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Language]](#(local)/User_databases/LCCHPDev/Tables/Language)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Medium] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Medium­Name | varchar(50) | 50 |  |
| @Medium­Description | varchar(250) | 250 |  |
| @Trigger­Level | int | 4 |  |
| @Trigger­Level­Units­ID | smallint | 2 |  |
| @New­Medium­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Medium records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Medium] -- usp\_­Insert­Medium  -- Add the parameters for the stored procedure here  @Medium­Name varchar(50) = NULL,  @Medium­Description varchar(250) = NULL,  @Trigger­Level int = NULL,  @Trigger­Level­Units­ID smallint = NULL,  @New­Medium­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Medium ( Medium­Name, Medium­Description, Trigger­Level, Trigger­Level­Units­ID )  Values ( @Medium­Name, @Medium­Description, @Trigger­Level, @Trigger­Level­Units­ID );  SELECT @New­Medium­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Medium]](#(local)/User_databases/LCCHPDev/Tables/Medium)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Medium­Sample­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Medium­ID | int | 4 |  |
| @Medium­Sample­Value | numeric(9,4) | 5 |  |
| @Units­ID | smallint | 2 |  |
| @Sample­Level­Category­ID | tinyint | 1 |  |
| @Medium­Sample­Date | date | 3 |  |
| @Lab­ID | int | 4 |  |
| @Lab­Submission­Date | date | 3 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Is­Above­Trigger­Level | bit | 1 |  |
| @New­Medium­Sample­Results­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Medium­Sample­Results records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Medium­Sample­Results] -- usp\_­Insert­Medium­Sample­Results  -- Add the parameters for the stored procedure here  @Medium­ID int = NULL,  @Medium­Sample­Value numeric(9,4) = NULL,  @Units­ID smallint = NULL,  @Sample­Level­Category­ID tinyint = NULL,  @Medium­Sample­Date date = getdate,  @Lab­ID int = NULL,  @Lab­Submission­Date date = getdate,  @Notes varchar(3000) = NULL,  @Is­Above­Trigger­Level bit = NULL,  @New­Medium­Sample­Results­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Triggerlevel­Units­ID smallint, @Trigger­Level numeric(9,4), @Notes­ID int, @Notes\_results int;  -- Insert statements for procedure here    -- See if Value is above Trigger Level - initially assume units are identical  -- Determine Trigger level units and Trigger Level  Select @Trigger­Level = M.Trigger­Level , @Trigger­Level­Units­ID = M.Trigger­Level­Units­ID FROM Medium­Sampleresults AS MSR  JOIN Medium as M on M.Medium­ID = MSR.Medium­ID  JOIN Units AS TLU on M.Trigger­Level­Units­ID = TLU.Units­ID;  -- IF the units are the same,  IF (@Units­ID = @Triggerlevel­Units­ID )  BEGIN  print 'units are identical comparing values'  IF ( @Medium­Sample­Value < @Trigger­Level )  SET @Is­Above­Trigger­Level = 0;  ELSE  SET @Is­Above­Trigger­Level = 1;  END  ELSE  print 'consider converting values to the same units'  BEGIN TRY  INSERT into Medium­Sample­Results ( Medium­ID, Medium­Sample­Value, Units­ID, Sample­Level­Category­ID, Medium­Sample­Date, Lab­ID,  Lab­Submission­Date, Is­Above­Trigger­Level )  Values ( @Medium­ID, @Medium­Sample­Value, @Units­ID, @Sample­Level­Category­ID, @Medium­Sample­Date, @Lab­ID,  @Lab­Submission­Date, @Is­Above­Trigger­Level );  SELECT @New­Medium­Sample­Results­ID = SCOPE\_­IDENTITY();  IF (@Notes IS NOT NULL)  EXEC @Notes\_results = [usp\_­Insert­Medium­Sample­Results­Notes]  @Medium­Sample­Results\_­ID = @New­Medium­Sample­Results­ID,  @Notes = @Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Medium]](#(local)/User_databases/LCCHPDev/Tables/Medium)

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[Units]](#(local)/User_databases/LCCHPDev/Tables/Units)

[[dbo].[usp\_­Insert­Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Medium­Sample­Results­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Medium­Sample­Results\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150417  -- Description: stored procedure to insert Medium­Sample­Results notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Medium­Sample­Results­Notes]  -- Add the parameters for the stored procedure here  @Medium­Sample­Results\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Medium­Sample information  INSERT INTO Medium­Sample­Results­Notes (Medium­Sample­Results­ID, Notes)  values (@Medium­Sample­Results\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

|  |
| --- |
| [dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Sample\_­Date | date | 3 |  |
| @Lab\_­Date | date | 3 |  |
| @Blood\_­Lead\_­Result | numeric(4,1) | 5 |  |
| @Flag | int | 4 |  |
| @Test\_­Type | tinyint | 1 |  |
| @Lab | varchar(50) | 50 |  |
| @Lab\_­ID | int | 4 |  |
| @Child\_­Status\_­Code | smallint | 2 |  |
| @Child\_­Status\_­Date | date | 3 |  |
| @Hemoglobin\_­Value | numeric(4,1) | 5 |  |
| @DEBUG | bit | 1 |  |
| @Blood\_­Test\_­Results\_­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141217  -- Description: stored procedure to insert data retrieved from  -- the Blood Lead Test Results web screen  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Sample\_­Date date = NULL,  @Lab\_­Date date = Null,  @Blood\_­Lead\_­Result numeric(4,1)= NULL, -- Is this Lead value?  @Flag INT = 365, -- flag follow up date  @Test\_­Type tinyint = NULL, -- Sample­Type­ID need to determine if/how new test­Types are created  @Lab varchar(50) = NULL, -- is this necessary i think the lab should be selected from a drop down with the option to add a new lab and an id should be passed?  @Lab\_­ID int = NULL,  @Child\_­Status\_­Code smallint = NULL, -- Status­ID need to determine if/how new status­Codes are created  @Child\_­Status\_­Date date = NULL,  @Hemoglobin\_­Value numeric(4,1) = NULL,  @DEBUG bit = 0,  @Blood\_­Test\_­Results\_­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Blood­Test­Result\_return\_value int, @Retest­Date\_return\_value int  ,@Retest\_­Date date, @Child­Status­Code\_return\_value int, @Error­Log­ID int;  -- set default date if necessary  IF (@Sample\_­Date is null)  BEGIN  set @Sample\_­Date = Get­Date();  RAISERROR ('Need to specify the Sample­Date, setting to today by default', 5, 0);  END    IF (@Person\_­ID IS NULL)  BEGIN  RAISERROR ('Client name must be supplied', 11, -1);  RETURN;  END;  BEGIN TRY  EXEC @Blood­Test­Result\_return\_value = [dbo].[usp\_­Insert­Blood­Test­Results]  @is­Baseline = NULL,  @Person­ID = @Person\_­ID,  @Sample­Date = @Sample\_­Date,  @Lab­Submission­Date = @Lab\_­Date,  @Lead­Value = @Blood\_­Lead\_­Result,  @Lead­Value­Category­ID = NULL,  @Hemoglobin­Value = @Hemoglobin\_­Value,  @Hemoglobin­Value­Category­ID = NULL,  @Hematocrit­Value­Category­ID = NULL,  @Lab­ID = @Lab\_­ID,  @Client­Status­ID = @Child\_­Status\_­Code,  @Blood­Test­Costs = NULL,  @sample­Type­ID = @Test\_­Type,  @New\_­Notes = NULL,  @Taken­After­Property­Remediation­Completed = NULL,  @Blood­Test­Result­ID = @Blood\_­Test\_­Results\_­ID OUTPUT  --IF (@Child\_­Status\_­Code IS NOT NULL)  --BEGIN  -- IF (@Child\_­Status\_­Date IS NULL)  -- SELECT @Child\_­Status\_­Date = Get­Date();  -- IF @DEBUG = 1  -- SELECT '@Child­Status­Code\_return\_value = [dbo].[usp\_­Insert­Personto­Status] @Person­ID = @Person\_­ID, @Status­ID = @Child\_­Status\_­Code, @Status­Date = @Sample\_­Date'  -- ,@Person\_­ID , @Child\_­Status\_­Code, @Sample\_­Date  -- EXEC @Child­Status­Code\_return\_value = [dbo].[usp\_­Insert­Personto­Status]  -- @Person­ID = @Person\_­ID,  -- @Status­ID = @Child\_­Status\_­Code,  -- @Status­Date = @Sample\_­Date  --END  -- set the retest date based on integer value passed in as Flag  SET @Retest\_­Date = DATEADD(dd,@Flag,@Sample\_­Date);  -- update Person table with the new retest date  -- anyone with a blood test is a client  EXEC @Retest­Date\_return\_value = [dbo].[usp\_up­Person]  @Person\_­ID = @Person\_­ID  , @New\_­Retest­Date = @Retest\_­Date  , @New\_­Client­Status­ID = @Child\_­Status\_­Code  , @New\_is­Client = 1;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­New­Client­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Family\_­ID | int | 4 |  |
| @First\_­Name | varchar(50) | 50 |  |
| @Middle\_­Name | varchar(50) | 50 |  |
| @Last\_­Name | varchar(50) | 50 |  |
| @Birth\_­Date | date | 3 |  |
| @Gender\_ | char | 1 |  |
| @Language\_­ID | tinyint | 1 |  |
| @Ethnicity\_­ID | int | 4 |  |
| @Moved\_ | bit | 1 |  |
| @Travel | bit | 1 |  |
| @Travel\_­Notes | varchar(3000) | 3000 |  |
| @Out\_of\_­Site | bit | 1 |  |
| @Hobby\_­ID | smallint | 2 |  |
| @Hobby\_­Notes | varchar(3000) | 3000 |  |
| @Client\_­Notes | varchar(3000) | 3000 |  |
| @Release\_­Notes | varchar(3000) | 3000 |  |
| @is\_­Smoker | bit | 1 |  |
| @Occupation\_­ID | smallint | 2 |  |
| @Occupation\_­Start\_­Date | date | 3 |  |
| @Override­Duplicate­Person | bit | 1 |  |
| @Email­Address | varchar(320) | 320 |  |
| @is\_­Client | bit | 1 |  |
| @Nursing­Mother | bit | 1 |  |
| @Nursing­Infant | bit | 1 |  |
| @Pregnant | bit | 1 |  |
| @Client­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141115  -- Description: stored procedure to insert data from the Add a new client web page  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­New­Client­Web­Screen]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @First\_­Name varchar(50) = NULL,  @Middle\_­Name varchar(50) = NULL,  @Last\_­Name varchar(50) = NULL,  @Birth\_­Date date = NULL,  @Gender\_ char(1) = NULL,  @Language\_­ID tinyint = NULL,  @Ethnicity\_­ID int = NULL,  @Moved\_ bit = NULL,  @Travel bit = NULL, --Foreign­Travel REMOVE AFTE MOVING TO ADDNew­Family­Web­Screen  @Travel\_­Notes varchar(3000) = NULL, -- REMOVE AFTE MOVING TO ADDNew­Family­Web­Screen  @Out\_of\_­Site bit = NULL,  @Hobby\_­ID smallint = NULL,  @Hobby\_­Notes varchar(3000) = NULL,  @Client\_­Notes varchar(3000) = NULL,  @Release\_­Notes varchar(3000) = NULL,  @is\_­Smoker bit = NULL,  @Occupation\_­ID smallint = NULL,  @Occupation\_­Start\_­Date date = NULL,  @Override­Duplicate­Person bit = 0,  @Email­Address VARCHAR(320) = NULL,  @is\_­Client bit = 1,  @Nursing­Mother bit = NULL,  @Nursing­Infant bit = NULL,  @Pregnant bit = NULL,  @Client­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  BEGIN  DECLARE @Error­Log­ID int,  @Ethnicity\_return\_value int,  @Personto­Family\_return\_value int,  @Personto­Language\_return\_value int,  @Personto­Hobby\_return\_value int,  @Personto­Occupation\_return\_value int,  @Personto­Ethnicity\_return\_value int;    -- If no family ID was passed in exit  IF (@Family\_­ID IS NULL)  BEGIN  RAISERROR ('Family name must be supplied', 11, -1);  RETURN;  END;  -- If the family doesn't exist, return an error  IF ((select Family­ID from family where Family­ID = @Family\_­ID) is NULL)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String = 'Unable to associate non-existent family. Family does not exist.'  RAISERROR (@Error­String, 11, -1);  RETURN;  END    if (@Last\_­Name is null)  BEGIN  select @Last\_­Name = Lastname from Family where Family­ID = @Family\_­ID  END  BEGIN TRY -- insert new person  EXEC [dbo].[usp\_­Insert­Person]  @First­Name = @First\_­Name,  @Middle­Name = @Middle\_­Name,  @Last­Name = @Last\_­Name,  @Birth­Date = @Birth\_­Date,  @Gender = @Gender\_,  @Moved = @Moved\_,  @Foreign­Travel = @Travel,  @Email­Address = @Email­Address,  @Outof­Site = @Out\_of\_­Site,  @New\_­Notes = @Client\_­Notes,  @Hobby\_­Notes = @Hobby\_­Notes,  @Release\_­Notes = @Release\_­Notes,  @Travel\_­Notes = @Travel\_­Notes,  @is­Smoker = @is\_­Smoker,  @is­Client = @is\_­Client,  @Nursing­Mother = @Nursing­Mother,  @Nursing­Infant = @Nursing­Infant,  @Pregnant = @Pregnant,  @Override­Duplicate = @Override­Duplicate­Person,  @PID = @CLient­ID OUTPUT;  -- Associate person to Ethnicity  IF (@Ethnicity\_­ID IS NOT NULL)  EXEC @Ethnicity\_return\_value = [dbo].[usp\_­Insert­Personto­Ethnicity]  @Person­ID = @Client­ID,  @Ethnicity­ID = @Ethnicity\_­ID  -- Associate person to family  if (@Family\_­ID is not NULL)  EXEC @Personto­Family\_return\_value = usp\_­Insert­Personto­Family  @Person­ID = @Client­ID, @Family­ID = @Family\_­ID, @OUTPUT = @Personto­Family\_return\_value OUTPUT;  -- Associate person to language  if (@Language\_­ID is not NULL)  EXEC @Personto­Language\_return\_value = usp\_­Insert­Personto­Language  @Language­ID = @Language\_­ID, @Person­ID = @Client­ID, @is­Primary­Language = 1;  -- associate person to Hobby  if (@Hobby\_­ID is not NULL)  EXEC @Personto­Hobby\_return\_value = usp\_­Insert­Personto­Hobby  @Hobby­ID = @Hobby\_­ID, @Person­ID = @Client­ID;  -- associate person to occupation  if (@Occupation\_­ID is not NULL)  EXEC @Personto­Occupation\_return\_value = [dbo].[usp\_­Insert­Personto­Occupation]  @Person­ID = @Client­ID,  @Occupation­ID = @Occupation\_­ID  END TRY  BEGIN CATCH -- insert person  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH; -- insert new person  END  IF (@Family\_­ID is not NULL AND @Personto­Family\_return\_value <> 0)  BEGIN  RAISERROR ('Error associating person to family', 11, -1);  RETURN;  END    IF (@Hobby\_­ID is not NULL AND @Personto­Hobby\_return\_value <> 0)  BEGIN  RAISERROR ('Error associating person to Hobby', 11, -1);  RETURN;  END    IF (@Language\_­ID is not NULL AND @Personto­Language\_return\_value <> 0)  BEGIN  RAISERROR ('Error associating person to language', 11, -1);  RETURN;  END    IF (@Occupation\_­ID is not NULL and @Personto­Occupation\_return\_value <> 0)  BEGIN  RAISERROR ('Error associating person to occupation', 11, -1);  RETURN;  END  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_­Insert­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity)

[[dbo].[usp\_­Insert­Personto­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily)

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_­Insert­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage)

[[dbo].[usp\_­Insert­Personto­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­New­Family­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Family­Last­Name | varchar(50) | 50 |  |
| @Address\_­Line1 | varchar(100) | 100 |  |
| @Address\_­Line2 | varchar(100) | 100 |  |
| @City­Name | varchar(50) | 50 |  |
| @State­Abbr | char(2) | 2 |  |
| @Zip­Code | varchar(10) | 10 |  |
| @Year\_­Built | date | 3 |  |
| @Movein\_­Date | date | 3 |  |
| @Move­Out\_­Date | date | 3 |  |
| @Owner\_id | int | 4 |  |
| @is\_­Owner\_­Occupied | bit | 1 |  |
| @is\_­Residential | bit | 1 |  |
| @has\_­Peeling\_­Chipping\_­Paint | bit | 1 |  |
| @is\_­Rental | bit | 1 |  |
| @Primary­Phone | bigint | 8 |  |
| @Primary­Phone­Priority | tinyint | 1 |  |
| @Secondary­Phone | bigint | 8 |  |
| @Secondary­Phone­Priority | tinyint | 1 |  |
| @Language | tinyint | 1 |  |
| @Num­Smokers | tinyint | 1 |  |
| @Pets | tinyint | 1 |  |
| @Frequently\_­Wash\_­Pets | bit | 1 |  |
| @Petsinandout | bit | 1 |  |
| @Family­Notes | varchar(3000) | 3000 |  |
| @Property­Notes | varchar(3000) | 3000 |  |
| @Travel | bit | 1 |  |
| @Travel\_­Notes | varchar(3000) | 3000 |  |
| @Travel\_­Start\_­Date | varchar(3000) | 3000 |  |
| @Travel\_­End\_­Date | varchar(3000) | 3000 |  |
| @Override­Duplicate­Property | bit | 1 |  |
| @Override­Duplicate­Family­Property­Association | bit | 1 |  |
| @Owner­Contact­Information | varchar(1000) | 1000 |  |
| @DEBUG | bit | 1 |  |
| @Family­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141115  -- Description: stored procedure to insert data from the Add a new family web page  -- =============================================  -- 20150102 Fixed bug with family/property association checking  CREATE PROCEDURE [dbo].[usp\_­Insert­New­Family­Web­Screen]  -- Add the parameters for the stored procedure here  @Family­Last­Name varchar(50) = NULL,  @Address\_­Line1 varchar(100) = NULL,  @Address\_­Line2 varchar(100) = NULL,  @City­Name varchar(50) = NULL,  @State­Abbr char(2) = NULL,  @Zip­Code varchar(10) = NULL,  @Year\_­Built date = NULL,  @Movein\_­Date date = NULL,  @Move­Out\_­Date date = NULL,  @Owner\_id int = NULL,  @is\_­Owner\_­Occupied bit = NULL,  @is\_­Residential bit = NULL,  @has\_­Peeling\_­Chipping\_­Paint bit = NULL,  @is\_­Rental bit = NULL,  @Primary­Phone bigint = NULL,  @Primary­Phone­Priority tinyint = 1,  @Secondary­Phone bigint = NULL,  @Secondary­Phone­Priority tinyint = 2,  @Language tinyint = NULL,  @Num­Smokers tinyint = NULL,  @Pets tinyint = NULL,  @Frequently\_­Wash\_­Pets bit = NULL,  @Petsinandout bit = NULL,  @Family­Notes varchar(3000) = NULL,  @Property­Notes varchar(3000) = NULL,  @Travel bit = NULL,  @Travel\_­Notes varchar(3000) = NULL,  @Travel\_­Start\_­Date varchar(3000) = NULL,  @Travel\_­End\_­Date varchar(3000) = NULL,  @Override­Duplicate­Property bit = 0,  @Override­Duplicate­Family­Property­Association bit = 0,  @Owner­Contact­Information varchar(1000) = NULL,  @DEBUG BIT = 0,  @Family­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;    IF (@Family­Last­Name IS NULL  AND @Address\_­Line1 IS NULL  AND @Address\_­Line2 IS NULL  AND @Primary­Phone IS NULL  AND @Secondary­Phone IS NULL)  BEGIN  RAISERROR ('You must supply at least one of the following: Family name, Street­Number, Street Name, Street Suffix, Apartment number, Primary phone, or Secondary phone', 11, -1);  RETURN;  END;  BEGIN  DECLARE @return\_value int,  @Phone­Type­ID tinyint,  @Family\_return\_value int,  @Prop­ID int, @LID tinyint,  @Inserted­Familyto­Property­ID int,  @Familyto­Property\_return\_value int,  @Primaryphone\_return\_value int,  @Secondaryphone\_return\_value int,  @New­Family­Notes­ID int,  @Travel­Notes­Return­Value int,  @Error­Log­ID int;  BEGIN TRY  -- Insert the property address if it doesn't already exist  -- Check if the property already exists, if it does, return the property­ID  SELECT @Prop­ID = [dbo].udf\_­Does­Property­Exist (  @Address\_­Line1,  @Address\_­Line2,  @City­Name,  @State­Abbr,  @Zip­Code  )  --if (@is\_­Owner\_­Occupied = 1)  -- select @Owner\_id = IDENT\_­CURRENT('Family')+1  select 'Property­ID ' = @Prop­ID;  if ( @Prop­ID is NULL)  BEGIN -- enter property  EXEC [dbo].[usp\_­Insert­Property]  @Address­Line1 = @Address\_­Line1,  @Address­Line2 = @Address\_­Line2,  @City = @City­Name,  @State = @State­Abbr,  @Zipcode = @Zip­Code,  @New\_­Property­Notes = @Property­Notes,  @Year­Built = @Year\_­Built,  @Ownerid = @Owner\_id,  @is­Owner­Occuppied = @is\_­Owner\_­Occupied,  @is­Residential = @is\_­Residential,  @has­Peeling­Chipping­Paint = @has\_­Peeling\_­Chipping\_­Paint,  @is­Rental = @is\_­Rental,  @Override­Duplicate = @Override­Duplicate­Property,  @Owner­Contact­Information = @Owner­Contact­Information,  @Property­ID = @Prop­ID OUTPUT;  END -- enter property  -- Check if Family is already associated with property, if so, skip insert and return warning:  if ((select count(Primaryproperty­ID) from Family where Last­Name = @Family­Last­Name and Primary­Property­ID = @Prop­ID) > 0)  BEGIN  if ( @Override­Duplicate­Family­Property­Association = 1)  BEGIN  -- update address in the future??  RAISERROR ('Family is already associated with that Property', 11, -1);  RETURN;  END  ELSE  BEGIN  RAISERROR ('Family is already associated with that Property', 11, -1);  RETURN;  END;  END  ELSE  BEGIN  EXEC [dbo].[usp\_­Insert­Family]  @Last­Name = @Family­Last­Name,  @Numberof­Smokers = @Num­Smokers,  @Primary­Language­ID = @Language,  @Notes = @Family­Notes,  @Foreign­Travel = @Travel,  @New\_­Travel\_­Notes = @Travel\_­Notes,  @Travel\_­Start\_­Date = @Travel\_­Start\_­Date,  @Travel\_­End\_­Date = @Travel\_­End\_­Date,  @Pets = @Pets,  @Petsinandout = @Petsinandout,  @Frequently­Wash­Pets = @Frequently\_­Wash\_­Pets,  @Primary­Property­ID = @Prop­ID,  @FID = @Family­ID OUTPUT;  END  -- Associate family to property  EXEC @Familyto­Property\_return\_value = [usp\_­Insert­Familyto­Property]  @Family­ID = @Family­ID,  @Property­ID = @Prop­ID,  @Start­Date = @Movein\_­Date,  @End­Date = @Move­Out\_­Date,  @DEBUG = @DEBUG,  @New­Familyto­Property­ID = @Inserted­Familyto­Property­ID OUTPUT  if (@Primary­Phone is not NULL)  BEGIN -- insert Primary Phone  DECLARE @Primary­Phone­Number­ID\_­OUTPUT bigint;  SELECT @Phone­Type­ID = Phone­Number­Type­ID from Phone­Number­Type where Phone­Number­Type­Name = 'Primary Phone';  EXEC @Primaryphone\_return\_value = [dbo].[usp\_­Insert­Phone­Number]  @Phone­Number = @Primary­Phone,  @Phone­Number­Type­ID = @Phone­Type­ID,  @DEBUG = @DEBUG,  @Phone­Number­ID\_­OUTPUT = @Primary­Phone­Number­ID\_­OUTPUT OUTPUT    EXEC [dbo].[usp\_­Insert­Familyto­Phone­Number]  @Family­ID = @Family­ID,  @Number­Priority = @Primary­Phone­Priority,  @Phone­Number­ID = @Primary­Phone­Number­ID\_­OUTPUT,  @DEBUG = @DEBUG  END -- insert Primary Phone  if (@Secondary­Phone is not NULL)  BEGIN -- insert Secondary Phone  DECLARE @Secondary­Phone­Number­ID\_­OUTPUT bigint;  SELECT @Phone­Type­ID = Phone­Number­Type­ID from Phone­Number­Type where Phone­Number­Type­Name = 'Secondary Phone';  EXEC @Secondaryphone\_return\_value = [dbo].[usp\_­Insert­Phone­Number]  @Phone­Number = @Secondary­Phone,  @Phone­Number­Type­ID = @Phone­Type­ID,  @DEBUG = @DEBUG,  @Phone­Number­ID\_­OUTPUT = @Secondary­Phone­Number­ID\_­OUTPUT OUTPUT  EXEC [dbo].[usp\_­Insert­Familyto­Phone­Number]  @Family­ID = @Family­ID,  @Number­Priority = @Secondary­Phone­Priority,  @Phone­Number­ID = @Secondary­Phone­Number­ID\_­OUTPUT,  @DEBUG = @DEBUG    END -- insert Secondary Phone  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType)

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

[[dbo].[usp\_­Insert­Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber)

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

[[dbo].[usp\_­Insert­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

[[dbo].[udf\_­Does­Property­Exist]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DoesPropertyExist)

|  |
| --- |
| [dbo].[usp\_­Insert­New­Questionnaire­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Questionnaire­Date | date | 3 |  |
| @Paint­Peeling | bit | 1 |  |
| @Paint­Date | date | 3 |  |
| @Visit­Remodel | bit | 1 |  |
| @Remodel­Date | date | 3 |  |
| @Vitamins | bit | 1 |  |
| @Hand­Wash | bit | 1 |  |
| @Bottle | bit | 1 |  |
| @Nursing­Mother | bit | 1 |  |
| @Nursing­Infant | bit | 1 |  |
| @Pregnant | bit | 1 |  |
| @Pacifier | bit | 1 |  |
| @Bites­Nails | bit | 1 |  |
| @Eats­Outdoors | bit | 1 |  |
| @Non­Food­In­Mouth | bit | 1 |  |
| @Eats­Non­Food | bit | 1 |  |
| @Sucks­Thumb | bit | 1 |  |
| @Mouthing | bit | 1 |  |
| @Daycare­ID | int | 4 |  |
| @Visits­Old­Homes | bit | 1 |  |
| @Day­Care­Notes | varchar(3000) | 3000 |  |
| @Source | int | 4 |  |
| @Questionnaire­Notes | varchar(3000) | 3000 |  |
| @Hobby1ID | smallint | 2 |  |
| @Hobby2ID | smallint | 2 |  |
| @Hobby3ID | smallint | 2 |  |
| @Hobby­Notes | varchar(3000) | 3000 |  |
| @DEBUG | bit | 1 |  |
| @Questionnaire\_return\_value | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141208  -- Description: stored procedure to insert data  -- from the Lead Research Subject Questionnaire web page  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Questionnaire­Date date = NULL,  @Paint­Peeling bit = NULL,  @Paint­Date date = NULL,  @Visit­Remodel bit = NULL,  @Remodel­Date date = NULL,  @Vitamins bit = NULL,  @Hand­Wash bit = NULL,  @Bottle bit = NULL,  @Nursing­Mother bit = NULL,  @Nursing­Infant bit = NULL,  @Pregnant bit = NULL,  @Pacifier bit = NULL,  @Bites­Nails bit = NULL,  @Eats­Outdoors bit = NULL,  @Non­Food­In­Mouth bit = NULL,  @Eats­Non­Food bit = NULL,  @Sucks­Thumb bit = NULL,  @Mouthing bit = NULL,  @Daycare­ID int = NULL,  @Visits­Old­Homes bit = NULL,  @Day­Care­Notes varchar(3000) = NULL,  @Source int = NULL,  @Questionnaire­Notes varchar(3000) = NULL,  @Hobby1ID smallint = NULL,  @Hobby2ID smallint = NULL,  @Hobby3ID smallint = NULL,  @Hobby­Notes varchar(3000) = NULL,  @DEBUG bit = 0,  @Questionnaire\_return\_value int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  IF (@Paint­Peeling = '') SET @Paint­Peeling = NULL;  DECLARE @Error­Log­ID int, @New\_­Notes varchar(3000), @Personto­Hobby1\_return\_value int  , @Personto­Hobby2\_return\_value int, @Personto­Hobby3\_return\_value int, @Hobby­Notes­ID int;  BEGIN TRY  -- set default date if necessary  IF (@Questionnaire­Date is null)  BEGIN  print 'Need to specify Questionnaire­Date, setting to today by defualt';  set @Questionnaire­Date = Get­Date();  END  IF (@Person\_­ID IS NULL)  BEGIN  RAISERROR ('Client name must be supplied', 11, -1);  RETURN;  END;  -- Client ID must already exist in the database  IF ( (select Person­ID from person where person­ID = @Person\_­ID ) is NULL)  BEGIN  RAISERROR ('Specified Client­ID does not exist', 11, -1);  RETURN;  END  SET @New\_­Notes = concat(@Questionnaire­Notes,' ',@Day­Care­Notes);    EXEC [dbo].[usp\_­Insert­Questionnaire]  @Person­ID = @Person\_­ID,  @Questionnaire­Date = @Questionnaire­Date,  @Questionnaire­Data­Source­ID = @Source,  @Visit­Remodeled­Property = @Visit­Remodel,  @Paint­Date = @Paint­Date,  @Remodel­Property­Date = @Remodel­Date,  @is­Exposedto­Peeling­Paint = @Paint­Peeling,  @is­Taking­Vitamins = @Vitamins,  @Nursing­Mother = @Nursing­Mother,  @Nursing­Infant = @Nursing­Infant,  @Pregnant = @Pregnant,  @is­Using­Pacifier = @Pacifier,  @is­Using­Bottle = @Bottle,  @Bites­Nails = @Bites­Nails,  @Non­Food­Eating = @Eats­Non­Food,  @Non­Foodin­Mouth = @Non­Food­In­Mouth,  @Eat­Outside = @Eats­Outdoors,  @Suckling = @Sucks­Thumb,  @Mouthing = @Mouthing,  @Frequent­Hand­Washing = @Hand­Wash,  @Daycare­ID = @Daycare­ID,  @Visits­Old­Homes = @Visits­Old­Homes,  @New\_­Notes = @New\_­Notes,  @DEBUG = @DEBUG,  @Questionnaire­ID = @Questionnaire\_return\_value OUTPUT  -- Set Nursing­Mother, Nursing­Infant, and Pregnant attributes of the person according to the questionnaire  -- anyone that completes a questionnaire is a client  EXEC [dbo].[usp\_up­Person] @Person\_­ID = @Person\_­ID, @New\_­Nursing­Mother = @Nursing­Mother, @New\_­Nursing­Infant = @Nursing­Infant  , @New\_­Pregnant = @Pregnant, @New\_is­Client = 1, @DEBUG = @DEBUG  -- associate person to Hobby  if (@Hobby1ID is not NULL)  EXEC @Personto­Hobby1\_return\_value = usp\_­Insert­Personto­Hobby  @Hobby­ID = @Hobby1ID, @Person­ID = @Person\_­ID;  if (@Hobby2ID is not NULL)  EXEC @Personto­Hobby2\_return\_value = usp\_­Insert­Personto­Hobby  @Hobby­ID = @Hobby2ID, @Person­ID = @Person\_­ID;  if (@Hobby3ID is not NULL)  EXEC @Personto­Hobby3\_return\_value = usp\_­Insert­Personto­Hobby  @Hobby­ID = @Hobby3ID, @Person­ID = @Person\_­ID;  -- insert hobby notes  IF (@Hobby­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Hobby­Notes]  @Person\_­ID = @Person\_­ID,  @Notes = @Hobby­Notes,  @Inserted­Notes­ID = @Hobby­Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_­Insert­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Occupation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Occupation­Name | varchar(50) | 50 |  |
| @Occupation­Description | varchar(256) | 256 |  |
| @Lead­Exposure | bit | 1 |  |
| @New­Occupation­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Occupation records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Occupation] -- usp\_­Insert­Occupation  -- Add the parameters for the stored procedure here  @Occupation­Name varchar(50) = NULL,  @Occupation­Description varchar(256) = NULL,  @Lead­Exposure bit = NULL,  @New­Occupation­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Occupation ( Occupation­Name, Occupation­Description, Lead­Exposure )  Values ( @Occupation­Name, @Occupation­Description, @Lead­Exposure );  SELECT @New­Occupation­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Occupation]](#(local)/User_databases/LCCHPDev/Tables/Occupation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Person] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @First­Name | varchar(50) | 50 |  |
| @Middle­Name | varchar(50) | 50 |  |
| @Last­Name | varchar(50) | 50 |  |
| @Birth­Date | date | 3 |  |
| @Gender | char | 1 |  |
| @Status­ID | smallint | 2 |  |
| @Foreign­Travel | bit | 1 |  |
| @Outof­Site | bit | 1 |  |
| @Eats­Foreign­Food | bit | 1 |  |
| @Email­Address | varchar(320) | 320 |  |
| @Retest­Date | datetime | 8 |  |
| @Moved | bit | 1 |  |
| @Moved­Date | date | 3 |  |
| @is­Closed | bit | 1 |  |
| @is­Resolved | bit | 1 |  |
| @New\_­Notes | varchar(3000) | 3000 |  |
| @Release\_­Notes | varchar(3000) | 3000 |  |
| @Hobby\_­Notes | varchar(3000) | 3000 |  |
| @Travel\_­Notes | varchar(3000) | 3000 |  |
| @Guardian­ID | int | 4 |  |
| @is­Smoker | bit | 1 |  |
| @is­Client | bit | 1 |  |
| @Nursing­Mother | bit | 1 |  |
| @Nursing­Infant | bit | 1 |  |
| @Pregnant | bit | 1 |  |
| @Override­Duplicate | bit | 1 |  |
| @PID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130506  -- Description: Stored Procedure to insert new people records  -- =============================================  -- DROP PROCEDURE usp\_­Insert­Person  CREATE PROCEDURE [dbo].[usp\_­Insert­Person] -- usp\_­Insert­Person "Bonifacic",'James','Marco','19750205','M'  -- Add the parameters for the stored procedure here  @First­Name varchar(50) = NULL,  @Middle­Name varchar(50) = NULL,  @Last­Name varchar(50) = NULL,  @Birth­Date date = NULL,  @Gender char(1) = NULL,  @Status­ID smallint = NULL,  @Foreign­Travel bit = NULL,  @Outof­Site bit = NULL,  @Eats­Foreign­Food bit = NULL,  @Email­Address VARCHAR(320) = NULL,  @Retest­Date datetime = NULL,  @Moved bit = NULL,  @Moved­Date date = NULL,  @is­Closed bit = 0,  @is­Resolved bit = 0,  @New\_­Notes varchar(3000) = NULL,  @Release\_­Notes varchar(3000) = NULL,  @Hobby\_­Notes varchar(3000) = NULL,  @Travel\_­Notes varchar(3000) = NULL,  @Guardian­ID int = NULL,  @is­Smoker bit = NULL,  @is­Client bit = 1,  @Nursing­Mother bit = 0,  @Nursing­Infant bit = 0,  @Pregnant bit = 0,  @Override­Duplicate bit = 0,  @PID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int;  -- set default retest date if none specified  IF @Retest­Date is null  SET @Retest­Date = DATEADD(yy,1,Get­Date());    Select @PID = Person­ID from Person where Lastname = @Last­Name and First­Name = @First­Name AND Birth­Date = @Birth­Date;  IF (@PID IS NOT NULL AND @Override­Duplicate = 0)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='Person appears to be a duplicate of person­ID: ' + cast(@PID as varchar(256))  RAISERROR (@Error­String, 11, -1);  RETURN;  END  -- Insert statements for procedure here  BEGIN TRY  INSERT into person ( Last­Name, First­Name, Middle­Name, Birth­Date, Gender, Status­ID,  Foreign­Travel, Outof­Site, Eats­Foreign­Food, Email­Address, Retest­Date,  Moved, Moved­Date, is­Closed, is­Resolved, Guardian­ID, is­Smoker,  is­Client, Nursing­Mother, Nursing­Infant, Pregnant)  Values (@Last­Name, @First­Name, @Middle­Name, @Birth­Date, @Gender, @Status­ID,  @Foreign­Travel, @Outof­Site, @Eats­Foreign­Food, @Email­Address, @Retest­Date,  @Moved, @Moved­Date, @is­Closed, @is­Resolved, @Guardian­ID, @is­Smoker,  @is­Client, @Nursing­Mother, @Nursing­Infant, @Pregnant);  SET @PID = SCOPE\_­IDENTITY();  IF (@New\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Notes]  @Person\_­ID = @PID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@Release\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Release­Notes]  @Person\_­ID = @PID,  @Notes = @Release\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@Travel\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Travel­Notes]  @Person\_­ID = @PID,  @Notes = @Travel\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@Hobby\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Hobby­Notes]  @Person\_­ID = @PID,  @Notes = @Hobby\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

[[dbo].[usp\_­Insert­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes)

[[dbo].[usp\_­Insert­Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes)

[[dbo].[usp\_­Insert­Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Person­Hobby­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Person­Hobby notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Person­Hobby­Notes]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Person­Hobby information  INSERT INTO Person­Hobby­Notes (Person­ID, Notes)  values (@Person\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonHobbyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

|  |
| --- |
| [dbo].[usp\_­Insert­Person­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Person notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Person­Notes]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Person information  INSERT INTO Person­Notes (Person­ID, Notes)  values (@Person\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

|  |
| --- |
| [dbo].[usp\_­Insert­Person­Release­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Person­Release notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Person­Release­Notes]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Person­Release information  INSERT INTO Person­Release­Notes (Person­ID, Notes)  values (@Person\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonReleaseNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Access­Agreement] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Access­Agreement­ID | int | 4 |
| @Access­Agreement­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Access­Agreement records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Access­Agreement] -- usp\_­Insert­Personto­Access­Agreement  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Access­Agreement­ID int = NULL,  @Access­Agreement­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Access­Agreement( Person­ID, Access­Agreement­ID, Access­Agreement­Date) --, End­Date)  Values ( @Person­ID, @Access­Agreement­ID, @Access­Agreement­Date ) -- , @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Tables/PersontoAccessAgreement)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Daycare] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Daycare­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Daycare­Notes | varchar(3000) | 3000 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Daycare records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Daycare] -- usp\_­Insert­Personto­Daycare  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Daycare­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @Daycare­Notes varchar(3000) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Daycare( Person­ID, Daycare­ID, Start­Date, End­Date)  Values ( @Person­ID, @Daycare­ID, @Start­Date, @End­Date);  --SELECT SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Daycare]](#(local)/User_databases/LCCHPDev/Tables/PersontoDaycare)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Employer] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Employer­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Employer records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Employer] -- usp\_­Insert­Personto­Employer  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Employer­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Employer( Person­ID, Employer­ID, Start­Date, End­Date)  Values ( @Person­ID, @Employer­ID, @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Employer]](#(local)/User_databases/LCCHPDev/Tables/PersontoEmployer)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Ethnicity] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Ethnicity­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Ethnicity records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Ethnicity] -- usp\_­Insert­Personto­Ethnicity  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Ethnicity­ID int = NULL  --@Start­Date date = NULL,  --@End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  -- only provide support for association with a single ethnicity as per initial scope  IF EXISTS (Select Person­ID from Personto­Ethnicity where Person­ID = @Person­ID)  update Personto­Ethnicity set Ethnicity­ID = @Ethnicity­ID where Person­ID = @Person­ID;  ELSE  INSERT into Personto­Ethnicity( Person­ID, Ethnicity­ID )  Values ( @Person­ID, @Ethnicity­ID )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/PersontoEthnicity)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Family] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person­ID | int | 4 |  |
| @Family­ID | int | 4 |  |
| @OUTPUT | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Family records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Family] -- usp\_­Insert­Personto­Family  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Family­ID int = NULL,  @OUTPUT int OUTPUT  --@Start­Date date = NULL,  --@End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Family( Person­ID, Family­ID ) --, Start­Date, End­Date)  Values ( @Person­ID, @Family­ID ) -- , @Start­Date, @End­Date);  SELECT @OUTPUT = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Family]](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Foreign­Food] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Foreign­Food­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Foreign­Food records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Foreign­Food] -- usp\_­Insert­Personto­Foreign­Food  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Foreign­Food­ID int = NULL  --@Start­Date date = NULL,  --@End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Foreign­Food( Person­ID, Foreign­Food­ID ) --, Start­Date, End­Date)  Values ( @Person­ID, @Foreign­Food­ID ) -- , @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Tables/PersontoForeignFood)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Hobby] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Hobby­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Hobby records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Hobby] -- usp\_­Insert­Personto­Hobby  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Hobby­ID int = NULL  --@Start­Date date = NULL,  --@End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Hobby( Person­ID, Hobby­ID ) --, Start­Date, End­Date)  Values ( @Person­ID, @Hobby­ID ) -- , @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Hobby]](#(local)/User_databases/LCCHPDev/Tables/PersontoHobby)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Home­Remedy] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Home­Remedy­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Home­Remedy records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Home­Remedy] -- usp\_­Insert­Personto­Home­Remedy  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Home­Remedy­ID int = NULL  --@Start­Date date = NULL,  --@End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Home­Remedy( Person­ID, Home­Remedy­ID ) --, Start­Date, End­Date)  Values ( @Person­ID, @Home­Remedy­ID ) -- , @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Tables/PersontoHomeRemedy)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Insurance] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Insurance­ID | smallint | 2 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Group­ID | varchar(20) | 20 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Insurance records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Insurance] -- usp\_­Insert­Personto­Insurance  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Insurance­ID smallint = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @Group­ID varchar(20) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Insurance( Person­ID, Insurance­ID, Start­Date, End­Date, Group­ID)  Values ( @Person­ID, @Insurance­ID, @Start­Date, @End­Date, @Group­ID);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Insurance]](#(local)/User_databases/LCCHPDev/Tables/PersontoInsurance)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Language] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Language­ID | smallint | 2 |
| @is­Primary­Language | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Language records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Language] -- usp\_­Insert­Personto­Language  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Language­ID smallint = NULL,  @is­Primary­Language bit = NULL  --@Start­Date date = NULL,  --@End­Date date = NULL,  --@Group­ID varchar(20) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  IF EXISTS (SELECT Person­ID from Personto­Language where Person­ID = @Person­ID and Language­ID = @Language­ID)  BEGIN  -- make sure there are no other primary languages  IF (@is­Primary­Language = 1)  update Personto­Language set is­Primary­Language = 0 WHERE Person­ID = @Person­ID AND Language­ID != @Language­ID AND is­Primary­Language = 1  update Personto­Language set is­Primary­Language = @is­Primary­Language WHERE Person­ID = @Person­ID AND Language­ID = @Language­ID  END  ELSE  INSERT into Personto­Language( Person­ID, Language­ID, is­Primary­Language ) -- Start­Date, End­Date, Group­ID)  Values ( @Person­ID, @Language­ID, @is­Primary­Language ) -- @Start­Date, @End­Date, @Group­ID);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Language]](#(local)/User_databases/LCCHPDev/Tables/PersontoLanguage)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Occupation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Occupation­ID | smallint | 2 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Occupation records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Occupation] -- usp\_­Insert­Personto­Occupation  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Occupation­ID smallint = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  --@Group­ID varchar(20) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @return\_value int, @Error­Log­ID int;  -- at the very least assume the start date is today  IF (@Start­Date is NULL) SELECT @Start­Date = GETDATE();  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Occupation( Person­ID, Occupation­ID, Start­Date, End­Date)  Values ( @Person­ID, @Occupation­ID, @Start­Date, @End­Date);  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Occupation]](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Person] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person1ID | int | 4 |
| @Person2ID | smallint | 2 |
| @Relationship­Type | int | 4 |
| @is­Guardian | bit | 1 |
| @is­Primary­Contact | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150323  -- Description: Stored Procedure to insert  -- new Personto­Person records how  -- they are related  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Person] -- usp\_­Insert­Personto­Person  -- Add the parameters for the stored procedure here  @Person1ID int = NULL,  @Person2ID smallint = NULL,  @Relationship­Type int = NULL,  @is­Guardian bit = NULL, -- True if P1 is guardian of P2  @is­Primary­Contact bit = NULL  --@End­Date date = NULL,  --@Group­ID varchar(20) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Person( Person1ID, Person2ID, Relationship­Type­ID, is­Guardian, is­Primary­Contact )  Values ( @Person1ID, @Person2ID, @Relation­Ship­Type, @is­Guardian, @is­Primary­Contact )    -- Switch is­Guardian information to update reciprocal relationship  --IF (@is­Guardian = 1) SET @is­Guardian = 0;  --ELSE SET @is­Guardian = 1;    --INSERT into Personto­Person (Person1ID, Person2ID, is­Guardian) values (@Person2ID, @Person1ID, @is­Guardian)  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Person]](#(local)/User_databases/LCCHPDev/Tables/PersontoPerson)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Phone­Number] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Phone­Number­ID | int | 4 |
| @Number­Priority | tinyint | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Phone­Number records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Phone­Number] -- usp\_­Insert­Personto­Phone­Number  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Phone­Number­ID int = NULL,  @Number­Priority tinyint = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Phone­Number( Person­ID, Phone­Number­ID, Number­Priority)  Values ( @Person­ID, @Phone­Number­ID, @Number­Priority )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PersontoPhoneNumber)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person­ID | int | 4 |  |
| @Property­ID | int | 4 |  |
| @Start­Date | date | 3 |  |
| @End­Date | date | 3 |  |
| @is­Primary­Residence | bit | 1 |  |
| @New­Personto­Property­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Property] -- usp\_­Insert­Personto­Property  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Property­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @is­Primary­Residence bit = NULL,  @New­Personto­Property­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Property( Person­ID, Property­ID, Start­Date, End­Date, is­Primary­Residence)  Values ( @Person­ID, @Property­ID, @Start­Date, @End­Date, @is­Primary­Residence )  SELECT @New­Personto­Property­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Property]](#(local)/User_databases/LCCHPDev/Tables/PersontoProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Personto­Travel­Country] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Travel­Country­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Personto­Travel­Country records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Personto­Travel­Country] -- usp\_­Insert­Personto­Travel­Country  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Travel­Country­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Personto­Travel­Country( Person­ID, Country­ID, Start­Date, End­Date )  Values ( @Person­ID, @Travel­Country­ID, @Start­Date, @End­Date )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person­To­Travel­Country]](#(local)/User_databases/LCCHPDev/Tables/PersonToTravelCountry)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Person­Travel­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Person­Travel notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Person­Travel­Notes]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Person­Travel information  INSERT INTO Person­Travel­Notes (Person­ID, Notes)  values (@Person\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Tables/PersonTravelNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

|  |
| --- |
| [dbo].[usp\_­Insert­Phone­Number] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Country­Code | tinyint | 1 |  |
| @Phone­Number | bigint | 8 |  |
| @Phone­Number­Type­ID | tinyint | 1 |  |
| @DEBUG | bit | 1 |  |
| @Phone­Number­ID\_­OUTPUT | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Phone­Number records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Phone­Number] -- usp\_­Insert­Phone­Number  -- Add the parameters for the stored procedure here  @Country­Code tinyint = 1,  @Phone­Number bigint = NULL,  @Phone­Number­Type­ID tinyint = NULL,  @DEBUG bit = NULL,  @Phone­Number­ID\_­OUTPUT int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  -- Determine if the phone number already exists  SELECT @Phone­Number­ID\_­OUTPUT = Phone­Number­ID from PHone­Number where Phone­Number = @Phone­Number  -- If the phone number doesn't exist, insert it and get the new id  IF (@Phone­Number­ID\_­OUTPUT IS NULL)  BEGIN  IF (@DEBUG = 1)  SELECT 'INSERT into Phone­Number ( Country­Code, Phone­Number, Phone­Number­Type­ID )  Values ( @Country­Code, @Phone­Number, @Phone­Number­Type­ID );', @Country­Code, @Phone­Number, @Phone­Number­Type­ID  INSERT into Phone­Number ( Country­Code, Phone­Number, Phone­Number­Type­ID )  Values ( @Country­Code, @Phone­Number, @Phone­Number­Type­ID );  SELECT @Phone­Number­ID\_­OUTPUT = SCOPE\_­IDENTITY();  END  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Phone­Number­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Phone­Number­Type­Name | varchar(50) | 50 |  |
| @Phone­Number­Type­ID\_­OUTPUT | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20141220  -- Description: Stored Procedure to insert new Phone­Number records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Phone­Number­Type] -- usp\_­Insert­Phone­Number­Type  -- Add the parameters for the stored procedure here  @Phone­Number­Type­Name Var­Char(50) = NULL,  @Phone­Number­Type­ID\_­OUTPUT int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;    DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Phone­Number­Type ( Phone­Number­Type­Name )  Values ( @Phone­Number­Type­Name );  SELECT @Phone­Number­Type­ID\_­OUTPUT = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Construction­Type­ID | tinyint | 1 |  |
| @Area­ID | int | 4 |  |
| @isin­Historic­District | bit | 1 |  |
| @is­Remodeled | bit | 1 |  |
| @Remodel­Date | date | 3 |  |
| @isin­City­Limits | bit | 1 |  |
| @Address­Line1 | varchar(100) | 100 |  |
| @Address­Line2 | varchar(100) | 100 |  |
| @City | varchar(50) | 50 |  |
| @State | char(2) | 2 |  |
| @Zipcode | varchar(12) | 12 |  |
| @Year­Built | date | 3 |  |
| @Ownerid | int | 4 |  |
| @is­Owner­Occuppied | bit | 1 |  |
| @Replaced­Pipes­Faucets | tinyint | 1 |  |
| @Total­Remediation­Costs | money | 8 |  |
| @New\_­Property­Notes | varchar(3000) | 3000 |  |
| @is­Residential | bit | 1 |  |
| @is­Currently­Being­Remodeled | bit | 1 |  |
| @has­Peeling­Chipping­Paint | bit | 1 |  |
| @County | varchar(50) | 50 |  |
| @is­Rental | bit | 1 |  |
| @Over­Ride­Duplicate | bit | 1 |  |
| @Owner­Contact­Information | varchar(1000) | 1000 |  |
| @Property­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Property] -- usp\_­Insert­Property  -- Add the parameters for the stored procedure here  @Construction­Type­ID tinyint = NULL,  @Area­ID int = NULL,  @isin­Historic­District bit = NULL,  @is­Remodeled bit = NULL,  @Remodel­Date date = NULL,  @isin­City­Limits bit = NULL,  @Address­Line1 varchar(100) = NULL,  @Address­Line2 varchar(100) = NULL,  @City varchar(50) = NULL,  @State char(2) = NULL,  @Zipcode varchar(12) = NULL,  @Year­Built date = NULL,  @Ownerid int = NULL,  @is­Owner­Occuppied bit = NULL,  @Replaced­Pipes­Faucets tinyint = 0,  @Total­Remediation­Costs money = NULL,  @New\_­Property­Notes varchar(3000) = NULL,  @is­Residential bit = NULL,  @is­Currently­Being­Remodeled bit = NULL,  @has­Peeling­Chipping­Paint bit = NULL,  @County varchar(50) = NULL,  @is­Rental bit = NULL,  @Over­Ride­Duplicate bit = 1,  @Owner­Contact­Information varchar(1000) = NULL,  @Property­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int;  -- Insert statements for procedure here  BEGIN TRY  -- Check if the property already exists, if it does, return the property­ID  SELECT @Property­ID = [dbo].udf\_­Does­Property­Exist (  @Address­Line1,  @Address­Line2,  @City,  @State,  @Zip­Code  )  if (@Property­ID i­S NOT NULL)  BEGIN  if (@Override­Duplicate = 0)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String = 'Property Address ' + @Address­Line1 + ', ' + @City + ', ' + @State + ', ' + @Zip­Code  + ' ' + ' appears to be a duplicate of: ' + cast(@Property­ID as varchar(30));  RAISERROR('@Property­ID exists: @Address­Line1, @City, @State, @Zip­Code', 11, -1);  RETURN;  END  -- RETURN THE Property­ID of the matching property  PRINT 'returning existing property­ID: ' + cast(@Property­ID as varchar);  RETURN;  END  INSERT into property (Construction­Type­ID, Area­ID, isin­Historic­District, is­Remodeled, Remodel­Date,  isin­City­Limits, Address­Line1, Address­Line2, City, [State], Zipcode,  Year­Built, Owner­ID, is­Owner­Occuppied, Replaced­Pipes­Faucets, Total­Remediation­Costs,  is­Residential, is­Currently­Being­Remodeled, has­Peeling­Chipping­Paint, County, is­Rental  , Owner­Contact­Information)  Values ( @Construction­Type­ID, @Area­ID, @isin­Historic­District, @is­Remodeled, @Remodel­Date,  @isin­City­Limits, @Address­Line1, @Address­Line2, @City, @State, @Zipcode,  @Year­Built, @Owner­ID, @is­Owner­Occuppied, @Replaced­Pipes­Faucets, @Total­Remediation­Costs,  @is­Residential, @is­Currently­Being­Remodeled, @has­Peeling­Chipping­Paint, @County, @is­Rental  , @Owner­Contact­Information);  SET @Property­ID = SCOPE\_­IDENTITY();  IF (@New\_­Property­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Property­Notes]  @Property\_­ID = @Property­ID,  @Notes = @New\_­Property­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[usp\_­Insert­Property­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

[[dbo].[udf\_­Does­Property­Exist]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DoesPropertyExist)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Property­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Property\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Property notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Property­Notes]  -- Add the parameters for the stored procedure here  @Property\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Property information  INSERT INTO Property­Notes (Property­ID, Notes)  values (@Property\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Property­Notes]](#(local)/User_databases/LCCHPDev/Tables/PropertyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp\_up­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty)

|  |
| --- |
| [dbo].[usp\_­Insert­Property­Sample­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @is­Baseline | bit | 1 |  |
| @Property­ID | int | 4 |  |
| @Lab­Submission­Date | date | 3 |  |
| @Lab­ID | int | 4 |  |
| @Sample­Type­ID | tinyint | 1 |  |
| @Notes | varchar(3000) | 3000 |  |
| @New­Property­Sample­Results­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Property­Sample­Results records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Property­Sample­Results] -- usp\_­Insert­Property­Sample­Results  -- Add the parameters for the stored procedure here  @is­Baseline bit = NULL,  @Property­ID int = NULL,  @Lab­Submission­Date date = getdate,  @Lab­ID int = NULL,  @Sample­Type­ID tinyint = NULL,  @Notes varchar(3000) = NULL,  @New­Property­Sample­Results­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Exists­Property­ID int, @Notes­ID int, @Notes\_­Results int;  -- check if the property has a record in Blood­Test­Results Table  select @Exists­Property­ID = Property­ID from Property­Sample­Results  -- Insert statements for procedure here  BEGIN TRY  -- Determine if this person already has an entry in Blood­Test­Results and set is­Baseline appropriately.  IF ( @is­Baseline is NULL ) -- nothing passed in for baseline  BEGIN  IF ( @Exists­Property­ID is not NULL )  BEGIN  SET @is­Baseline = 0;  END  ELSE -- the person has no entry in Blood­Test­Results, this is a baseline entry  BEGIN  SET @is­Baseline = 1;  END  END  ELSE IF ( @is­Baseline = 0 ) -- this should not be a baseline entry according to passed in argument  BEGIN  IF (@Exists­Property­ID is NULL) -- the person does not have an entry in Blood­Test­Results, this is a baseline entry  BEGIN  Set @is­Baseline = 1;  END  END  ELSE IF ( @is­Baseline = 1 ) -- this should be a baseline entry according to passed in argument  BEGIN  IF (@Exists­Property­ID is not NULL) -- the person already has an entry in Blood­Test­Results, this isn't a baseline entry  BEGIN  Set @is­Baseline = 0;  END  END  INSERT into Property­Sample­Results ( is­Baseline, Property­ID, Lab­Submission­Date, Lab­ID,  Sample­Type­ID )  Values ( @is­Baseline, @Property­ID, @Lab­Submission­Date, @Lab­ID,  @Sample­Type­ID );  SELECT @New­Property­Sample­Results­ID = SCOPE\_­IDENTITY();  IF (@Notes IS NOT NULL)  EXEC @Notes\_results = [usp\_­Insert­Property­Sample­Results­Notes]  @Property­Sample­Results\_­ID = @New­Property­Sample­Results­ID,  @Notes = @Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults)

[[dbo].[usp\_­Insert­Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Property­Sample­Results­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Property­Sample­Results\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150417  -- Description: stored procedure to insert Property­Sample­Results notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Property­Sample­Results­Notes]  -- Add the parameters for the stored procedure here  @Property­Sample­Results\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Property­Sample information  INSERT INTO Property­Sample­Results­Notes (Property­Sample­Results­ID, Notes)  values (@Property­Sample­Results\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResults)

|  |
| --- |
| [dbo].[usp\_­Insert­Propertyto­Cleanup­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Property­ID | int | 4 |
| @Cleanup­Status­ID | tinyint | 1 |
| @Cleanup­Status­Date | date | 3 |
| @Costof­Cleanup | money | 8 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Propertyto­Cleanup­Status records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Propertyto­Cleanup­Status] -- usp\_­Insert­Propertyto­Cleanup­Status  -- Add the parameters for the stored procedure here  @Property­ID int = NULL,  @Cleanup­Status­ID tinyint = NULL,  @Cleanup­Status­Date date = NULL,  @Costof­Cleanup money = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Propertyto­Cleanup­Status( Property­ID, Cleanup­Status­ID, Cleanup­Status­Date, Costof­Cleanup )  Values ( @Property­ID, @Cleanup­Status­ID, @Cleanup­Status­Date, @Costof­Cleanup )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Tables/PropertytoCleanupStatus)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Property­ID | int | 4 |
| @Household­Sourcesof­Lead­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Propertyto­Household­Sourcesof­Lead records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead] -- usp\_­Insert­Propertyto­Household­Sourcesof­Lead  -- Add the parameters for the stored procedure here  @Property­ID int = NULL,  @Household­Sourcesof­Lead­ID int = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Propertyto­Household­Sourcesof­Lead( Property­ID, Household­Sourcesof­Lead­ID )  Values ( @Property­ID, @Household­Sourcesof­Lead­ID )  SELECT SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Tables/PropertytoHouseholdSourcesofLead)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Propertyto­Medium] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Property­ID | int | 4 |
| @Medium­ID | int | 4 |
| @Medium­Tested | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Propertyto­Medium records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Propertyto­Medium] -- usp\_­Insert­Propertyto­Medium  -- Add the parameters for the stored procedure here  @Property­ID int = NULL,  @Medium­ID int = NULL,  @Medium­Tested bit = 1  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Propertyto­Medium( Property­ID, Medium­ID, Medium­Tested )  Values ( @Property­ID, @Medium­ID, @Medium­Tested )  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Tables/PropertytoMedium)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Questionnaire] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Person­ID | int | 4 |  |
| @Questionnaire­Date | date | 3 |  |
| @Questionnaire­Data­Source­ID | int | 4 |  |
| @Visit­Remodeled­Property | bit | 1 |  |
| @Paint­Date | date | 3 |  |
| @Remodel­Property­Date | date | 3 |  |
| @is­Exposedto­Peeling­Paint | bit | 1 |  |
| @is­Taking­Vitamins | bit | 1 |  |
| @Nursing­Mother | bit | 1 |  |
| @Nursing­Infant | bit | 1 |  |
| @Pregnant | bit | 1 |  |
| @is­Using­Pacifier | bit | 1 |  |
| @is­Using­Bottle | bit | 1 |  |
| @Bites­Nails | bit | 1 |  |
| @Non­Food­Eating | bit | 1 |  |
| @Non­Foodin­Mouth | bit | 1 |  |
| @Eat­Outside | bit | 1 |  |
| @Suckling | bit | 1 |  |
| @Mouthing | bit | 1 |  |
| @Frequent­Hand­Washing | bit | 1 |  |
| @Visits­Old­Homes | bit | 1 |  |
| @Daycare­ID | int | 4 |  |
| @New\_­Notes | varchar(3000) | 3000 |  |
| @DEBUG | bit | 1 |  |
| @Questionnaire­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Questionnaire records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Questionnaire]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Questionnaire­Date date = getdate,  @Questionnaire­Data­Source­ID int = NULL,  @Visit­Remodeled­Property bit = NULL,  @Paint­Date date = NULL,  @Remodel­Property­Date date = NULL,  @is­Exposedto­Peeling­Paint bit = NULL,  @is­Taking­Vitamins bit = NULL,  @Nursing­Mother bit = NULL,  @Nursing­Infant bit = NULL,  @Pregnant bit = NULL,  @is­Using­Pacifier bit = NULL,  @is­Using­Bottle bit = NULL,  @Bites­Nails bit = NULL,  @Non­Food­Eating bit = NULL,  @Non­Foodin­Mouth bit = NULL,  @Eat­Outside bit = NULL,  @Suckling bit = NULL,  @Mouthing bit = NULL,  @Frequent­Hand­Washing bit = NULL,  @Visits­Old­Homes bit = NULL,  @Daycare­ID int = NULL,  @New\_­Notes varchar(3000) = NULL,  @DEBUG bit = NULL,  @Questionnaire­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Questionnaire ( Person­ID, Questionnaire­Date, Questionnaire­Data­Source­ID, Visit­Remodeled­Property, Paint­Date, Remodel­Property­Date,  is­Exposedto­Peeling­Paint, is­Taking­Vitamins, Nursing­Mother, Nursing­Infant, Pregnant, is­Using­Pacifier, is­Using­Bottle,  Bitesnails, Non­Food­Eating, Non­Foodin­Mouth, Eat­Outside, Suckling, Mouthing, Frequent­Hand­Washing,  Visits­Old­Homes, Daycare­ID )  Values ( @Person­ID, @Questionnaire­Date, @Questionnaire­Data­Source­ID, @Visit­Remodeled­Property, @Paint­Date, @Remodel­Property­Date,  @is­Exposedto­Peeling­Paint, @is­Taking­Vitamins, @Nursing­Mother, @Nursing­Infant, @Pregnant, @is­Using­Pacifier, @is­Using­Bottle,  @Bitesnails, @Non­Food­Eating, @Non­Foodin­Mouth, @Eat­Outside, @Suckling, @Mouthing, @Frequent­Hand­Washing,  @Visits­Old­Homes, @Daycare­ID );  SELECT @Questionnaire­ID = SCOPE\_­IDENTITY();  IF (@New\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Questionnaire­Notes]  @Questionnaire\_­ID = @Questionnaire­ID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp\_­Insert­Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

|  |
| --- |
| [dbo].[usp\_­Insert­Questionnaire­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Questionnaire\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to insert Questionnaire notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Questionnaire­Notes]  -- Add the parameters for the stored procedure here  @Questionnaire\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Questionnaire information  INSERT INTO Questionnaire­Notes (Questionnaire­ID, Notes)  values (@Questionnaire\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Tables/QuestionnaireNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire)

[[dbo].[usp\_up­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire)

|  |
| --- |
| [dbo].[usp\_­Insert­Remediation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Remediation­Approval­Date | date | 3 |  |
| @Remediation­Start­Date | date | 3 |  |
| @Remediation­End­Date | date | 3 |  |
| @Property­ID | int | 4 |  |
| @Remediation­Action­Plan­ID | int | 4 |  |
| @Access­Agreement­ID | int | 4 |  |
| @Final­Remediation­Report­File | varbinary(max) | max |  |
| @Final­Remediation­Report­Date | date | 3 |  |
| @Remediation­Cost | money | 8 |  |
| @One­Year­Remediation­Complete­Date | date | 3 |  |
| @Notes | varchar(3000) | 3000 |  |
| @One­Year­Remediatio­NComplete | bit | 1 |  |
| @New­Remediation­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Remediation records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Remediation] -- usp\_­Insert­Remediation  -- Add the parameters for the stored procedure here  @Remediation­Approval­Date date = getdate,  @Remediation­Start­Date date = NULL,  @Remediation­End­Date date = NULL,  @Property­ID int = NULL,  @Remediation­Action­Plan­ID int = NULL,  @Access­Agreement­ID int = NULL,  @Final­Remediation­Report­File varbinary(max) = NULL,  @Final­Remediation­Report­Date date = Null,  @Remediation­Cost money = NULL,  @One­Year­Remediation­Complete­Date date = NULL,  @Notes varchar(3000) = NULL,  @One­Year­Remediatio­NComplete bit = NULL,  @New­Remediation­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int, @Remediation­Notes\_return int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Remediation ( Remediation­Approval­Date, Remediation­Start­Date, Remediation­End­Date, Property­ID  , Remediation­Action­Plan­ID, Access­Agreement­ID, Final­Remediation­Report­File, Final­Remediation­Report­Date  , Remediation­Cost, One­Year­Remediation­Complete­Date, One­Year­Remediation­Complete )  Values ( @Remediation­Approval­Date, @Remediation­Start­Date, @Remediation­End­Date, @Property­ID  , @Remediation­Action­Plan­ID, @Access­Agreement­ID, @Final­Remediation­Report­File, @Final­Remediation­Report­Date  , @Remediation­Cost, @One­Year­Remediation­Complete­Date, @One­Year­Remediation­Complete);  SELECT @New­Remediation­ID = SCOPE\_­IDENTITY();  IF (@Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Remediation­Notes]  @Remediation\_­ID = @New­Remediation­ID,  @Notes = @Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

[[dbo].[usp\_­Insert­Remediation­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Remediation­Action­Plan] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Remediation­Action­Plan­Approval­Date | date | 3 |  |
| @Home­Owner­Consultation­Date | date | 3 |  |
| @Contractor­Completed­Investigation­Date | date | 3 |  |
| @Environmental­Investigation­ID | int | 4 |  |
| @Remediation­Action­Plan­Final­Report­Submission­Date | date | 3 |  |
| @Remediation­Action­Plan­File | varbinary(max) | max |  |
| @Property­ID | int | 4 |  |
| @New­Remediation­Action­Plan­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Remediation­Action­Plan records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Remediation­Action­Plan] -- usp\_­Insert­Remediation­Action­Plan  -- Add the parameters for the stored procedure here  @Remediation­Action­Plan­Approval­Date date = getdate,  @Home­Owner­Consultation­Date date = NULL,  @Contractor­Completed­Investigation­Date date = NULL,  @Environmental­Investigation­ID int = NULL,  @Remediation­Action­Plan­Final­Report­Submission­Date date = NULL,  @Remediation­Action­Plan­File varbinary(max) = NULL,  @Property­ID int = NULL,  @New­Remediation­Action­Plan­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Remediation­Action­Plan ( Remediation­Action­Plan­Approval­Date, Home­Owner­Consultation­Date, Contractor­Completed­Investigation­Date  , Environmental­Investigation­ID, Remediation­Action­Plan­Final­Report­Submission­Date,  Remediation­Action­Plan­File, Property­ID )  Values ( @Remediation­Action­Plan­Approval­Date, @Home­Owner­Consultation­Date, @Contractor­Completed­Investigation­Date  , @Environmental­Investigation­ID, @Remediation­Action­Plan­Final­Report­Submission­Date  , @Remediation­Action­Plan­File, @Property­ID );  SELECT @New­Remediation­Action­Plan­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Remediation­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Remediation\_­ID | int | 4 |  |
| @Notes | varchar(3000) | 3000 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150417  -- Description: stored procedure to insert Remediation notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Remediation­Notes]  -- Add the parameters for the stored procedure here  @Remediation\_­ID int = NULL,  @Notes VARCHAR(3000) = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Remediation information  INSERT INTO Remediation­Notes (Remediation­ID, Notes)  values (@Remediation\_­ID, @Notes);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Remediation­Notes]](#(local)/User_databases/LCCHPDev/Tables/RemediationNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediation)

|  |
| --- |
| [dbo].[usp\_­Insert­Sample­Level­Category] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Sample­Level­Category­Name | varchar(20) | 20 |  |
| @Sample­Level­Category­Description | varchar(256) | 256 |  |
| @New­Sample­Level­Category­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Sample­Level­Category records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Sample­Level­Category] -- usp\_­Insert­Sample­Level­Category  -- Add the parameters for the stored procedure here  @Sample­Level­Category­Name varchar(20) = NULL,  @Sample­Level­Category­Description varchar(256) = NULL,  @New­Sample­Level­Category­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Sample­Level­Category ( Sample­Level­Category­Name, Sample­Level­Category­Description )  Values ( @Sample­Level­Category­Name, @Sample­Level­Category­Description );  SELECT @New­Sample­Level­Category­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Tables/SampleLevelCategory)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Sample­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Sample­Type­Name | varchar(20) | 20 |  |
| @Sample­Type­Description | varchar(256) | 256 |  |
| @New­Sample­Type­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Sample­Type records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Sample­Type] -- usp\_­Insert­Sample­Type  -- Add the parameters for the stored procedure here  @Sample­Type­Name varchar(20) = NULL,  @Sample­Type­Description varchar(256) = NULL,  @New­Sample­Type­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Sample­Type ( Sample­Type­Name, Sample­Type­Description )  Values ( @Sample­Type­Name, @Sample­Type­Description );  SELECT @New­Sample­Type­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Sample­Type]](#(local)/User_databases/LCCHPDev/Tables/SampleType)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Status­Name | varchar(20) | 20 |  |
| @Status­Description | varchar(256) | 256 |  |
| @New­Status­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to insert new Status records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Status] -- usp\_­Insert­Status  -- Add the parameters for the stored procedure here  @Status­Name varchar(20) = NULL,  @Status­Description varchar(256) = NULL,  @New­Status­ID int OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int;  -- Insert statements for procedure here  BEGIN TRY  INSERT into Status ( Status­Name, Status­Description )  Values ( @Status­Name, @Status­Description );  SELECT @New­Status­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Insert­Travel­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Family\_­ID | int | 4 |  |
| @Travel\_­Notes | varchar(3000) | 3000 |  |
| @Start\_­Date | date | 3 |  |
| @End\_­Date | date | 3 |  |
| @Inserted­Notes­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150319  -- Description: stored procedure to insert Travel notes  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Insert­Travel­Notes]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @Travel\_­Notes VARCHAR(3000) = NULL,  @Start\_­Date date = NULL,  @End\_­Date date = NULL,  @Inserted­Notes­ID INT OUTPUT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int  -- Insert statements for procedure here  BEGIN TRY -- update Property information  INSERT INTO Travel­Notes (Family­ID, Notes, Start­Date, End­Date)  values (@Family\_­ID, @Travel\_­Notes, @Start\_­Date, @End\_­Date);  SET @Inserted­Notes­ID = SCOPE\_­IDENTITY();  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  -- inserting information in the Error­Log.  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Travel­Notes]](#(local)/User_databases/LCCHPDev/Tables/TravelNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

|  |
| --- |
| [dbo].[usp\_­SLAll­Blood­Test­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person\_­ID | int | 4 |
| @Min\_­Lead\_­Value | numeric(4,1) | 5 |
| @Max\_­Lead\_­Value | numeric(4,1) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20141222  -- Description: select blood test results  -- optionally only return for a specific  -- client  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLAll­Blood­Test­Results]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Min\_­Lead\_­Value numeric(4,1) = NULL,  @Max\_­Lead\_­Value numeric(4,1) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Order­By NVARCHAR(500),  @Recompile BIT = 1, @Error­Log­ID int;  BEGIN  -- Insert statements for procedure here  SELECT @spexecutesql­Str = N'SELECT ''Client­ID'' = [P].[personid], ''Last­Name'' = [P].[Last­Name], [P].[First­Name], ''Birth­Date'' = [P].[Birth­Date]  , [BTR].[Sample­Date], ''Pb\_ug\_­Per\_dl'' = [BTR].[Lead­Value]  , ''Hb\_g\_­Per\_dl'' = [BTR].[Hemoglobin­Value], ''Retest­BL'' = [P].[Retest­Date]  , ''Closed'' = [P].[is­Closed] , ''Moved'' = [P].[Moved], ''Movedate'' = [P].[Moved­Date]  from [Person] [P]  join [Blood­Test­Results] [BTR] on [P].[Person­ID] = [BTR].[Person­ID]  WHERE [P].[is­Client] = 1'  IF @Person\_­ID IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [p].[Person­ID] = @Person­ID'  SET @Order­By = ' ORDER BY [BTR].[Lead­Value],[BTR].[Sample­Date] desc'  END  IF @Min\_­Lead\_­Value IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] >= @Min­Lead­Value'  END  IF @Max\_­Lead\_­Value IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] < @Max­Lead­Value'  END  IF @Person\_­ID is NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str;  SET @Order­By = N' ORDER BY [p].[Last­Name], [P].[Person­ID] ASC, [BTR].[Sample­Date] DESC';  END  SELECT @spexecutesql­Str = @spexecutesql­Str + @Order­By  IF ( (@Person\_­ID IS NULL) AND (@Min\_­Lead\_­Value IS NULL) )  SET @Recompile = 0;  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  -- If debugging print out query  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Person­ID' = @Person\_­ID, 'Min­Lead­Value' = @Min\_­Lead\_­Value, 'Max­Lead­Value' = @Max\_­Lead\_­Value, 'Recompile' = @Recompile;  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Person­ID int,@Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1)'  , @Person­ID = @Person\_­ID, @Min­Lead­Value = @Min\_­Lead\_­Value, @Max­Lead­Value = @Max\_­Lead\_­Value;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLAll­Blood­Test­Results2] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person\_­ID | int | 4 |
| @Min\_­Lead\_­Value | numeric(4,1) | 5 |
| @Max\_­Lead\_­Value | numeric(4,1) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20141222  -- Description: select blood test results  -- optionally only return for a specific  -- client  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLAll­Blood­Test­Results2]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Min\_­Lead\_­Value numeric(4,1) = NULL,  @Max\_­Lead\_­Value numeric(4,1) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Order­By NVARCHAR(500),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  SET FMTONLY OFF  SELECT @spexecutesql­Str = N'SELECT ''Client­ID'' = [P].[personid], ''Last­Name'' = [P].[Last­Name], ''Birth­Date'' = [P].[Birth­Date]  , [BTR].[Sample­Date], ''Pb\_ug\_­Per\_dl'' = [BTR].[Lead­Value], ''Hb\_g\_per\_dl'' = [BTR].[Hemoglobin­Value], ''Retest­Date'' = [P].[Retest­Date]  , ''Close'' = [P].[is­Closed], ''Moved'' = [P].[Moved], ''Movedate'' = [P].[Moved­Date]  from [Person] [P]  join [Blood­Test­Results] [BTR] on [P].[Person­ID] = [BTR].[Person­ID]  WHERE 1 = 1'  IF @Person\_­ID IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [p].[Person­ID] = @Person­ID'  SET @Order­By = ' ORDER BY [BTR].[Lead­Value],[BTR].[Sample­Date] desc'  END  IF @Min\_­Lead\_­Value IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] >= @Min­Lead­Value'  END  IF @Max\_­Lead\_­Value IS NOT NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] < @Max­Lead­Value'  END  IF @Person\_­ID is NULL  BEGIN  SELECT @spexecutesql­Str = @spexecutesql­Str;  SET @Order­By = N' ORDER BY [BTR].[Leadvalue], [p].[Last­Name], [P].[Person­ID] ASC, [BTR].[Sample­Date] DESC';  END  SELECT @spexecutesql­Str = @spexecutesql­Str + @Order­By  IF ( (@Person\_­ID IS NULL) AND (@Min\_­Lead\_­Value IS NULL) )  SET @Recompile = 0;  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  -- If debugging print out query  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Person­ID' = @Person\_­ID, 'Min­Lead­Value' = @Min\_­Lead\_­Value, 'Max­Lead­Value' = @Max\_­Lead\_­Value, 'Recompile' = @Recompile;  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Person­ID int,@Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1)'  , @Person­ID = @Person\_­ID, @Min­Lead­Value = @Min\_­Lead\_­Value, @Max­Lead­Value = @Max\_­Lead\_­Value;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLAll­Blood­Test­Results­Meta­Data] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person\_­ID | int | 4 |
| @Min\_­Lead\_­Value | numeric(9,4) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20141222  -- Description: select blood test results  -- optionally only return for a specific  -- client  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLAll­Blood­Test­Results­Meta­Data]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Min\_­Lead\_­Value numeric(9,4) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Order­By NVARCHAR(500),  @Recompile BIT = 1, @Error­Log­ID int;  BEGIN  -- Insert statements for procedure here  SELECT 'Client­ID' = [P].[personid], 'Last­Name' = [P].[Last­Name], 'Birth­Date' = [P].[Birth­Date]  , [BTR].[Sample­Date], 'Pb\_ug\_­Per\_dl' = [BTR].[Lead­Value]  , 'Hb\_g\_­Per\_dl' = [BTR].[Hemoglobin­Value], 'Retest­BL' = DATEADD(yy,1,sampledate)  , 'Retest­HB' = DATEADD(yy,1,sampledate), 'Close' = [P].[is­Closed], 'Moved' = [P].[Moved]  , 'Movedate' = [P].[Moved­Date]  from [Person] [P]  join [Blood­Test­Results] [BTR] on [P].[Person­ID] = [BTR].[Person­ID]  WHERE 1 = 0  END  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[usp\_­Sl­Child­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Target­Type | varchar(50) | 50 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: returns valid status codes for passed in type - Child  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Child­Status]  -- Add the parameters for the stored procedure here  @Target­Type varchar(50) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(3000)  select @spexecutesql­Str =''  -- Insert statements for procedure here  SELECT [TS].[Status­Name],[TS].[Status­ID] from [Target­Status] AS [TS]  where 1 = 1 AND Target­Type = 'Person'  END  GO |

Uses

[[dbo].[Target­Status]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus)

|  |
| --- |
| [dbo].[usp\_­Sl­Client­Follow­Up] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150715  -- Description: stored procedure to list family members  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Client­Follow­Up]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'SELECT [P].[Person­ID],[P].[Last­Name],[P].[Firstname],[P].[Retest­Date] from [person] as [p]  where 1=1';  IF (@Start­Date IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [P].[Retest­Date] >= @Start­Date';  IF (@End­Date IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [P].[Retest­Date] < @End­Date';  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ' order by [P].[Retest­Date] ASC'  IF (Date­Diff(yy,@End­Date,@Start­Date) > 4)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Column­Details] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Table­Name | varchar(256) | 256 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141124  -- Description: stored procedure to list column details for each column in a table  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Column­Details]  -- Add the parameters for the stored procedure here  @Table­Name varchar(256) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  SELECT 'Table' = @Table­Name,  c.name 'Column Name',  t.Name 'Data type',  c.max\_length 'Max Length',  c.precision ,  c.scale ,  c.is\_nullable,  ISNULL(i.is\_primary\_key, 0) 'Primary Key'  FROM  sys.columns c  INNER JOIN  sys.types t ON c.user\_type\_id = t.user\_type\_id  LEFT OUTER JOIN  sys.index\_columns ic ON ic.object\_id = c.object\_id AND ic.column\_id = c.column\_id  LEFT OUTER JOIN  sys.indexes i ON ic.object\_id = i.object\_id AND ic.index\_id = i.index\_id  WHERE  c.object\_id = OBJECT\_­ID(@Table­Name)  END  GO |

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Adults] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Min­Age | tinyint | 1 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150610  -- Description: User defined stored procedure to  -- count adults visiting during  -- reporting period  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Adults]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @Min­Age tinyint = 17,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Return­Error int;  BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101'  IF (@ENDDate IS NULL)  SET @End­Date = Get­Date();  -- Create temporary table  CREATE Table #Temp­Potential­Adults  ( Person­ID int  , Test­ID int  , Age­At­Visit tinyint  , Most­Recent­Visit date  , Birthdate date  , Visits tinyint  )  -- insert values from bloodtest results  insert Into #Temp­Potential­Adults (Person­ID, Most­Recent­Visit, Test­ID)  select Person­ID,Most­Recent­Visit = Sample­Date, Test­ID = Blood­Test­Results­ID  from Bloodtest­Results  where Sample­Date >= @Start­Date AND Sample­Date < @End­Date  -- insert values from questionnaire  insert Into #Temp­Potential­Adults (Person­ID, Most­Recent­Visit, Test­ID)  Select Person­ID,Most­Recent­Visit = Questionnaire­Date, Test­ID = Questionnaire­ID  from Questionnaire  where Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  and (ISNULL(Questionnaire.Nursing­Mother,0) = 0 OR ISNULL(Questionnaire.Pregnant,0) = 0 )  -- populate birthdate only if the difference from most recent visit to birthdate is at least min­Age  update #Temp­Potential­Adults set Birth­Date = Person.Birthdate,  Age­At­Visit = [dbo].[udf\_­Calculate­Age]([Person].[Birth­Date],Most­Recent­Visit)  FROM #Temp­Potential­Adults  JOIN Person on Person.Person­ID = #Temp­Potential­Adults.Person­ID  where Datediff(yy,Person.Birth­Date,Most­Recent­Visit) > @Min­Age  Select Adults­Tested = count(distinct Person­ID) from #Temp­Potential­Adults  where Age­At­Visit > @Min­Age  drop table #Temp­Potential­Adults  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  -- DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

[[dbo].[udf\_­Calculate­Age]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_CalculateAge)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Blood­Lead­Levels] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Min­Lead­Value | numeric(4,1) | 5 |
| @Max­Lead­Value | numeric(4,1) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150601  -- Description: procedure returns the number of  -- entries in the persons table  -- with blood test results within  -- the specified date range, and  -- >= 5 and < 10  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Blood­Lead­Levels]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @Min­Lead­Value numeric(4,1) = NULL,  @Max­Lead­Value Numeric(4,1) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  SELECT @spexecutesql­Str = 'SELECT EBLLTests = count([Blood­Test­Results­ID]) from [Blood­Test­Results]  where 1 = 1'  IF (@Min­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Leadvalue >= @Min­Lead­Value'    IF (@Max­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Lead­Value < @Max­Lead­Value'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date >= @Start­Date'  IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date, [Min­Lead­Value] = @Min­Lead­Value, [Max­Lead­Value] = @Max­Lead­Value  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1)'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Min­Lead­Value = @Min­Lead­Value  , @Max­Lead­Value = @Max­Lead­Value  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Blood­Tests] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Min­Lead­Value | numeric(4,1) | 5 |
| @Max­Lead­Value | numeric(4,1) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- blood tests conducted within  -- the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Blood­Tests]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @Min­Lead­Value numeric(4,1) = NULL,  @Max­Lead­Value Numeric(4,1) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  SELECT @spexecutesql­Str = 'SELECT Blood­Tests = count([Blood­Test­Results­ID]) from [Blood­Test­Results]  where 1 = 1'  IF (@Min­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Leadvalue >= @Min­Lead­Value'    IF (@Max­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Lead­Value < @Max­Lead­Value'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date >= @Start­Date'  IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date < @End­Date'  IF ( (DATEDIFF(YYYY,@Start­Date,@End­Date)) > 5)  SET @Recompile = 0  IF ( (@Max­Lead­Value - @Min­Lead­Value) > 5)  SET @Recompile = 0    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date, [Min­Lead­Value] = @Min­Lead­Value, [Max­Lead­Value] = @Max­Lead­Value  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1)'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Min­Lead­Value = @Min­Lead­Value  , @Max­Lead­Value = @Max­Lead­Value  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Clients] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- blood tests conducted within  -- the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Clients]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();  IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  SELECT @spexecutesql­Str = 'Select Clients = count(Person­ID) from (  SELECT Person­ID from blood­Test­Results WHERE 1=1 AND Sample­Date >= @Start­Date AND Sample­Date < @End­Date  UNION  SELECT Person­ID from Questionnaire WHERE 1=1 AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  ) total'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Family­Members] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141125  -- Description: stored procedure to count family members  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Family­Members]  -- Add the parameters for the stored procedure here  @Family­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'SELECT [f].[familyid], Family­Name = [f].[lastname], Members = count([P].[Lastname]) from [Family] AS [F]  LEFT OUTER JOIN [personto­Family] [p2f] on [F].[Family­ID] = [p2F].[Familyid]  LEFT OUTER JOIN [Person] AS [P] on [P].[Personid] = [p2f].[Personid]  where 1=1';  IF (@Family­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [f].[family­ID] = @Family\_­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' group by [f].[familyid],[f].[lastname]  order by [f].[lastname],[f].[familyid]';  IF (@Family­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Family­ID' = @Family­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family\_­ID int'  , @Family\_­ID = @Family­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Home­Visit­Soil­Sample] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150611  -- Description: User defined stored procedure to  -- count clients that have a status  -- of home visit and/or soil sample  -- during the reporting period  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Home­Visit­Soil­Sample]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101'  IF (@ENDDate IS NULL)  SET @End­Date = Get­Date();  select @spexecutesql­Str ='select [Home­Visit­Soil­Samples] = count(Person­ID) from (  SELECT Person­ID  from Blood­Test­Results where Sample­Date >= @Start­Date and Sample­Date < @End­Date  AND Client­Status­ID in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  UNION  -- people with questionnaire but no blood test during reporting period  Select Q.Person­ID  from Questionnaire AS Q  LEFT OUTER JOIN [Blood­Test­Results] AS [BTR] on [BTR].[Blood­Test­Results­ID] = (  select top 1 [Blood­Test­Results­ID] from [Blood­Test­Results]  where [Blood­Test­Results].[Person­ID] = [Q].[Person­ID]  -- AND Sample­Date >= @Start­Date AND Sample­Date < @End­Date  AND BTR.Client­Status­ID  in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  order by Sample­Date desc  )  where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND BTR.Client­Status­ID  in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  ) Home­Visit­Soil­Samples'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Start­Date' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date datetime, @End­Date datetime'  , @Start­Date = @Start­Date  , @End­Date = @End­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  -- DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­New­Clients] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- clients onboarded during the  -- reporting period.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­New­Clients]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();    IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  SELECT @spexecutesql­Str = 'Select New­Clients = count(Person­ID) from Person WHERE is­Client = 1'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date >= @Start­Date'    IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­New­People] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- blood tests conducted within  -- the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­New­People]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();  IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  SELECT @spexecutesql­Str = 'Select New­People = count(Person­ID) from Person WHERE 1=1'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date >= @Start­Date'    IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Nursing­Infants] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150607  -- Description: procedure returns the number of  -- nursing infants that either had a  -- bloodtest or completed a questionnaire  -- within the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Nursing­Infants]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();  IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  SELECT @spexecutesql­Str = 'Select [Nursing­Infants] = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Nursing­Infant from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select TOP 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Nursing­Infant desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Nursing­Infant = 1  UNION  SELECT Person­ID,Nursing­Infant from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Nursing­Infant = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Nursing­Infant = 1'  IF ((Date­Diff(YYYY,@Start­Date,@End­Date) > 5))  SET @Recompile = 0;    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Nursing­Mothers] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150607  -- Description: procedure returns the number of  -- nursing Mothers that either had a  -- bloodtest or completed a questionnaire  -- within the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Nursing­Mothers]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();  IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  SELECT @spexecutesql­Str = 'Select [Nursing­Mothers] = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Nursing­Mother from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select top 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Nursing­Mother desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Nursing­Mother = 1  UNION  SELECT Person­ID,Nursing­Mother from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Nursing­Mother = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Nursing­Mother = 1'  IF ((Date­Diff(YYYY,@Start­Date,@End­Date) > 5))  SET @Recompile = 0;    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date'  , @Start­Date = @Start­Date  , @End­Date = @End­Date    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­People] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Max\_­Age | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 2/13/2014  -- Description: procedure returns the number of entries in the persons table, being the number of participants  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­People]  -- Add the parameters for the stored procedure here  @Max\_­Age int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int, @Max­Age int;    BEGIN TRY  SELECT @spexecutesql­Str = 'SELECT New­People = count([Person­Id]) from [person] WHERE 1=1'  IF (@Max\_­Age IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [Age] <= @Max­Age';    IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date >= @Start­Date';    IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date < @End­Date';  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Max­Age VARCHAR(50),@Start­Date date, @End­Date date'  , @Max­Age = @Max\_­Age  , @Start­Date = @Start­Date  , @End­Date = @End­Date  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­People­By­Age] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141222  -- Description: returns count of people grouped by  -- age. If a lastname is passed in  -- displays a list of people with that lastname  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­People­By­Age]  -- Add the parameters for the stored procedure here  -- @Last\_­Name varchar(50) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1;  -- Insert statements for procedure here  select @spexecutesql­Str ='select Age, ''Personcount'' = count(Person­ID)  from [person]  where is­Client = 1'    -- Return all families and associated properties if nothing was passed in  --IF (@Last\_­Name IS NOT NULL)  -- SELECT @spexecutesql­Str = @spexecutesql­Str + ' and [Last­Name] = @Last­Name'  --ELSE  -- SET @Recompile = 0  -- group people by age  SELECT  @spexecutesql­Str = @spexecutesql­Str + ' group by [dbo].udf\_­Calculate­Age(Birth­Date,Get­Date())'  -- order by age  SELECT  @spexecutesql­Str = @spexecutesql­Str + ' order by Age'  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';    EXEC [sp\_executesql] @spexecutesql­Str  -- , N'@Last­Name varchar(50)'  --, @Last­Name = @Last\_­Name;  END  GO |

|  |
| --- |
| [dbo].[usp\_­Sl­Count­People­By­Age­Group] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150112  -- Description: returns count of people grouped by  -- age categories. If a lastname is passed in  -- displays a list of people with that lastname  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­People­By­Age­Group]  -- Add the parameters for the stored procedure here  -- @Last\_­Name varchar(50) = NULL  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1;  ; with Age­Groups as  ( SELECT CASE  WHEN Age < 1 THEN '0'  WHEN Age >= 1 and Age < 4 THEN '01 - 03'  WHEN Age >= 4 AND Age < 7 THEN '04 - 06'  WHEN Age >= 7 AND Age < 18 THEN '07 - 17'  ELSE '18 and Over'  END AS Groups  -- , Max­Age = max(Person.Age)  FROM Person  where is­Client = 1  )  SELECT ROW\_­NUMBER() OVER(ORDER BY Groups DESC) AS Row, Age­Groups = Coalesce(Groups,'Total'),  Clients = Count(Groups) From Age­Groups group by Groups-- with ROLLUP  -- Insert statements for procedure here  select @spexecutesql­Str ='SELECT Age­Groups = Coalesce(Groups,''Total''),  Clients = Count(Groups) From Age­Groups group by Groups with ROLLUP'  END  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­People­By­Last­Name] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Last\_­Name | varchar(50) | 50 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: <Author,,Name>  -- Create date: <Create Date,,>  -- Description: <Description,,>  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­People­By­Last­Name]  @Last\_­Name VARCHAR(50) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @sp­Executesql­Str NVARCHAR(4000), @Recompile BIT = 1;  BEGIN TRY  SELECT @spexecutesql­Str = 'SELECT [lastname],''Members'' = count([firstname]) from [person] WHERE 1=1';  if (@Last\_­Name is not NULL)  BEGIN  SET @Recompile = 1;  SELECT @sp­Executesql­Str = @sp­Executesql­Str + ' AND [person].[Last­Name] = @Last­Name'  END  ELSE  SET @Recompile = 0  -- Group by last name for counting purposes  SELECT @sp­Executesql­Str = @sp­Executesql­Str + ' group by [lastname]'  -- force recompile for selective query  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  EXEC [sp\_executesql] @sp­Executesql­Str  , N'@Last­Name VARCHAR(50)'  , @Last­Name = @Last\_­Name;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Count­Pregnant­Women] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150610  -- Description: User defined stored procedure to  -- count Pregnant Women  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Count­Pregnant­Women]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101'  IF (@ENDDate IS NULL)  SET @End­Date = Get­Date();  select @spexecutesql­Str ='Select [Pregnant­Women] = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Pregnant from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select top 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Pregnant desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Pregnant = 1  UNION  SELECT Person­ID,Pregnant from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Pregnant = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Pregnant = 1'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Start­Date' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date datetime, @End­Date datetime'  , @Start­Date = @Start­Date  , @End­Date = @End­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  -- DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Daycare] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150319  -- Description: returns daycare name, id, description  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Daycare]  -- Add the parameters for the stored procedure here  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  select Daycare­ID,Daycare­Name,Daycare­Description from Daycare order by Daycare­Name  END  GO |

Uses

[[dbo].[Daycare]](#(local)/User_databases/LCCHPDev/Tables/Daycare)

|  |
| --- |
| [dbo].[usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150618  -- Description: stored procedure to select  -- bloodtestresults edit screen info  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    IF (@Person­ID IS NULL)  BEGIN  RAISERROR ('You must supply a person.', 11, -1);  RETURN;  END;    SELECT @spexecute­SQLStr =  N'select [BTR].[Blood­Test­Results­ID]  ,[BTR].[Sample­Date]  ,[BTR].[Lab­Submissiondate]  ,[L].[Lab­Name]  ,[BTR].[Lead­Value]  ,[Followup­Date] = [P].[Retest­Date]  ,[ST].[Sample­Type­Name]  ,[TS].[Status­Name]  ,[BTR].[Hemoglobin­Value]  from [Blood­Test­Results] AS [BTR]  LEFT OUTER JOIN [Person] AS [P] on [BTR].[Person­ID] = [P].[Person­ID]  LEFT OUTER JOIN [Lab] AS [L] on [BTR].[Lab­ID] = [L].[Lab­ID]  LEFT OUTER JOIN [Sample­Type] AS [ST] on [BTR].[Sample­Type­ID] = [ST].[Sample­Type­ID]  LEFT OUTER JOIN [Target­Status] AS [TS] on [BTR].[Client­Status­ID] = [TS].[Status­ID]  where [BTR].[Person­ID] = @Person­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [BTR].[Sample­Date] desc';    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150408  -- Description: stored procedure to select  -- person edit screen info  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    IF (@Person­ID IS NULL)  BEGIN  RAISERROR ('You must supply a person.', 11, -1);  RETURN;  END;  SELECT @spexecute­SQLStr =  N'select [P].[Person­ID],[P].[Last­Name],[P].[First­Name],[P].[Middle­Name]  ,[P].[Birthdate],[P].[Gender]  ,[P].[is­Client]  ,[L].[Language­ID]  ,[L].[Language­Name]  ,[E].[Ethnicity­ID]  ,[E].[Ethnicity]  ,[P].[Moved]  ,[Moved­Outof­County] = cast([P].[Outof­Site] as varchar)  ,[Travel­V] = cast([P].[Foreign­Travel] as varchar)  ,[P].[Email­Address]  from [Person] AS [P]  LEFT OUTER JOIN [Personto­Language] AS [PL] on [P].[Person­ID] = [PL].[Person­ID]  LEFT OUTER JOIN [Language] AS [L] ON [PL].[Language­ID] = [L].[Language­ID]  LEFT OUTER JOIN [Personto­Ethnicity] AS [PE] ON [PE].[Person­ID] = [P].[Person­ID]  LEFT OUTER JOIN [Ethnicity] AS [E] ON [PE].[Ethnicity­ID] = [E].[Ethnicity­ID]  where [P].[Person­ID] = @Person­ID';    IF EXISTS ( SELECT Person­ID from Personto­Language where Person­ID = @Person­ID )  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [PL].[is­Primary­Language] = 1';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [L].[Created­Date] desc';    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Language]](#(local)/User_databases/LCCHPDev/Tables/PersontoLanguage)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150405  -- Description: returns Family Lastname, Primary Address,  -- Primary phonenumber, Secondary phonenumber,  -- number of smokers, number of pets,  -- if pets are in and out pets,  -- if pets are washed frequently  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information]  -- Add the parameters for the stored procedure here  @Family\_­ID INT = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @Primary­Phone­Number bigint, @Secondary­Phone­Number bigint,  @spexecute­SQLStr NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    IF (@Family\_­ID IS NULL)  BEGIN  RAISERROR ('You must supply the Family.', 11, -1);  RETURN;  END;    -- Select Primary Phone number  select @Primary­Phone­Number = dbo.udf\_­Sl­Family­Phone­Number(@Family\_­ID, 1)  -- Select Secondary Phone number  select @Secondary­Phone­Number = dbo.udf\_­Sl­Family­Phone­Number(@Family\_­ID, 2)    SELECT @spexecute­SQLStr =  N'SELECT [F].[Family­ID],[F].[Lastname],[P].[Address­Line1],[P].[Address­Line2]  ,[P].[City],[P].[State],[P].[Zip­Code],Year­Built = cast([P].[Year­Built] as date)  ,Movein­Date = cast(Start­Date as date), Moveout­Date = cast(End­Date as date)  , [Owner­Occupied] = cast([P].[is­Owner­Occuppied] as varchar)  , Primary­Phone­Number = @Primary­Phone­Number, Secondary­Phone­Number = @Secondary­Phone­Number  ,[F].[Numberof­Smokers],[F].[Pets],Petsinandout = cast([F].[Petsinandout] as varchar)  , [P].[Owner­Contact­Information]  FROM [Family] AS [F]  JOIN [Property] AS [P] ON [F].[Primary­Property­ID] = [P].[Property­ID]  JOIN [Familyto­Property] AS [F2P] ON [F].[Family­ID] = [F2P].[Family­ID] AND [F].[Primary­Property­ID] = [F2P].[Property­ID]  WHERE 1 = 1'  IF (@Family\_­ID IS NULL)  SET @Recompile = 0;  IF (@Family\_­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ' and [F].[Family­ID] = @Family­ID ORDER by [F].[Family­ID] desc'  IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Family­ID' = @Family\_­ID, 'Primary­Phone­Number' = @Primary­Phone­Number, 'Secondary­Phone­Number' = @Secondary­Phone­Number;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family­ID int, @Primary­Phone­Number bigint, @Secondary­Phone­Number bigint'  , @Family­ID = @Family\_­ID  , @Primary­Phone­Number = @Primary­Phone­Number  , @Secondary­Phone­Number = @Secondary­Phone­Number;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

[[dbo].[udf\_­Sl­Family­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_SlFamilyPhoneNumber)

|  |
| --- |
| [dbo].[usp\_­Sl­Edit­Property­Web­Screen­Information] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Property\_­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150405  -- Description: returns Address­Line1, Addressline2  -- City, State, and Zipcode  -- of a specific property  -- if no property ID is passed in,  -- informatin is returned for all properties  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Edit­Property­Web­Screen­Information]  -- Add the parameters for the stored procedure here  @Property\_­ID INT = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @spexecute­SQLStr NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    SELECT @spexecute­SQLStr =  N'SELECT [P].[Property­ID],[P].[Address­Line1],[P].[Address­Line2]  ,[P].[City],[P].[State],[P].[Zip­Code]  FROM [Property] AS [P]  WHERE 1 = 1'  IF (@Property\_­ID IS NULL)  SET @Recompile = 0;  IF (@Property\_­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ' and Property­ID = @Property­ID ORDER by Property­ID desc'  IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Property­ID' = @Property\_­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Property­ID int'  , @Property­ID = @Property\_­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Edit­Questionnaire­Web­Screen­Information] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150618  -- Description: stored procedure to select  -- questionnaire edit screen info  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Edit­Questionnaire­Web­Screen­Information]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    IF (@Person­ID IS NULL)  BEGIN  RAISERROR ('You must supply a person.', 11, -1);  RETURN;  END;    SELECT @spexecute­SQLStr =  N'select [Q].[Questionnaire­ID]  , [Q].[Questionnaire­Date]  , [Q].[is­Exposedto­Peeling­Paint]  , [Q].[Paint­Date]  , [Q].[Visit­Remodeled­Property]  , [Q].[Visits­Old­HOmes]  , [Q].[Remodel­Property­Date]  , [Q].[is­Taking­Vitamins]  , [Q].[Frequent­Hand­Washing]  , [Q].[is­Using­Bottle]  , [Q].[Nursing­Mother]  , [Q].[Pregnant]  , [Q].[Nursing­Infant]  , [Q].[is­Using­Pacifier]  , [Q].[Bites­Nails]  , [Q].[Eat­Outside]  , [Q].[Non­Foodin­Mouth]  , [Q].[Non­Food­Eating]  , [Q].[Suckling]  , [Q].[Mouthing]  from [Questionnaire] AS [Q]  where [Q].[Person­ID] = @Person­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [Q].[Questionnaire­Date] desc';    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Family­Members] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to list family members  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Family­Members]  -- Add the parameters for the stored procedure here  @Family­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'SELECT [f].[familyid], Family­Name = [f].[lastname], [P].[Last­Name], [P].[First­Name] from [Family] AS [F]  LEFT OUTER JOIN [personto­Family] [p2f] on [F].[Family­ID] = [p2F].[Familyid]  LEFT OUTER JOIN [Person] AS [P] on [P].[Personid] = [p2f].[Personid]  where 1=1';  IF (@Family­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [f].[family­ID] = @Family\_­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [f].[lastname],[f].[familyid]';  IF (@Family­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Family­ID' = @Family­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family\_­ID int'  , @Family\_­ID = @Family­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Family­Nameto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­Name | varchar(50) | 50 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20141123  -- Description: User defined stored procedure to  -- select family and property address  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Family­Nameto­Property]  -- Add the parameters for the stored procedure here  @Family\_­Name varchar(50) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  select @spexecutesql­Str ='SELECT ''Family­Name'' = [F].[Last­Name],[Prop].[Street­Number],[Prop].[Street],[Prop].[Street­Suffix],[Prop].[Zip­Code]  from [family] AS [F]  join [Property] as [Prop] on [F].[Primary­Property­ID] = [Prop].[Property­ID]  where 1 = 1'    -- Return all families and associated properties if nothing was passed in  IF (@Family\_­Name IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' and [F].[Last­Name] = @Family­Name'  ELSE  SET @Recompile = 0  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [F].[Last­Name]'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Family­Name varchar(50)'  , @Family­Name = @Family\_­Name;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Hobby] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150620  -- Description: returns hobby name, id, description  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Hobby]  -- Add the parameters for the stored procedure here  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  select Hobby­ID,Hobby­Name,Hobby­Description from Hobby order by Hobby­Name  END  GO |

Uses

[[dbo].[Hobby]](#(local)/User_databases/LCCHPDev/Tables/Hobby)

|  |
| --- |
| [dbo].[usp\_­SLInserted­Data] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Last\_­Name | varchar(50) | 50 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130509  -- Description:  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLInserted­Data]  -- Add the parameters for the stored procedure here  @Last\_­Name varchar(50) = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  SELECT @spexecutesql­Str = N'SELECT [P].[Person­ID]  , ''Family­Last­Name'' = [F].[Lastname]  , [P].[Last­Name]  , [P].[Middle­Name]  , [P].[First­Name]  , [P].[Birth­Date]  , [P].[Gender]  , ''Street­Address'' = cast([Prop].[Street­Number] as varchar)  + '' ''+ cast([Prop].[Street] as varchar) + '' ''  + cast([Prop].[Street­Suffix] as varchar)  , [Prop].[Apartment­Number]  , [Prop].[City]  , [Prop].[State]  , [Prop].[Zipcode]  , ''Primary­Phone­Number'' = [Ph].[Phone­Number]  , [L].[Language­Name]  , [F].[Numberof­Smokers]  , [F].[Pets]  , [F].[inandout]  , [F].[Notes]  , [P].[Moved]  , [P].[Foreign­Travel]  , [P].[Outof­Site]  , [H].[Hobby­Name]  , [P].[Notes]  , [P].[is­Smoker]  , [P].[Retest­Date]  , [Q].[Questionnaire­Date]  , [Q].[is­Exposedto­Peeling­Paint]  , ''Paint­Age'' = [Q].[Remodeled­Property­Age]  , [Q].[Visit­Remodeled­Property]  , ''Remodel­Property­Age'' = [Q].[Remodeled­Property­Age]  , [Q].[is­Taking­Vitamins]  , [Q].[Frequent­Hand­Washing]  , [Q].[is­Using­Bottle]  , [Q].[is­Nursing]  , [Q].[is­Using­Pacifier]  , [Q].[Bites­Nails]  , [Q].[Eat­Outside]  , [Q].[Non­Foodin­Mouth]  , [Q].[Non­Food­Eating]  , [Q].[Suckling]  , [Q].[Daycare]  , [Q].[Source]  , [Q].[Notes]  , [BTR].[Sample­Date]  , [BTR].[Lab­Submission­Date]  , [Lab].[Lab­Name]  , ''What is status code?''  , [BTR].[Hemoglobin­Value]  FROM [Lead­Tracking­Testing-Liam].[dbo].[Person] AS [P]  LEFT OUTER JOIN [Personto­Family] as [P2F] on [P].[Person­ID] = [P2F].[Person­ID]  LEFT OUTER JOIN [Family] AS [F] on [F].[Family­ID] = [P2F].[Family­ID]  LEFT OUTER JOIN [Personto­Property] as [P2P] on [P].Person­ID = [P2P].[Person­ID]  LEFT OUTER JOIN [Questionnaire] as [Q] on [P].[Person­ID] = [Q].[Person­ID]  LEFT OUTER JOIN [Blood­Test­Results] as [BTR] on [P].[Person­ID] = [BTR].[Person­ID]  LEFT OUTER JOIN [Personto­Language] as [P2L] on [P2L].[Person­ID] = [P].[Person­ID]  LEFT OUTER JOIN [Language] as [L] on [L].Language­ID = [P2L].[Language­ID]  LEFT OUTER JOIN [Property] as [Prop] on [Prop].[Property­ID] = [F].[Primary­Property­ID]  LEFT OUTER JOIN [Personto­Phone­Number] as [P2Ph] on [P].[Person­ID] = [P2Ph].[Person­ID]  LEFT OUTER JOIN [Phone­Number] as [Ph] on [Ph].[Phone­Number­ID] = [P2Ph].[Phone­Number­ID]  LEFT OUTER JOIN [Phone­Number­Type] as [Ph­T] on [Ph].[Phone­Number­Type­ID] = [Ph­T].[Phone­Number­Type­ID]  LEFT OUTER JOIN [Personto­Hobby] as [P2H] on [P].Person­ID = [P2H].[Hobby­ID]  LEFT OUTER JOIN [Hobby] as [H] on [H].[Hobby­ID] = [P2H].[Hobby­ID]  LEFT OUTER JOIN [Lab] on [BTR].[Lab­ID] = [Lab].[Lab­ID]  WHERE 1 = 1'  if @Last\_­Name IS NOT NULL  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [p].[Last­Name] = @Last­Name ORDER BY [P].[Person­ID] desc'  ELSE  SELECT @spexecutesql­Str = @spexecutesql­Str + N' ORDER BY [P].[Person­ID] desc'  IF @Last\_name is NULL  SET @Recompile = 0;  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Lastname varchar(50)'  , @Last­Name = @Last\_name;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLInserted­Data­Simplified] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130509  -- Description:  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLInserted­Data­Simplified]  -- Add the parameters for the stored procedure here    AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000),  @Recompile BIT = 1, @Error­Log­ID int  , @DEBUG BIT = 0;  -- Insert statements for procedure here  BEGIN TRY  SELECT [P].[Person­ID]  , 'P2FPerson­ID' = [P2F].[Person­ID]  , 'Family­Last­Name' = [F].[Lastname]  , [F].[Family­ID]  , 'P2FFamily­ID' = [P2F].[Family­ID]  , [P].[Last­Name]  , [P].[Middle­Name]  , [P].[First­Name]  , [P].[Birth­Date]  , [P].[Gender]  --, 'Street­Address' = cast([Prop].[Street­Number] as varchar)  -- + ' '+ cast([Prop].[Street] as varchar) + ' '  -- + cast([Prop].[Street­Suffix] as varchar)  --, [Prop].[Apartment­Number]  --, [Prop].[City]  --, [Prop].[State]  --, [Prop].[Zipcode]  --, 'Primary­Phone­Number' = [Ph].[Phone­Number]  --, [L].[Language­Name]  , [F].[Numberof­Smokers]  , [F].[Pets]  , [F].[Petsinandout]  , [FN].[Notes]  FROM [Person] AS [P]  FULL OUTER JOIN [Personto­Family] as [P2F] on [P].[Person­ID] = [P2F].[Person­ID]  FULL OUTER JOIN [Family] AS [F] on [F].[Family­ID] = [P2F].[Family­ID]  FULL OUTER JOIN [Family­Notes] AS [FN] on [F].[Family­ID] = [FN].[Family­ID]  -- FULL OUTER JOIN [Personto­Property] as [P2P] on [P].Person­ID = [P2P].[Person­ID]  -- FULL OUTER JOIN [Property] as [Prop] on [Prop].[Property­ID] = [F].[Primary­Property­ID]  -- where [P2F].Family­ID is NULL  -- People to families: 3470    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Family­Notes]](#(local)/User_databases/LCCHPDev/Tables/FamilyNotes)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Personto­Family]](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Lab­Name] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150304  -- Description: Lists lab names  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Lab­Name]  -- Add the parameters for the stored procedure here  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  select Lab­Name from Lab where Lab­Name in ('Lead­Care II','Tamarac','Quest Diagnostic','Other')  END  GO |

Uses

[[dbo].[Lab]](#(local)/User_databases/LCCHPDev/Tables/Lab)

|  |
| --- |
| [dbo].[usp\_­SLList­All­Family­Members] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150103  -- Description: stored procedure to list family members  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLList­All­Family­Members]  -- Add the parameters for the stored procedure here  @Family­ID int = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'SELECT [f].[familyid], Family­Name = [f].[lastname],[P].[Last­Name],[P].[Firstname] from [person] as [p]  join [personto­Family] [p2f] on [p].[personid] = [p2f].[personid]  join [family] AS [f] on [f].[familyid] = [p2f].[familyid]  where 1=1';  IF (@Family­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [f].[family­ID] = @Family\_­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [f].[Family­ID],[f].[lastname]';  IF (@Family­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family\_­ID int'  , @Family\_­ID = @Family­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Clients­By­Createdate] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150120  -- Description: User defined stored procedure to  -- select People by created date range  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Clients­By­Createdate]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  --SELECT [P].[Person­ID], 'Family­Name' = [F].[Last­Name]  -- , [P].[Last­Name], [P].[First­Name], [P].[Created­Date]  -- FROM [Person] AS [P]  -- JOIN Personto­Family AS P2F ON [P].[Person­ID] = [P2F].[Person­ID]  -- JOIN [family] AS [F] ON [P2F].[Family­ID] = [F].[Family­ID]  -- where 1 = 2 AND [P].[Created­Date] >= @Start­Date AND [P].[Created­Date] <= @End­Date order by [P].[Last­Name],[P].[Person­ID] OPTION(RECOMPILE)  select @spexecutesql­Str ='SELECT [P].[Person­ID],[P].[Last­Name],[P].[Middle­Name],[P].[First­Name],[P].[Birth­Date]  ,[P].[Gender],[P].[Age],[P].[Modified­Date],[P].[Created­Date]  from [Person] AS [P]  where 1 = 1'    -- Return all People if nothing was passed in  IF ((@Start­Date is NULL) AND (@End­Date is NULL))  SET @Recompile = 0  IF (@Start­Date is NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] >= @Begin­Date'  IF (@End­Date is NOT NULL)  BEGIN  SET @End­Date = Date­Add(dd,1,@End­Date)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] < @End­Date'  END  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [P].[Created­Date] DESC, [P].[Last­Name],  [P].[Person­ID] ASC'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'BEGINDate' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Begin­Date datetime, @End­Date datetime'  , @Begin­Date = @Start­Date  , @End­Date = @End­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Clients­By­Modifieddate] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150120  -- Description: User defined stored procedure to  -- select People by created date range  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Clients­By­Modifieddate]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  --SELECT [P].[Person­ID], 'Family­Name' = [F].[Last­Name]  -- , [P].[Last­Name], [P].[First­Name], [P].[Created­Date]  -- FROM [Person] AS [P]  -- JOIN Personto­Family AS P2F ON [P].[Person­ID] = [P2F].[Person­ID]  -- JOIN [family] AS [F] ON [P2F].[Family­ID] = [F].[Family­ID]  -- where 1 = 2 AND [P].[Created­Date] >= @Start­Date AND [P].[Created­Date] <= @End­Date order by [P].[Last­Name],[P].[Person­ID] OPTION(RECOMPILE)  select @spexecutesql­Str ='SELECT [P].[Person­ID],[P].[Last­Name],[P].[Middle­Name],[P].[First­Name],[P].[Birth­Date],[P].[Gender],[P].[Modified­Date]  from [Person] AS [P]  where 1 = 1'    -- Return all People if nothing was passed in  IF ((@Start­Date is NULL) AND (@End­Date is NULL))  SET @Recompile = 0  IF (@Start­Date is NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Modified­Date] >= @Begin­Date'  IF (@End­Date is NOT NULL)  BEGIN  SET @End­Date = Date­Add(dd,1,@End­Date)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Modified­Date] < @End­Date'  END  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [P].[Modified­Date] ASC, [P].[Last­Name],  [P].[Person­ID] ASC'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'BEGINDate' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Begin­Date datetime, @End­Date datetime'  , @Begin­Date = @Start­Date  , @End­Date = @End­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Families] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150110  -- Description: User defined stored procedure to  -- select all families  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Families]  -- Add the parameters for the stored procedure her  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  select @spexecutesql­Str ='SELECT [F].[Family­ID], ''Family­Name'' = [F].[Last­Name]  from [family] AS [F]  where 1 = 1'    -- Return all families and associated properties if nothing was passed in  SET @Recompile = 0  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [F].[Last­Name],[F].[Family­ID]'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecutesql­Str;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Family­Members] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family­ID | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150103  -- Description: stored procedure to list family members  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Family­Members]  -- Add the parameters for the stored procedure here  @Family­ID int = NULL  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'SELECT [f].[familyid], Family­Name = [f].[lastname],[P].[Last­Name],[P].[Firstname] from [person] as [p]  join [personto­Family] [p2f] on [p].[personid] = [p2f].[personid]  join [family] AS [f] on [f].[familyid] = [p2f].[familyid]  where 1=1';  IF (@Family­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [f].[family­ID] = @Family\_­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [f].[lastname],[f].[familyid]';  IF (@Family­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family\_­ID int'  , @Family\_­ID = @Family­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Nursing­Womenby­Create­Date­Range] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Begin\_­Date | date | 3 |
| @End\_­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150120  -- Description: User defined stored procedure to  -- select Nursing­Women by created date range  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Nursing­Womenby­Create­Date­Range]  -- Add the parameters for the stored procedure here  @Begin\_­Date date = NULL,  @End\_­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  select @spexecutesql­Str ='SELECT [P].[Person­ID],[P].[Last­Name],[P].[First­Name],[P].[Created­Date]  from [Person] AS [P]  where Nursing­Mother = 1'    -- Return all Nursing­Women if nothing was passed in  IF ((@Begin\_­Date is NULL) AND (@End\_­Date is NULL))  SET @Recompile = 0  IF (@Begin\_­Date is NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] >= @Begin­Date'  IF (@End\_­Date is NOT NULL)  BEGIN  SET @End\_­Date = Date­Add(dd,1,@End\_­Date)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] < @End­Date'  END  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [P].[Created­Date] DESC, [P].[Last­Name],  [P].[Person­ID] ASC'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'BEGINDate' = @Begin\_­Date, 'ENDDate' = @End\_­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Begin­Date datetime, @End­Date datetime'  , @Begin­Date = @Begin\_­Date  , @End­Date = @End\_­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Peopleby­Create­Date­Range] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Begin\_­Date | date | 3 |
| @End\_­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150120  -- Description: User defined stored procedure to  -- select People by created date range  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Peopleby­Create­Date­Range]  -- Add the parameters for the stored procedure here  @Begin\_­Date date = NULL,  @End\_­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  --SELECT [P].[Person­ID], 'Family­Name' = [F].[Last­Name]  -- , [P].[Last­Name], [P].[First­Name], [P].[Created­Date]  -- FROM [Person] AS [P]  -- JOIN Personto­Family AS P2F ON [P].[Person­ID] = [P2F].[Person­ID]  -- JOIN [family] AS [F] ON [P2F].[Family­ID] = [F].[Family­ID]  -- where 1 = 2 AND [P].[Created­Date] >= @Begin\_­Date AND [P].[Created­Date] <= @End\_­Date order by [P].[Last­Name],[P].[Person­ID] OPTION(RECOMPILE)  select @spexecutesql­Str ='SELECT [P].[Person­ID],[P].[Last­Name],[P].[First­Name],[P].[Created­Date]  from [Person] AS [P]  where 1 = 1'    -- Return all People if nothing was passed in  IF ((@Begin\_­Date is NULL) AND (@End\_­Date is NULL))  SET @Recompile = 0  IF (@Begin\_­Date is NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] >= @Begin­Date'  IF (@End\_­Date is NOT NULL)  BEGIN  SET @End\_­Date = Date­Add(dd,1,@End\_­Date)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] < @End­Date'  END  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [P].[Created­Date] DESC, [P].[Last­Name],  [P].[Person­ID] ASC'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'BEGINDate' = @Begin\_­Date, 'ENDDate' = @End\_­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Begin­Date datetime, @End­Date datetime'  , @Begin­Date = @Begin\_­Date  , @End­Date = @End\_­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLList­Potential­Duplicate­People] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Debug | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150127  -- Description: stored procedure to potential  -- duplicate people  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLList­Potential­Duplicate­People]  -- Add the parameters for the stored procedure here  @Debug bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    SELECT @spexecute­SQLStr =  N'SELECT P1Person­ID = P1.Person­ID  , P2Person­ID = P2.Person­ID  , P1Last­Name = P1.Last­Name  , P2Last­Name = P2.Last­Name  , P1First­Name = P1.First­Name  , P2First­Name = P2.First­Name  , P1Birth­Date = P1.Birth­Date  , P2Birth­Date = P2.Birth­Date  , P1Gender = P1.Gender  , P2Gender = P2.Gender  , P1Created­Date = P1.Created­Date  , P2Created­Date = P2.Created­Date  , P1Modified­Date = P1.Modified­Date  , P2Modified­Date = P2.Modified­Date  from person AS P1  JOIN person AS P2 on  P1.Last­Name = P2.Last­Name  AND P1.First­Name = P2.First­Name  AND P1.Age = P2.Age  AND P1.Person­ID != P2.Person­ID  OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr;  EXEC [sp\_executesql] @spexecute­SQLStr;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Log Errors  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLList­Potential­Duplicate­Properties] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Debug | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150127  -- Description: stored procedure to potential  -- duplicate properties  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLList­Potential­Duplicate­Properties]  -- Add the parameters for the stored procedure here  @Debug bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    SELECT @spexecute­SQLStr =  N'SELECT [P1Property­ID] = [P1].[Property­ID]  , [P2Property­ID] = [P2].[Property­ID]  , [P1Street­Number] = [P1].[Street­Number]  , [P2Street­Number] = [P2].[Street­Number]  , [P1Street] = [P1].[Street]  , [P2Street] = [P2].[Street]  , [P1Street­Suffix] = [P1].[Street­Suffix]  , [P2Street­Suffix] = [P2].[Street­Suffix]  , [P1City] = [P1].[City]  , [P2City] = [P2].[City]  , [P1State] = [P1].[State]  , [P2State] = [P2].[State]  , [P1Zip­Code] = [P1].[Zipcode]  , [P2Zip­Code] = [P2].[Zipcode]  , [P1County] = [P1].[County]  , [P2County] = [P2].[County]  , [P1Created­Date] = [P1].[Created­Date]  , [P2Created­Date] = [P2].[Created­Date]  , [P1Modified­Date] = [P1].[Modified­Date]  , [P2Modified­Date] = [P2].[Modified­Date]  from [Property] AS [P1]  JOIN [Property] AS [P2] on  [P1].[Street] = [P2].[Street]  AND [P1].[Street­Number] = [P2].[Street­Number]  AND [P1].[City] = [P2].[City]  AND [P1].[County] = [P2].[County]  AND [P1].[Zipcode] = [P2].[Zipcode]  AND [P1].[State] = [P2].[State]  AND [P1].[Property­ID] != [P2].[Property­ID]  OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr;  EXEC [sp\_executesql] @spexecute­SQLStr;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Log Errors  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Begin\_­Date | date | 3 |
| @End\_­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150120  -- Description: User defined stored procedure to  -- select Pregnant­Women by created date range  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range]  -- Add the parameters for the stored procedure here  @Begin\_­Date date = NULL,  @End\_­Date date = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int, @Return­Error int;  select @spexecutesql­Str ='SELECT [P].[Person­ID],[P].[Last­Name],[P].[First­Name],[P].[Created­Date]  from [Person] AS [P]  where Pregnant = 1'    -- Return all Pregnant­Women if nothing was passed in  IF ((@Begin\_­Date is NULL) AND (@End\_­Date is NULL))  SET @Recompile = 0  IF (@Begin\_­Date is NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] >= @Begin­Date'  IF (@End\_­Date is NOT NULL)  BEGIN  SET @End\_­Date = Date­Add(dd,1,@End\_­Date)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [P].[Created­Date] < @End­Date'  END  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [P].[Created­Date] DESC, [P].[Last­Name],  [P].[Person­ID] ASC'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'BEGINDate' = @Begin\_­Date, 'ENDDate' = @End\_­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Begin­Date datetime, @End­Date datetime'  , @Begin­Date = @Begin\_­Date  , @End­Date = @End\_­Date;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  SELECT @Return­Error = ERROR\_­NUMBER();  -- DROP TABLE ##Returned­Values;  RETURN @Return­Error  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­SLMost­Recent­Blood­Test­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person\_­ID | int | 4 |
| @Min\_­Lead\_­Value | numeric(4,1) | 5 |
| @Max\_­Lead\_­Value | numeric(4,1) | 5 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20141222  -- Description: select most recent blood test results  -- optionally only return for a specific  -- client  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­SLMost­Recent­Blood­Test­Results]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @Min\_­Lead\_­Value numeric(4,1) = NULL,  @Max\_­Lead\_­Value numeric(4,1) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Order­By NVARCHAR(500),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  SELECT @spexecutesql­Str = N'Select [P].[Last­Name],[P].[First­Name],[P].[Person­ID],[BTR].[Lead­Value], [BTR].[Sample­Date],[BTR].[Hemoglobin­Value]  ,[BTR].[Created­Date],[BTR].[Modified­Date],[BTR].[Blood­Test­Results­ID] from [Person] AS [P]  JOIN [Blood­Test­Results] AS [BTR] on [BTR].[Blood­Test­Results­ID] = (  select top 1 [Blood­Test­Results­ID] from [Blood­Test­Results]  where [Blood­Test­Results].[Person­ID] = [P].[Person­ID]  -- AND [Lead­Value] > @Min­Lead­Value uncomment to list most recent tests with BLL above minimum  )  WHERE 1=1';  IF @Min\_­Lead\_­Value IS NULL  SET @Min\_­Lead\_­Value = 0.0;  IF @Person\_­ID IS NOT NULL  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [p].[Person­ID] = @Person­ID';  IF (@Min\_­Lead\_­Value > 0)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] >= @Min­Lead­Value';  IF (@Max\_­Lead\_­Value > 0)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' AND [BTR].[Lead­Value] < @Max­Lead­Value';  IF @Person\_­ID is NULL  SELECT @spexecutesql­Str = @spexecutesql­Str + N' ORDER BY [p].[Last­Name], [P].[Person­ID] ASC, [BTR].[Sample­Date] DESC';    IF ( (@Person\_­ID IS NULL) AND (@Min\_­Lead\_­Value = 0) )  SET @Recompile = 0;  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  -- If debugging print out query  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'PID' = @Person\_­ID, 'MLV' = @Min\_­Lead\_­Value, 'Max­LV' = @Max\_­Lead\_­Value, 'R' = @Recompile;  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Person­ID int,@Min­Lead­Value numeric(4,1), @Max­Leadvalue numeric(4,1)'  , @Person­ID = @Person\_­ID, @Min­Lead­Value = @Min\_­Lead\_­Value, @Maxlead­Value = @Max\_­Lead\_­Value;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Person­Notes] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to list  -- person and their ethnicities  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Person­Notes]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'select P.Person­ID,Last­Name,First­Name, PN.Notes,P.Modified­Date from Person AS P  LEFT OUTER JOIN Person­Notes AS PN on P.Person­ID = PN.PErson­ID  where Notes is not null';  IF (@Person­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [P].[Person­ID] = @Person­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [P].[lastname],[P].[Personid]';  IF (@Person­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Personto­Ethnicity] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to list  -- person and their ethnicities  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Personto­Ethnicity]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'select P.Person­ID,Last­Name,First­Name,E.Ethnicity from Person AS P  LEFT OUTER JOIN Personto­Ethnicity AS P2E on P.Person­ID = P2E.PErson­ID  LEFT OUTER JOIN Ethnicity AS E on P2E.Ethnicity­ID = E.Ethnicity­ID  WHERE 1 = 1';  IF (@Person­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [P].[Person­ID] = @Person­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [P].[lastname],[P].[Personid]';  IF (@Person­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Personto­Language] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: stored procedure to list  -- person and their languages  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Personto­Language]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecute­SQLStr NVARCHAR(4000)  , @Recompile BIT = 1, @Error­Log­ID int;    --IF (@Family­ID IS NULL)  --BEGIN  -- RAISERROR ('You must supply at least one parameter.', 11, -1);  -- RETURN;  --END;  SELECT @spexecute­SQLStr =  N'select [P].Person­ID,Last­Name,First­Name, L.Language­Name from Person AS P  LEFT OUTER JOIN Personto­Language AS P2L on P.Person­ID = P2L.PErson­ID  LEFT OUTER JOIN Language AS L on P2L.Language­ID = L.Language­ID  WHERE 1 = 1';  IF (@Person­ID IS NOT NULL)  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' AND [P].[Person­ID] = @Person­ID';  SELECT @spexecute­SQLStr = @spexecute­SQLStr  + N' order by [P].[lastname],[P].[Personid]';  IF (@Person­ID IS NULL)  SET @Recompile = 0;    IF @Recompile = 1  SELECT @spexecute­SQLStr = @spexecute­SQLStr + N' OPTION(RECOMPILE)';  BEGIN TRY  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Person­ID' = @Person­ID;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Person­ID int'  , @Person­ID = @Person­ID;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Relation­Ship­Types] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150110  -- Description: User defined stored procedure to  -- select all relationship types and IDs  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Relation­Ship­Types]  -- Add the parameters for the stored procedure her  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR (4000),  @Recompile BIT = 1, @Error­Log­ID int;  -- Insert statements for procedure here  select @spexecutesql­Str ='SELECT [RT].[Relationship­Type­ID], [RT].[Relationship­Type­Name]  from [Relationship­Type] AS [RT]  where 1 = 1'    -- Return all families and associated properties if nothing was passed in  SET @Recompile = 0  -- order by last name  SELECT @spexecutesql­Str = @spexecutesql­Str + N' order by [RT].[Relationship­Type­Name]'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  BEGIN TRY  EXEC [sp\_executesql] @spexecutesql­Str;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Add error information to errorlog  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Status] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Target­Type | varchar(50) | 50 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150215  -- Description: returns valid status codes for passed in type - Child  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Status]  -- Add the parameters for the stored procedure here  @Target­Type varchar(50) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  -- IF (@Status­Type = 'Child')  select status­Name from Target­Status where Target­Type = @Target­Type  END  GO |

Uses

[[dbo].[Target­Status]](#(local)/User_databases/LCCHPDev/Tables/TargetStatus)

|  |
| --- |
| [dbo].[usp\_­Sl­Summary­Report] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Min­Lead­Value | numeric(4,1) | 5 |
| @Max­Lead­Value | numeric(4,1) | 5 |
| @Min­Age | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- blood tests conducted within  -- the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Summary­Report]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @Min­Lead­Value numeric(4,1) = NULL,  @Max­Lead­Value Numeric(4,1) = NULL,  @Min­Age int = 18,  @DEBUG bit = 0    AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int, @Parm­Definition nvarchar(500)  , @Client­Count int, @New­Client­Count int, @BLLCount int, @EBLLCount int, @Pregnant­Women­Count int  , @Nursing­Mother­Count int, @Nursing­Infant­Count int, @Adult­Count int, @Blood­Test­Count int, @Home­Soil­Count int;    BEGIN TRY  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101';    IF (@End­Date IS NULL)  SET @End­Date = GETDATE();  IF (@Start­Date >= @End­Date)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String ='End­Date must be after Start­Date: Start­Date: ' + cast(@Start­Date as varchar) + ' End­Date: ' + cast(@End­Date as varchar)  RAISERROR (@Error­String, 11, -1);  RETURN;  END  -- clients  SELECT @spexecutesql­Str = 'Select @Clients = count(Person­ID) from (  SELECT Person­ID from blood­Test­Results WHERE 1=1 AND Sample­Date >= @Start­Date AND Sample­Date < @End­Date  UNION  SELECT Person­ID from Questionnaire WHERE 1=1 AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  ) total'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Clients int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Clients = @Client­Count OUTPUT;  -- New­Clients  SELECT @spexecutesql­Str = 'Select @New­Clients = count(Person­ID) from Person WHERE is­Client = 1'    IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date >= @Start­Date'    IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Created­Date < @End­Date'  IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @New­Clients int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @New­Clients = @New­Client­Count OUTPUT  -- Total Blood­Lead Tests  SELECT @spexecutesql­Str = 'SELECT @Blood­Test­Count = count([Blood­Test­Results­ID]) from [Blood­Test­Results]  where 1 = 1'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date >= @Start­Date'  IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Blood­Test­Count int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Blood­Test­Count = @Blood­Test­Count OUTPUT  -- BLL 5 ug/dl - 9.9 ug/dl  SET @Min­Lead­Value = 5.0;  SET @Max­Lead­Value = 10.0;  SELECT @spexecutesql­Str = 'SELECT @BLLCount = count([Blood­Test­Results­ID]) from [Blood­Test­Results]  where 1 = 1'  IF (@Min­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Leadvalue >= @Min­Lead­Value'    IF (@Max­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Lead­Value < @Max­Lead­Value'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date >= @Start­Date'  IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date, [Min­Lead­Value] = @Min­Lead­Value, [Max­Lead­Value] = @Max­Lead­Value  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1), @BLLCount int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Min­Lead­Value = @Min­Lead­Value  , @Max­Lead­Value = @Max­Lead­Value  , @BLLCount = @BLLCount OUTPUT  -- BLL 10 ug/dl and above  SET @Min­Lead­Value = 10;  SET @Max­Lead­Value = NULL;  SELECT @spexecutesql­Str = 'SELECT @EBLLCount = count([Blood­Test­Results­ID]) from [Blood­Test­Results]  where 1 = 1'  IF (@Min­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Leadvalue >= @Min­Lead­Value'    IF (@Max­Lead­Value IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Lead­Value < @Max­Lead­Value'  IF (@Start­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date >= @Start­Date'  IF (@End­Date IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND Sample­Date < @End­Date'    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date, [Min­Lead­Value] = @Min­Lead­Value, [Max­Lead­Value] = @Max­Lead­Value  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Min­Lead­Value numeric(4,1), @Max­Lead­Value numeric(4,1), @EBLLCount int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Min­Lead­Value = @Min­Lead­Value  , @Max­Lead­Value = @Max­Lead­Value  , @EBLLCount = @EBLLCount OUTPUT  -- Pregnant women  select @spexecutesql­Str ='Select @Pregnant­Women = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Pregnant from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select top 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Pregnant desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Pregnant = 1  UNION  SELECT Person­ID,Pregnant from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Pregnant = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Pregnant = 1'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Start­Date' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date datetime, @End­Date datetime, @Pregnant­Women int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Pregnant­Women = @Pregnant­Women­Count OUTPUT;  -- Nursing Mothers  SELECT @spexecutesql­Str = 'Select @Nursing­Mothers = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Nursing­Mother from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select top 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Nursing­Mother desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Nursing­Mother = 1  UNION  SELECT Person­ID,Nursing­Mother from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Nursing­Mother = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Nursing­Mother = 1'  IF ((Date­Diff(YYYY,@Start­Date,@End­Date) > 5))  SET @Recompile = 0;    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Nursing­Mothers int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Nursing­Mothers = @Nursing­Mother­Count OUTPUT  -- Nursing Infants  SELECT @spexecutesql­Str = 'Select @Nursing­Infants = COUNT(Person­ID) from (  Select BTR.Person­ID,Q.Nursing­Infant from Blood­Test­Results AS BTR  LEFT OUTER JOIN [Questionnaire] AS [Q] on [Q].[Questionnaire­ID] = (  select TOP 1 [Questionnaire­ID] from [Questionnaire]  where [Questionnaire].[Person­ID] = [BTR].[Person­ID]  AND Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  order by Nursing­Infant desc  )  where Sample­Date >= @Start­Date and Sample­Date < @End­Date AND Q.Nursing­Infant = 1  UNION  SELECT Person­ID,Nursing­Infant from Questionnaire where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND Nursing­Infant = 1  ) Clientsin­Reporting­Period  where Clientsin­Reporting­Period.Nursing­Infant = 1'  IF ((Date­Diff(YYYY,@Start­Date,@End­Date) > 5))  SET @Recompile = 0;    IF (@Recompile = 1)  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, [Start­Date] = @Start­Date, [End­Date] = @End­Date  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date date, @End­Date date, @Nursing­Infants int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Nursing­Infants = @Nursing­Infant­Count OUTPUT  -- Adults  IF (@Start­Date IS NULL)  SET @Start­Date = '18000101'  IF (@ENDDate IS NULL)  SET @End­Date = Get­Date();  -- Create temporary table  CREATE Table #Temp­Potential­Adults  ( Person­ID int  , Test­ID int  , Age­At­Visit tinyint  , Most­Recent­Visit date  , Birthdate date  , Visits tinyint  )  -- insert values from bloodtest results  insert Into #Temp­Potential­Adults (Person­ID, Most­Recent­Visit, Test­ID)  select Person­ID,Most­Recent­Visit = Sample­Date, Test­ID = Blood­Test­Results­ID  from Bloodtest­Results  where Sample­Date >= @Start­Date AND Sample­Date < @End­Date  -- insert values from questionnaire  insert Into #Temp­Potential­Adults (Person­ID, Most­Recent­Visit, Test­ID)  Select Person­ID,Most­Recent­Visit = Questionnaire­Date, Test­ID = Questionnaire­ID  from Questionnaire  where Questionnaire­Date >= @Start­Date AND Questionnaire­Date < @End­Date  and (ISNULL(Questionnaire.Nursing­Mother,0) = 0 OR ISNULL(Questionnaire.Pregnant,0) = 0 )  -- populate birthdate only if the difference from most recent visit to birthdate is at least min­Age  update #Temp­Potential­Adults set Birth­Date = Person.Birthdate,  Age­At­Visit = [dbo].[udf\_­Calculate­Age]([Person].[Birth­Date],Most­Recent­Visit)  FROM #Temp­Potential­Adults  JOIN Person on Person.Person­ID = #Temp­Potential­Adults.Person­ID  where Datediff(yy,Person.Birth­Date,Most­Recent­Visit) > @Min­Age  Select @Adult­Count = count(distinct Person­ID) from #Temp­Potential­Adults  where Age­At­Visit > @Min­Age  drop table #Temp­Potential­Adults  -- Home visits and soil testing  select @spexecutesql­Str ='select @Home­Soil­Count = count(Person­ID) from (  SELECT Person­ID  from Blood­Test­Results where Sample­Date >= @Start­Date and Sample­Date < @End­Date  AND Client­Status­ID in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  UNION  -- people with questionnaire but no blood test during reporting period  Select Q.Person­ID  from Questionnaire AS Q  LEFT OUTER JOIN [Blood­Test­Results] AS [BTR] on [BTR].[Blood­Test­Results­ID] = (  select top 1 [Blood­Test­Results­ID] from [Blood­Test­Results]  where [Blood­Test­Results].[Person­ID] = [Q].[Person­ID]  -- AND Sample­Date >= @Start­Date AND Sample­Date < @End­Date  AND BTR.Client­Status­ID  in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  order by Sample­Date desc  )  where Questionnaire­Date >= @Start­Date and Questionnaire­Date < @End­Date  AND BTR.Client­Status­ID  in ( SELECT [TS].[Status­ID] from [Target­Status] AS [TS]  where Target­Type = ''Person''  AND Status­Name in (''Home visit'', ''Home Visit and Soil Sample'', ''Soil Sample'')  )  ) Home­Visit­Soil­Samples'    IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + N' OPTION(RECOMPILE)';  IF (@DEBUG = 1)  SELECT @spexecutesql­Str, 'Start­Date' = @Start­Date, 'ENDDate' = @End­Date, 'DEBUG' = @Debug  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Start­Date datetime, @End­Date datetime, @Home­Soil­Count int OUTPUT'  , @Start­Date = @Start­Date  , @End­Date = @End­Date  , @Home­Soil­Count = @Home­Soil­Count OUTPUT;  -- total tests  select 'Client­Count' = @Client­Count, 'New­Client­Count' = @New­Client­Count, 'Blood­Test­Count' = @Blood­Test­Count  , 'BLL5to10ug­Perdl' = @BLLCount, 'EBLLCount' = @EBLLCount, 'Pregnant­Women' = @Pregnant­Women­Count  , 'Nursing­Mother­Count' = @Nursing­Mother­Count, 'Nursing­Infant­Count' = @Nursing­Infant­Count  , 'Adult­Count' = @Adult­Count, 'Home­Soil­Count' = @Home­Soil­Count;  -- SELECT @s­SQL = N'SELECT @retval­OUT = MAX(Person­ID) FROM ' + @tablename;  --SET @Parm­Definition = N'@retval­OUT int OUTPUT';  --EXEC sp\_executesql @s­SQL, @Parm­Definition, @retval­OUT=@retval OUTPUT;  --SELECT @retval;    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

[[dbo].[udf\_­Calculate­Age]](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_CalculateAge)

|  |
| --- |
| [dbo].[usp\_­Sl­Summary­Report\_­Meta­Data] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @Min­Lead­Value | numeric(4,1) | 5 |
| @Max­Lead­Value | numeric(4,1) | 5 |
| @Min­Age | int | 4 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure returns the number of  -- blood tests conducted within  -- the specified date range.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Summary­Report\_­Meta­Data]  -- Add the parameters for the stored procedure here  @Start­Date date = NULL,  @End­Date date = NULL,  @Min­Lead­Value numeric(4,1) = NULL,  @Max­Lead­Value Numeric(4,1) = NULL,  @Min­Age int = 18,  @DEBUG bit = 0    AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int, @Parm­Definition nvarchar(500)  , @Client­Count int, @New­Client­Count int, @BLLCount int, @EBLLCount int, @Pregnant­Women­Count int  , @Nursing­Mother­Count int, @Nursing­Infant­Count int, @Adult­Count int, @Blood­Test­Count int, @Home­Soil­Count int;    BEGIN TRY    select 'Client­Count' = @Client­Count, 'New­Client­Count' = @New­Client­Count, 'Blood­Test­Count' = @Blood­Test­Count  , 'BLL5to10ug­Perdl' = @BLLCount, 'EBLLCount' = @EBLLCount, 'Pregnant­Women' = @Pregnant­Women­Count  , 'Nursing­Mother­Count' = @Nursing­Mother­Count, 'Nursing­Infant­Count' = @Nursing­Infant­Count  , 'Adult­Count' = @Adult­Count, 'Home­Soil­Count' = @Home­Soil­Count  where 1 = 0;  -- SELECT @s­SQL = N'SELECT @retval­OUT = MAX(Person­ID) FROM ' + @tablename;  --SET @Parm­Definition = N'@retval­OUT int OUTPUT';  --EXEC sp\_executesql @s­SQL, @Parm­Definition, @retval­OUT=@retval OUTPUT;  --SELECT @retval;    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_­Sl­Target­Sample­Type] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Sample\_­Target | varchar(50) | 50 |
| @p2 | int | 4 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150102  -- Description: retrieve sample types for people (lead levels)  -- =============================================  CREATE PROCEDURE [dbo].[usp\_­Sl­Target­Sample­Type]  -- Add the parameters for the stored procedure here  @Sample\_­Target varchar(50) = NULL,  @p2 int = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str nvarchar(4000), @RECOMPILE bit =1;  -- Insert statements for procedure here  SELECT @spexecutesql­Str = 'SELECT [Sample­Type­ID],[Sample­Type­Name] from [Sample­Type] where 1=1'    if (@Sample\_­Target IS NOT NULL)  SELECT @spexecutesql­Str = @spexecutesql­Str + ' AND [Sample­Type].[Sample­Target] = @Sample­Target'  IF @Recompile = 1  SELECT @spexecutesql­Str = @spexecutesql­Str + ' OPTION(RECOMPILE)';  EXEC [sp\_executesql] @spexecutesql­Str  , N'@Sample­Target varchar(50)', @Sample­Target = @Sample\_­Target  END  GO |

|  |
| --- |
| [dbo].[usp\_up­Blood­Test­Results] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Blood­Test­Results­ID | int | 4 |
| @New\_­Sample\_­Date | date | 3 |
| @New\_­Lab\_­Date | date | 3 |
| @New\_­Blood\_­Lead\_­Result | numeric(4,1) | 5 |
| @New\_­Hemoglobin\_­Value | numeric(4,1) | 5 |
| @New\_­Lab\_­ID | int | 4 |
| @New\_­Blood\_­Test\_­Costs | money | 8 |
| @New\_­Sample\_­Type\_­ID | tinyint | 1 |
| @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed | bit | 1 |
| @New\_­Exclude\_­Result | bit | 1 |
| @New\_­Client\_­Status\_­ID | smallint | 2 |
| @New\_­Notes | varchar(3000) | 3000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130618  -- Description: Stored Procedure to update  -- blood test results records  -- =============================================  -- DROP PROCEDURE usp\_up­Blood­Test­Results  CREATE PROCEDURE [dbo].[usp\_up­Blood­Test­Results]  -- Add the parameters for the stored procedure here  @Blood­Test­Results­ID int = NULL,  @New\_­Sample\_­Date date = NULL,  @New\_­Lab\_­Date date = NULL,  @New\_­Blood\_­Lead\_­Result numeric(4,1) = NULL,  -- @New\_­Flag smallint = NULL,  @New\_­Hemoglobin\_­Value numeric(4,1) = NULL,  @New\_­Lab\_­ID int = NULL,  @New\_­Blood\_­Test\_­Costs money = NULL,  @New\_­Sample\_­Type\_­ID tinyint = NULL,  @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed bit = NULL,  @New\_­Exclude\_­Result bit = NULL,  @New\_­Client\_­Status\_­ID smallint = NULL,  @New\_­Notes varchar(3000) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int, @spupdate­Blood­Test­Resultssql­Str NVARCHAR(4000);  -- insert statements for procedure here  BEGIN TRY  -- Check if Blood­Test­Results­ID is valid, if not return  IF NOT EXISTS (SELECT Blood­Test­Results­ID from Blood­Test­Results where Blood­Test­Results­ID = @Blood­Test­Results­ID)  BEGIN  RAISERROR(15000, -1,-1,'usp\_up­Blood­Test­Results');  END    -- BUILD update statement  if (@New\_­Blood\_­Lead\_­Result is null)  select @New\_­Blood\_­Lead\_­Result = Lead­Value from Blood­Test­Results where Blood­Test­Results­ID = @Blood­Test­Results­ID    SELECT @spupdate­Blood­Test­Resultssql­Str = N'update Blood­Test­Results set Lead­Value = @Blood\_­Lead\_­Result'  IF (@New\_­Sample\_­Date IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Sample­Date = @Sample\_­Date'  IF (@New\_­Lab\_­Date IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Lab­Submission­Date = @Lab\_­Date'  IF (@New\_­Hemoglobin\_­Value IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Hemoglobin­Value = @Hemoglobin\_­Value'  IF (@New\_­Lab\_­ID IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Lab­ID = @Lab\_­ID'  IF (@New\_­Blood\_­Test\_­Costs IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Blood­Test­Costs = @Blood\_­Test\_­Costs'  IF (@New\_­Sample\_­Type\_­ID IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Sample­Type­ID = @Sample\_­Type\_­ID'  IF (@New\_­Taken\_­After\_­Property\_­Remediation\_­Completed IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Taken­After­Property­Remediation­Completed = @Taken\_­After\_­Property\_­Remediation\_­Completed'  IF (@New\_­Exclude\_­Result IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Exclude­Result = @Exclude\_­Result'  IF (@New\_­Client\_­Status\_­ID IS NOT NULL)  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N', Client­Status­ID = @Client\_­Status\_­ID'  -- make sure to only update record for specified Blood­Test­Results  SELECT @spupdate­Blood­Test­Resultssql­Str = @spupdate­Blood­Test­Resultssql­Str + N' WHERE Blood­Test­Results­ID = @Blood­Test­Results­ID'  IF (@DEBUG = 1)  SELECT @spupdate­Blood­Test­Resultssql­Str, Lead­Value = @New\_­Blood\_­Lead\_­Result, Sample­Date = @New\_­Sample\_­Date, Lab­Submission­Date = @New\_­Lab\_­Date  , Hemoglobin­Vlaue = @New\_­Hemoglobin\_­Value, Lab­ID = @New\_­Lab\_­ID, Blood­Test­Costs = @New\_­Blood\_­Test\_­Costs, Sample­Type­ID = @New\_­Sample\_­Type\_­ID  , Taken­After­Property­Remediation­Completed = @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed, Exclude­Result = @New\_­Exclude\_­Result  , Client­Status­ID = @New\_­Client\_­Status\_­ID, Blood­Test­Results­ID = @Blood­Test­Results­ID  EXEC [sp\_executesql] @spupdate­Blood­Test­Resultssql­Str  , N'@Blood\_­Lead\_­Result numeric(4,1), @Sample\_­Date date, @Lab\_­Date date, @Hemoglobin\_­Value numeric(4,1), @Lab\_­ID int, @Blood\_­Test\_­Costs money  , @Sample\_­Type\_­ID tinyint, @Taken\_­After\_­Property\_­Remediation\_­Completed bit, @Exclude\_­Result bit  , @Client\_­Status\_­ID smallint, @Blood­Test­Results­ID int'  , @Blood\_­Lead\_­Result = @New\_­Blood\_­Lead\_­Result  , @Sample\_­Date = @New\_­Sample\_­Date  , @Lab\_­Date = @New\_­Lab\_­Date  , @Hemoglobin\_­Value = @New\_­Hemoglobin\_­Value  , @Lab\_­ID = @New\_­Lab\_­ID  , @Blood\_­Test\_­Costs = @New\_­Blood\_­Test\_­Costs  , @Sample\_­Type\_­ID = @New\_­Sample\_­Type\_­ID  , @Taken\_­After\_­Property\_­Remediation\_­Completed = @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed  , @Exclude\_­Result = @New\_­Exclude\_­Result  , @Client\_­Status\_­ID = @New\_­Client\_­Status\_­ID  , @Blood­Test­Results­ID = @Blood­Test­Results­ID  IF (@New\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Blood­Test­Results­Notes]  @Blood­Test­Results\_­ID = @Blood­Test­Results­ID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[usp\_­Insert­Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_up­Blood­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Blood­Test­Results­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Blood­Test­Results­ID | int | 4 |
| @New\_­Sample\_­Date | date | 3 |
| @New\_­Lab\_­Date | date | 3 |
| @New\_­Blood\_­Lead\_­Result | numeric(4,1) | 5 |
| @New\_­Sample\_­Type\_­ID | tinyint | 1 |
| @New\_­Lab\_­ID | int | 4 |
| @New\_­Flag | smallint | 2 |
| @New\_­Client\_­Status\_­ID | smallint | 2 |
| @New\_­Hemoglobin\_­Value | numeric(4,1) | 5 |
| @New\_­Blood\_­Test\_­Costs | money | 8 |
| @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed | bit | 1 |
| @New\_­Exclude\_­Result | bit | 1 |
| @New\_­Notes | varchar(3000) | 3000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150618  -- Description: stored procedure to update blood test results  -- data  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Blood­Test­Results­Web­Screen]  -- Add the parameters for the stored procedure here  @Blood­Test­Results­ID int = NULL,  @New\_­Sample\_­Date date = NULL,  @New\_­Lab\_­Date date = NULL,  @New\_­Blood\_­Lead\_­Result numeric(4,1) = NULL,  @New\_­Sample\_­Type\_­ID tinyint = NULL,  @New\_­Lab\_­ID int = NULL,  @New\_­Flag smallint = NULL,  @New\_­Client\_­Status\_­ID smallint = NULL,  @New\_­Hemoglobin\_­Value numeric(4,1) = NULL,  @New\_­Blood\_­Test\_­Costs money = NULL,  @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed bit = NULL,  @New\_­Exclude\_­Result bit = NULL,  @New\_­Notes varchar(3000) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  BEGIN  DECLARE @Error­Log­ID int, @Retest­Date\_return\_value int,  @update­Blood­Test­Results­Return­Value int;    -- If no family ID was passed in exit  IF (@Blood­Test­Results­ID IS NULL)  BEGIN  RAISERROR ('Blood test results ID must be supplied', 11, -1);  RETURN;  END;  BEGIN TRY  -- update person flag/retest date  IF (@New\_­Flag IS NOT NULL)  BEGIN  declare @Retest\_­Date date, @Sample\_­Date date, @Person\_­ID int;  -- determine person­ID  select @Person\_­ID = Person­ID, @Sample\_­Date = Sample­Date from Blood­Test­Results where Blood­Test­Results­ID = @Blood­Test­Results­ID    -- set the retest date based on integer value passed in as Flag  SET @Retest\_­Date = DATEADD(dd,@New\_­Flag,@Sample\_­Date);  -- update Person table with the new retest date  -- anyone with a blood test is a client  EXEC @Retest­Date\_return\_value = [dbo].[usp\_up­Person]  @Person\_­ID = @Person\_­ID  , @New\_­Retest­Date = @Retest\_­Date  , @New\_­Client­Status­ID = @New\_­Client\_­Status\_­ID;  END  -- update bloodtest­Results  EXEC @update­Blood­Test­Results­Return­Value = [dbo].[usp\_up­Blood­Test­Results]  @Blood­Test­Results­ID = @Blood­Test­Results­ID,  @New\_­Sample\_­Date = @New\_­Sample\_­Date,  @New\_­Lab\_­Date = @New\_­Lab\_­Date,  @New\_­Blood\_­Lead\_­Result = @New\_­Blood\_­Lead\_­Result,  @New\_­Hemoglobin\_­Value = @New\_­Hemoglobin\_­Value,  @New\_­Lab\_­ID = @New\_­Lab\_­ID,  @New\_­Blood\_­Test\_­Costs = @New\_­Blood\_­Test\_­Costs,  @New\_­Sample\_­Type\_­ID = @New\_­Sample\_­Type\_­ID,  @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed = @New\_­Taken\_­After\_­Property\_­Remediation\_­Completed,  @New\_­Exclude\_­Result = @New\_­Exclude\_­Result,  @New\_­Client\_­Status\_­ID = @New\_­Client\_­Status\_­ID,  @New\_­Notes = @New\_­Notes,  @DEBUG = @DEBUG  END TRY  BEGIN CATCH -- insert person  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH; -- insert new person  END  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[usp\_up­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_up­Client­Flag] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150605  -- Description: procedure to update the is­Client  -- flag to 1 if the person has completed  -- a bloodtest or a questionnaire.  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Client­Flag]  -- Add the parameters for the stored procedure here  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @spexecutesql­Str NVARCHAR(4000), @Recompile BIT = 1, @Error­Log­ID int;    BEGIN TRY  -- Set is­Client true if person has a bloodtest or questionnaire  update Person Set is­Client = 1 where is­Client = 0 AND Person­ID IN  ( Select Person­ID from Blood­Test­Results  UNION  Select Person­ID from Questionnaire  )  -- Set is­Client false if person does not have a bloodtest or a questionnaire  update Person Set is­Client = 0 where is­Client = 1 AND Person­ID NOT IN  ( Select Person­ID from Blood­Test­Results  UNION  Select Person­ID from Questionnaire  )    END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_up­Client­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­ID | int | 4 |
| @Person\_­ID | int | 4 |
| @New\_­First­Name | varchar(50) | 50 |
| @New\_­Middle­Name | varchar(50) | 50 |
| @New\_­Last­Name | varchar(50) | 50 |
| @New\_­Birth­Date | date | 3 |
| @New\_­Gender | char | 1 |
| @New\_­Status­ID | smallint | 2 |
| @New\_­Foreign­Travel | bit | 1 |
| @New\_­Outof­Site | bit | 1 |
| @New\_­Eats­Foreign­Food | bit | 1 |
| @New\_­Email­Address | varchar(320) | 320 |
| @New\_­Retest­Date | date | 3 |
| @New\_­Moved | bit | 1 |
| @New\_­Moved­Date | date | 3 |
| @New\_is­Closed | bit | 1 |
| @New\_is­Resolved | bit | 1 |
| @New\_­Client­Notes | varchar(3000) | 3000 |
| @New\_­Travel­Notes | varchar(3000) | 3000 |
| @New\_­Hobby­Notes | varchar(3000) | 3000 |
| @New\_­Release­Notes | varchar(3000) | 3000 |
| @New\_­Guardian­ID | int | 4 |
| @New\_­Person­Code | smallint | 2 |
| @New\_is­Smoker | bit | 1 |
| @New\_is­Client | bit | 1 |
| @New\_­Nursing­Mother | bit | 1 |
| @New\_­Nursing­Infant | bit | 1 |
| @New\_­Pregnant | bit | 1 |
| @New\_­Ethnicity­ID | tinyint | 1 |
| @New\_­Language­ID | tinyint | 1 |
| @New\_­Primary­Language | bit | 1 |
| @New\_­Hobby­ID | int | 4 |
| @New\_­Occupation­ID | int | 4 |
| @New\_­Occupation\_­Start­Date | date | 3 |
| @New\_­Occupation\_­End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150325  -- Description: stored procedure to update data  -- from the Add a new client web page  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Client­Web­Screen]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @Person\_­ID int = NULL,  @New\_­First­Name varchar(50) = NULL,  @New\_­Middle­Name varchar(50) = NULL,  @New\_­Last­Name varchar(50) = NULL,  @New\_­Birth­Date date = NULL,  @New\_­Gender char(1) = NULL,  @New\_­Status­ID smallint = NULL,  @New\_­Foreign­Travel bit = NULL,  @New\_­Outof­Site bit = NULL,  @New\_­Eats­Foreign­Food bit = NULL,  @New\_­Email­Address varchar(320) = NULL,  @New\_­Retest­Date date = NULL,  @New\_­Moved bit = NULL,  @New\_­Moved­Date date = NULL,  @New\_is­Closed bit = 0,  @New\_is­Resolved bit = 0,  @New\_­Client­Notes varchar(3000) = NULL,  @New\_­Travel­Notes varchar(3000) = NULL,  @New\_­Hobby­Notes varchar(3000) = NULL,  @New\_­Release­Notes varchar(3000) = NULL,  @New\_­Guardian­ID int = NULL,  @New\_­Person­Code smallint = NULL,  @New\_is­Smoker bit = NULL,  @New\_is­Client bit = NULL,  @New\_­Nursing­Mother bit = NULL,  @New\_­Nursing­Infant bit = NULL,  @New\_­Pregnant bit = NULL,  --@New\_is­Nursing bit = NULL,  --@New\_is­Pregnant bit = NULL,  @New\_­Ethnicity­ID tinyint = NULL,  @New\_­Language­ID tinyint = NULL,  @New\_­Primary­Language bit = 1,  @New\_­Hobby­ID int = NULL,  @New\_­Occupation­ID int = NULL,  @New\_­Occupation\_­Start­Date date = NULL,  @New\_­Occupation\_­End­Date date = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  BEGIN  DECLARE @Error­Log­ID int,  @update­Person\_return\_value int,  @Ethnicity\_return\_value int,  @Personto­Family\_return\_value int,  @Personto­Language\_return\_value int,  @Personto­Hobby\_return\_value int,  @Personto­Occupation\_return\_value int,  @Personto­Ethnicity\_return\_value int;    -- If no family ID was passed in exit  IF (@Family\_­ID IS NULL OR @Person\_­ID IS NULL)  BEGIN  RAISERROR ('Family and Person must be supplied', 11, -1);  RETURN;  END;  if (@New\_­Last­Name is null)  BEGIN  select @New\_­Last­Name = Lastname from Family where Family­ID = @Family\_­ID  END  BEGIN TRY -- update person  EXEC @update­Person\_return\_value = [dbo].[usp\_up­Person]  @Person\_­ID = @Person\_­ID,  @New\_­First­Name = @New\_­First­Name,  @New\_­Middle­Name = @New\_­Middle­Name,  @New\_­Last­Name = @New\_­Last­Name,  @New\_­Birth­Date = @New\_­Birth­Date,  @New\_­Gender = @New\_­Gender,  @New\_­Status­ID = @New\_­Status­ID,  @New\_­Foreign­Travel = @New\_­Foreign­Travel,  @New\_­Outof­Site = @New\_­Outof­Site,  @New\_­Eats­Foreign­Food = @New\_­Eats­Foreign­Food,  @New\_­Email­Address = @New\_­Email­Address,  @New\_­Retest­Date = @New\_­Retest­Date,  @New\_­Moved = @New\_­Moved,  @New\_­Moved­Date = @New\_­Moved­Date,  @New\_is­Closed = @New\_is­Closed,  @New\_is­Resolved = @New\_is­Resolved,  @New\_­Person­Notes = @New\_­Client­Notes,  @New\_­Hobby­Notes = @New\_­Hobby­Notes,  @New\_­Travel­Notes = @New\_­Travel­Notes,  @New\_­Release­Notes = @New\_­Release­Notes,  @New\_­Guardian­ID = @New\_­Guardian­ID,  @New\_­Person­Code = @New\_­Person­Code,  @New\_is­Smoker = @New\_is­Smoker,  @New\_is­Client = @New\_is­Client,  @New\_­Nursing­Mother = @New\_­Nursing­Mother,  @New\_­Nursing­Infant = @New\_­Nursing­Infant,  @New\_­Pregnant = @New\_­Pregnant,  @DEBUG = @DEBUG  -- Associate person to Ethnicity  IF ((@New\_­Ethnicity­ID IS NOT NULL) AND  (NOT EXISTS (SELECT Person­ID from Personto­Ethnicity where Ethnicity­ID = @New\_­Ethnicity­ID and Person­ID = @Person\_­ID)))  EXEC @Ethnicity\_return\_value = [dbo].[usp\_­Insert­Personto­Ethnicity]  @Person­ID = @Person\_­ID,  @Ethnicity­ID = @New\_­Ethnicity­ID  -- CODE FOR FUTURE EXTENSIBILITY OF UPDATING ETHNICITY  --IF (@New\_­Ethnicity IS NOT NULL)  --EXEC @Ethnicity\_return\_value = [dbo].[usp\_up­Ethnicity]  -- @Person­ID = @Person\_­ID,  -- @New\_­Ethnicity­ID = @New\_­Ethnicity­ID,  -- @DEBUG = @DEBUG,  -- @Personto­Ethnicity­ID = @New\_­Personto­Ethnicity­ID OUTPUT  -- Associate person to family  -- If the person isn't already associated with that family  if NOT EXISTS(SELECT Person­ID from Personto­Family where Family­ID = @Family\_­ID and Person­ID = @Person\_­ID)  EXEC @Personto­Family\_return\_value = usp\_­Insert­Personto­Family  @Person­ID = @Person\_­ID, @Family­ID = @Family\_­ID, @OUTPUT = @Personto­Family\_return\_value OUTPUT;  -- Associate person to language  IF (@New\_­Language­ID is not NULL)  EXEC @Personto­Language\_return\_value = usp\_­Insert­Personto­Language  @Language­ID = @New\_­Language­ID, @Person­ID = @Person\_­ID, @is­Primary­Language = @New\_­Primary­Language;  -- associate person to Hobby  IF ((@New\_­Hobby­ID is not NULL) AND  (NOT EXISTS (SELECT Person­ID from Personto­Hobby where Hobby­ID = @New\_­Hobby­ID and Person­ID = @Person\_­ID)) )  EXEC @Personto­Hobby\_return\_value = usp\_­Insert­Personto­Hobby  @Hobby­ID = @New\_­Hobby­ID, @Person­ID = @Person\_­ID;  -- associate person to occupation  if ((@New\_­Occupation­ID is not NULL))  IF (NOT EXISTS (SELECT Person­ID from Personto­Occupation where Occupation­ID = @New\_­Occupation­ID and Person­ID = @Person\_­ID))  EXEC @Personto­Occupation\_return\_value = [dbo].[usp\_­Insert­Personto­Occupation]  @Person­ID = @Person\_­ID,  @Occupation­ID = @New\_­Occupation­ID,  @Start­Date = @New\_­Occupation\_­Start­Date,  @End­Date = @New\_­Occupation\_­End­Date  ELSE  EXEC @Personto­Occupation\_return\_value = [dbo].[usp\_up­Occupation]  @Person­ID = @Person\_­ID,  @Occupation­ID = @New\_­Occupation­ID,  @Occupation\_­Start­Date = @New\_­Occupation\_­Start­Date,  @Occupation\_­End­Date = @New\_­Occupation\_­End­Date,  @DEBUG = @DEBUG;  END TRY  BEGIN CATCH -- insert person  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH; -- insert new person  END  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Tables/PersontoEthnicity)

[[dbo].[Personto­Family]](#(local)/User_databases/LCCHPDev/Tables/PersontoFamily)

[[dbo].[Personto­Hobby]](#(local)/User_databases/LCCHPDev/Tables/PersontoHobby)

[[dbo].[Personto­Occupation]](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation)

[[dbo].[usp\_­Insert­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity)

[[dbo].[usp\_­Insert­Personto­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily)

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_­Insert­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage)

[[dbo].[usp\_­Insert­Personto­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation)

[[dbo].[usp\_up­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upOccupation)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_up­Family] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­ID | int | 4 |
| @New\_­Last\_­Name | varchar(50) | 50 |
| @New\_­Number\_of\_­Smokers | tinyint | 1 |
| @New\_­Primary\_­Language\_­ID | tinyint | 1 |
| @New\_­Notes | varchar(3000) | 3000 |
| @New\_­Pets | tinyint | 1 |
| @New\_­Frequently\_­Wash\_­Pets | bit | 1 |
| @New\_­Pets\_in\_and\_out | bit | 1 |
| @New\_­Primary\_­Property\_­ID | int | 4 |
| @New\_­Foreign­Travel | bit | 1 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150214  -- Description: Stored Procedure to update Family information  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Family]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @New\_­Last\_­Name varchar(50) = NULL,  @New\_­Number\_of\_­Smokers tinyint = 0,  @New\_­Primary\_­Language\_­ID tinyint = 1,  @New\_­Notes varchar(3000) = NULL,  @New\_­Pets tinyint = NULL,  @New\_­Frequently\_­Wash\_­Pets bit = NULL,  @New\_­Pets\_in\_and\_out bit = NULL,  @New\_­Primary\_­Property\_­ID int = NULL,  @New\_­Foreign­Travel bit = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @Error­Log­ID int, @spupdate­Familysql­Str nvarchar(4000)  , @Notes­ID INT, @Recompile BIT = 1;    BEGIN TRY -- update Family information  -- BUILD update statement  IF (@New\_­Last\_­Name IS NULL)  SELECT @New\_­Last\_­Name = Last­Name from family where Family­ID = @Family\_­ID;    SELECT @spupdate­Familysql­Str = N'update Family set Lastname = @Last­Name'  IF (@New\_­Number\_of\_­Smokers IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Numberof­Smokers = @Numberof­Smokers'  IF (@New\_­Primary\_­Language\_­ID IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Primary­Language­ID = @Primary­Language­ID'  IF (@New\_­Pets IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Pets = @Pets'  IF (@New\_­Frequently\_­Wash\_­Pets IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Frequently­Wash­Pets = @Frequently­Wash­Pets'    IF (@New\_­Pets\_in\_and\_out IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Petsinandout = @Petsinandout'  IF (@New\_­Primary\_­Property\_­ID IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Primary­Property­ID = @Primary­Property­ID'  IF (@New\_­Foreign­Travel IS NOT NULL)  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N', Foreign­Travel = @Foreign­Travel'  SELECT @spupdate­Familysql­Str = @spupdate­Familysql­Str + N' WHERE Family­ID = @Family­ID'  IF @DEBUG = 1  SELECT @spupdate­Familysql­Str, 'Lastname' = @New\_­Last\_­Name, 'Numberof­Smokers' = @New\_­Number\_of\_­Smokers  , 'Primary­Language­ID' = @New\_­Primary\_­Language\_­ID, 'Pets' = @New\_­Pets, 'Petsinandout' = @New\_­Pets\_in\_and\_out  , 'Primary­Property­ID' = @New\_­Primary\_­Property\_­ID, 'Frequently­Wash­Pets' = @New\_­Frequently\_­Wash\_­Pets  , 'Foreign­Travel' = @New\_­Foreign­Travel    IF (@New\_­Notes IS NOT NULL)  BEGIN  IF @DEBUG = 1  SELECT 'EXEC [dbo].[usp\_­Insert­Family­Notes] @Family\_­ID = @Family\_­ID, @Notes = @New\_­Notes, @Inserted­Notes­ID = @Notes­ID OUTPUT '  , @Family\_­ID, @New\_­Notes  EXEC [dbo].[usp\_­Insert­Family­Notes]  @Family\_­ID = @Family\_­ID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END    -- update Family table  EXEC [sp\_executesql] @spupdate­Familysql­Str  , N'@Last­Name VARCHAR(50), @Numberof­Smokers tinyint, @Primary­Language­ID tinyint  , @Pets tinyint, @Petsinandout BIT, @Primary­Property­ID int, @Frequently­Wash­Pets bit, @Foreign­Travel bit, @Family­ID int'  , @Last­Name = @New\_­Last\_­Name  , @Numberof­Smokers = @New\_­Number\_of\_­Smokers  , @Primary­Language­ID = @New\_­Primary\_­Language\_­ID  , @Pets = @New\_­Pets  , @Petsinandout = @New\_­Pets\_in\_and\_out  , @Primary­Property­ID = @New\_­Primary\_­Property\_­ID  , @Frequently­Wash­Pets = @New\_­Frequently\_­Wash\_­Pets  , @Foreign­Travel = @New\_­Foreign­Travel  , @Family­ID = @Family\_­ID  END TRY -- update Family  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[usp\_­Insert­Family­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Familyto­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Familyto­Property­ID | int | 4 |
| @Property­Link­Type­ID | int | 4 |
| @Start­Date | date | 3 |
| @End­Date | date | 3 |
| @is­Primary­Residence | bit | 1 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150417  -- Description: Stored Procedure to update new Familyto­Property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Familyto­Property]  -- Add the parameters for the stored procedure here  @Familyto­Property­ID int = NULL,  @Property­Link­Type­ID int = NULL,  @Start­Date date = NULL,  @End­Date date = NULL,  @is­Primary­Residence bit = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @spexecute­SQLStr nvarchar(4000);  -- Insert statements for procedure here  BEGIN TRY  SELECT @spexecute­SQLStr = N'update Familyto­Property set End­Date = @End\_­Date';    IF @Property­Link­Type­ID IS NOT NULL  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ', Property­Link­Type­ID = @Property\_­Link\_­Type\_­ID'  IF @Start­Date IS NOT NULL  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ', Start­Date = @Start\_­Date'  IF @is­Primary­Residence IS NOT NULL  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ', is­Primary­Residence = @is\_­Primary\_­Residence'  -- Add filters to update the correct record  SELECT @spexecute­SQLStr = @spexecute­SQLStr + ' Where Familyto­Property­ID = @Family\_to\_­Property\_­ID'  IF (@DEBUG = 1)  SELECT @spexecute­SQLStr, 'Familyto­Property­ID' = @Familyto­Property­ID, 'Property­Link­Type­ID' = @Property­Link­Type­ID  , 'Start­Date' = @Start­Date, 'End­Date' = @End­Date, 'is­Primary­Residence' = @is­Primary­Residence;    EXEC [sp\_executesql] @spexecute­SQLStr  , N'@Family\_to\_­Property\_­ID int, @Property\_­Link\_­Type\_­ID int, @Start\_­Date date, @End\_­Date date, @is\_­Primary\_­Residence bit'  , @Family\_to\_­Property\_­ID = @Familyto­Property­ID  , @Property\_­Link\_­Type\_­ID = @Property­Link­Type­ID  , @Start\_­Date = @Start­Date  , @End\_­Date = @End­Date  , @is\_­Primary\_­Residence = @is­Primary­Residence;  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

|  |
| --- |
| [dbo].[usp\_up­Family­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­ID | int | 4 |
| @is­New­Address | bit | 1 |
| @New\_­Last\_­Name | varchar(50) | 50 |
| @Property­ID | int | 4 |
| @New\_­Construction­Type | int | 4 |
| @New\_­Area­ID | int | 4 |
| @New\_isin­Historic­District | bit | 1 |
| @New\_is­Remodeled | bit | 1 |
| @New\_­Remodel­Date | date | 3 |
| @New\_isin­City­Limits | bit | 1 |
| @New\_­Address\_­Line1 | varchar(100) | 100 |
| @New\_­Address\_­Line2 | varchar(100) | 100 |
| @New\_­City­Name | varchar(50) | 50 |
| @New\_­County | varchar(50) | 50 |
| @New\_­State­Abbr | char(2) | 2 |
| @New\_­Zip­Code | varchar(10) | 10 |
| @New\_­Year\_­Built | date | 3 |
| @New\_­Property­Link­Type­ID | tinyint | 1 |
| @New\_­Movein\_­Date | date | 3 |
| @New\_­Move­Out\_­Date | date | 3 |
| @New\_is­Primary­Residence | bit | 1 |
| @New\_­Owner\_id | int | 4 |
| @New\_is\_­Owner\_­Occupied | bit | 1 |
| @New\_­Replaced­Pipes­Faucets | bit | 1 |
| @New\_­Total­Remediation­Costs | money | 8 |
| @New\_­Property­Notes | varchar(3000) | 3000 |
| @New\_is\_­Residential | bit | 1 |
| @New\_is­Currently­Being­Remodeled | bit | 1 |
| @New\_has\_­Peeling\_­Chipping\_­Patin | bit | 1 |
| @New\_is\_­Rental | bit | 1 |
| @New\_­Primary­Phone | bigint | 8 |
| @Primary­Phone­Priority | tinyint | 1 |
| @New\_­Secondary­Phone | bigint | 8 |
| @Secondary­Phone­Priority | tinyint | 1 |
| @New\_­Number\_of\_­Smokers | tinyint | 1 |
| @New\_­Primary\_­Language\_­ID | tinyint | 1 |
| @New\_­Family\_­Notes | varchar(3000) | 3000 |
| @New\_­Pets | tinyint | 1 |
| @New\_­Frequently\_­Wash\_­Pets | bit | 1 |
| @New\_­Pets\_in\_and\_out | bit | 1 |
| @New\_­Foreign­Travel | bit | 1 |
| @New\_­Owner­Contact­Information | varchar(1000) | 1000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150329  -- Description: Stored Procedure to update Family  -- web screen information  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Family­Web­Screen]  -- Add the parameters for the stored procedure here  @Family\_­ID int = NULL,  @is­New­Address bit = 0,  @New\_­Last\_­Name varchar(50) = NULL,  @Property­ID int = NULL,  @New\_­Construction­Type int = NULL,  @New\_­Area­ID int = NULL,  @New\_isin­Historic­District bit = NULL,  @New\_is­Remodeled bit = NULL,  @New\_­Remodel­Date date = NULL,  @New\_isin­City­Limits bit = NULL,  @New\_­Address\_­Line1 varchar(100) = NULL,  @New\_­Address\_­Line2 varchar(100) = NULL,  @New\_­City­Name varchar(50) = NULL,  @New\_­County varchar(50) = NULL,  @New\_­State­Abbr char(2) = NULL,  @New\_­Zip­Code varchar(10) = NULL,  @New\_­Year\_­Built date = NULL,  @New\_­Property­Link­Type­ID tinyint = NULL,  @New\_­Movein\_­Date date = NULL,  @New\_­Move­Out\_­Date date = NULL,  @New\_is­Primary­Residence bit = NULL,  @New\_­Owner\_id int = NULL,  @New\_is\_­Owner\_­Occupied bit = NULL,  @New\_­Replaced­Pipes­Faucets bit = NULL,  @New\_­Total­Remediation­Costs money = NULL,  @New\_­Property­Notes varchar(3000) = NULL,  @New\_is\_­Residential bit = NULL,  @New\_is­Currently­Being­Remodeled bit = NULL,  @New\_has\_­Peeling\_­Chipping\_­Patin bit = NULL,  @New\_is\_­Rental bit = NULL,  @New\_­Primary­Phone bigint = NULL,  @Primary­Phone­Priority tinyint = 1,  @New\_­Secondary­Phone bigint = NULL,  @Secondary­Phone­Priority tinyint = 2,  @New\_­Number\_of\_­Smokers tinyint = NULL,  @New\_­Primary\_­Language\_­ID tinyint = 1,  @New\_­Family\_­Notes varchar(3000) = NULL,  @New\_­Pets tinyint = NULL,  @New\_­Frequently\_­Wash\_­Pets bit = NULL,  @New\_­Pets\_in\_and\_out bit = NULL,  -- @New\_­Primary\_­Property\_­ID int = NULL,  @New\_­Foreign­Travel bit = NULL,  @New\_­Owner­Contact­Information varchar(1000) = NULL,  @DEBUG BIT = 1  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  DECLARE @Error­Log­ID int  , @Update\_­Family\_return\_value int  , @Update\_­Property\_return\_value int  , @Notes­ID INT, @Recompile BIT = 1  , @New\_­Primary\_­Property\_­ID int  , @Primaryphone\_return\_value int  , @Secondaryphone\_return\_value int  , @Familyto­Property\_return\_value int  , @Inserted­Familyto­Property­ID int  , @Debug­Log­ID int    BEGIN TRY -- update Family information  -- Exit if family isn't specified  IF (@Family\_­ID IS NULL or @Family\_­ID = '')  BEGIN  RAISERROR ('Family must be specified', 11, 1);  RETURN;  END;  -- Existing property need the primary property id  IF (@is­New­Address = 0)  BEGIN  -- Select the primary property id  Select @Property­ID = Primary­Property­ID from Family where Family­ID = @Family\_­ID;  EXEC @Update\_­Property\_return\_value = [dbo].[usp\_up­Property]  @Property­ID = @Property­ID,  @New\_­Construction­Type­ID = @New\_­Construction­Type,  @New\_­Area­ID = @New\_­Area­ID,  @New\_isin­Historic­District = @New\_isin­Historic­District,  @New\_is­Remodeled = @New\_is­Remodeled,  @New\_­Remodel­Date = @New\_­Remodel­Date,  @New\_isin­City­Limits = @New\_isin­City­Limits,  @New\_­Address­Line1 = @New\_­Address\_­Line1,  @New\_­Address­Line2 = @New\_­Address\_­Line2,  @New\_­City = @New\_­City­Name,  @New\_­State = @New\_­State­Abbr,  @New\_­Zipcode = @New\_­Zip­Code,  @New\_­Year­Built = @New\_­Year\_­Built,  @New\_­Ownerid = @New\_­Owner\_id,  @New\_is­Owner­Occuppied = @New\_is\_­Owner\_­Occupied,  @New\_­Replaced­Pipes­Faucets = @New\_­Replaced­Pipes­Faucets,  @New\_­Total­Remediation­Costs = @New\_­Total­Remediation­Costs,  @New\_­Property­Notes = @New\_­Property­Notes,  @New\_is­Residential = @New\_is\_­Residential,  @New\_is­Currently­Being­Remodeled = @New\_is­Currently­Being­Remodeled,  @New\_has­Peeling­Chipping­Paint = @New\_has\_­Peeling\_­Chipping\_­Patin,  @New\_­County = @New\_­County,  @New\_is­Rental = @New\_is\_­Rental,  @New\_­Owner­Contact­Information = @New\_­Owner­Contact­Information,  @DEBUG = @DEBUG  -- SET the new primary property ID  SET @New\_­Primary\_­Property\_­ID = @Property­ID;  END  IF (@is­New­Address = 1)  BEGIN  EXEC [dbo].[usp\_­Insert­Property]  @Construction­Type­ID = @New\_­Construction­Type,  @Area­ID = @New\_­Area­ID,  @isin­Historic­District = @New\_isin­Historic­District,  @is­Remodeled = @New\_is­Remodeled,  @Remodel­Date = @New\_­Remodel­Date,  @isin­City­Limits = @New\_isin­City­Limits,  @Address­Line1 = @New\_­Address\_­Line1,  @Address­Line2 = @New\_­Address\_­Line2,  @City = @New\_­City­Name,  @State = @New\_­State­Abbr,  @Zipcode = @New\_­Zip­Code,  @Year­Built = @New\_­Year\_­Built,  @Ownerid = @New\_­Owner\_id,  @is­Owner­Occuppied = @New\_is\_­Owner\_­Occupied,  @Replaced­Pipes­Faucets = @New\_­Replaced­Pipes­Faucets,  @Total­Remediation­Costs = @New\_­Total­Remediation­Costs,  @New\_­Property­Notes = @New\_­Property­Notes,  -- @is­Residential = @New\_is­Residential,  @is­Currently­Being­Remodeled = @New\_is­Currently­Being­Remodeled,  @has­Peeling­Chipping­Paint = @New\_has\_­Peeling\_­Chipping\_­Patin,  @County = @New\_­County,  -- @is­Rental = @New\_is­Rental,  --@Over­Ride­Duplicate = @New\_­Over­Ride­Duplicate,  @Owner­Contact­Information = @New\_­Owner­Contact­Information,  @Property­ID = @New\_­Primary\_­Property\_­ID OUTPUT  END  EXEC @Update\_­Family\_return\_value = [dbo].[usp\_up­Family]  @Family\_­ID = @Family\_­ID,  @New\_­Last\_­Name = @New\_­Last\_­Name,  @New\_­Number\_of\_­Smokers = @New\_­Number\_of\_­Smokers,  @New\_­Primary\_­Language\_­ID = @New\_­Primary\_­Language\_­ID,  @New\_­Notes = @New\_­Family\_­Notes,  @New\_­Pets = @New\_­Pets,  @New\_­Frequently\_­Wash\_­Pets = @New\_­Frequently\_­Wash\_­Pets,  @New\_­Pets\_in\_and\_out = @New\_­Pets\_in\_and\_out,  @New\_­Primary\_­Property\_­ID = @New\_­Primary\_­Property\_­ID,  @New\_­Foreign­Travel = @New\_­Foreign­Travel,  @DEBUG = @DEBUG;  EXEC @Familyto­Property\_return\_value = [usp\_­Insert­Familyto­Property]  @Family­ID = @Family\_­ID,  @Property­ID = @New\_­Primary\_­Property\_­ID,  @Property­Link­Type­ID = @New\_­Property­Link­Type­ID,  @Start­Date = @New\_­Movein\_­Date,  @End­Date = @New\_­Move­Out\_­Date,  @is­Primary­Residence = @New\_is­Primary­Residence,  @New­Familyto­Property­ID = @Inserted­Familyto­Property­ID OUTPUT  if (@New\_­Primary­Phone is not NULL)  BEGIN -- insert Primary Phone  DECLARE @Primary­Phone­Number­ID\_­OUTPUT bigint, @Phone­Type­ID tinyint;  SELECT @Phone­Type­ID = Phone­Number­Type­ID from Phone­Number­Type where Phone­Number­Type­Name = 'Primary Phone';  EXEC @Primaryphone\_return\_value = [dbo].[usp\_­Insert­Phone­Number]  @Phone­Number = @New\_­Primary­Phone,  @Phone­Number­Type­ID = @Phone­Type­ID,  @DEBUG = @DEBUG,  @Phone­Number­ID\_­OUTPUT = @Primary­Phone­Number­ID\_­OUTPUT OUTPUT    EXEC [dbo].[usp\_­Insert­Familyto­Phone­Number]  @Family­ID = @Family\_­ID,  @Number­Priority = @Primary­Phone­Priority,  @Phone­Number­ID = @Primary­Phone­Number­ID\_­OUTPUT,  @DEBUG = @DEBUG  END -- insert Primary Phone  if (@New\_­Secondary­Phone is not NULL)  BEGIN -- insert Secondary Phone  DECLARE @Secondary­Phone­Number­ID\_­OUTPUT bigint;  SELECT @Phone­Type­ID = Phone­Number­Type­ID from Phone­Number­Type where Phone­Number­Type­Name = 'Secondary Phone';  EXEC @Secondaryphone\_return\_value = [dbo].[usp\_­Insert­Phone­Number]  @Phone­Number = @New\_­Secondary­Phone,  @Phone­Number­Type­ID = @Phone­Type­ID,  @DEBUG = @DEBUG,  @Phone­Number­ID\_­OUTPUT = @Secondary­Phone­Number­ID\_­OUTPUT OUTPUT  EXEC [dbo].[usp\_­Insert­Familyto­Phone­Number]  @Family­ID = @Family\_­ID,  @Number­Priority = @Secondary­Phone­Priority,  @Phone­Number­ID = @Secondary­Phone­Number­ID\_­OUTPUT,  @DEBUG = @DEBUG    END -- insert Secondary Phone  END TRY -- update Family  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumberType)

[[dbo].[usp\_­Insert­Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber)

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

[[dbo].[usp\_­Insert­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp\_up­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily)

[[dbo].[usp\_up­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp\_up­Occupation] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person­ID | int | 4 |
| @Occupation­ID | tinyint | 1 |
| @Occupation\_­Start­Date | date | 3 |
| @Occupation\_­End­Date | date | 3 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20150327  -- Description: Stored Procedure to update occupation records  -- =============================================  -- DROP PROCEDURE usp\_up­Occupation  CREATE PROCEDURE [dbo].[usp\_up­Occupation]  -- Add the parameters for the stored procedure here  @Person­ID int = NULL,  @Occupation­ID tinyint = NULL,  @Occupation\_­Start­Date date = NULL,  @Occupation\_­End­Date date = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Error­Message NVARCHAR(4000), @Update bit, @spupdate­Occupationsql­Str NVARCHAR(4000);  -- Assume there is nothing to update  SET @Update = 0;  -- insert statements for procedure here  BEGIN TRY  IF (@Person­ID IS NULL OR @Occupation­ID IS NULL)  BEGIN  RAISERROR('Occupation and Person must be specified',11,50000);  RETURN;  END  IF (NOT EXISTS (SELECT Person­ID from Personto­Occupation where Person­ID = @Person­ID AND Occupation­ID = @Occupation­ID))  BEGIN  SELECT @Error­Message = 'Secified person: ' + cast(@Person­ID as varchar) + ' is not associated with occupation: '  + cast(@Occupation­ID as varchar) +'. Try creating the assocation with usp\_­Insert­Personto­Occupation';  RAISERROR(@Error­Message,8,50000);  RETURN;  END    -- BUILD update statement  SELECT @spupdate­Occupationsql­Str = N'update Personto­Occupation Set Person­ID = @Person­ID'    IF (@Occupation\_­Start­Date IS NOT NULL)  BEGIN  SET @Update = 1;  SELECT @spupdate­Occupationsql­Str = @spupdate­Occupationsql­Str + ', Start­Date = @Start­Date'  END  IF (@Occupation\_­Start­Date IS NOT NULL)  BEGIN  SET @Update = 1;  SELECT @spupdate­Occupationsql­Str = @spupdate­Occupationsql­Str + ', ENDDate = @ENDDate'  END  IF (@Update = 1)  BEGIN  SELECT @spupdate­Occupationsql­Str = @spupdate­Occupationsql­Str + ' WHERE Person­ID = @Person­ID and Occupation­ID = @Occupation­ID'  IF (@DEBUG = 1)  SELECT @spupdate­Occupationsql­Str, 'Start­Date' = @Occupation\_­Start­Date, 'End­Date' = @Occupation\_­End­Date,  'Person­ID' = @Person­ID, 'Occupation­ID' = @Occupation­ID  EXEC [sp\_executesql] @spupdate­Occupationsql­Str  , N'@Occupation­ID tinyint, @Person­ID int, @Start­Date date, @End­Date date'  , @Occupation­ID = @Occupation­ID  , @Person­ID = @Person­ID  , @Start­Date = @Occupation\_­Start­Date  , @End­Date = @Occupation\_­End­Date  END  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Personto­Occupation]](#(local)/User_databases/LCCHPDev/Tables/PersontoOccupation)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Person] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Person\_­ID | int | 4 |
| @New\_­First­Name | varchar(50) | 50 |
| @New\_­Middle­Name | varchar(50) | 50 |
| @New\_­Last­Name | varchar(50) | 50 |
| @New\_­Birth­Date | date | 3 |
| @New\_­Gender | char | 1 |
| @New\_­Status­ID | smallint | 2 |
| @New\_­Foreign­Travel | bit | 1 |
| @New\_­Outof­Site | bit | 1 |
| @New\_­Eats­Foreign­Food | bit | 1 |
| @New\_­Email­Address | varchar(320) | 320 |
| @New\_­Retest­Date | date | 3 |
| @New\_­Moved | bit | 1 |
| @New\_­Moved­Date | date | 3 |
| @New\_is­Closed | bit | 1 |
| @New\_is­Resolved | bit | 1 |
| @New\_­Person­Notes | varchar(3000) | 3000 |
| @New\_­Travel­Notes | varchar(3000) | 3000 |
| @New\_­Hobby­Notes | varchar(3000) | 3000 |
| @New\_­Release­Notes | varchar(3000) | 3000 |
| @New\_­Guardian­ID | int | 4 |
| @New\_­Person­Code | smallint | 2 |
| @New\_is­Smoker | bit | 1 |
| @New\_is­Client | bit | 1 |
| @New\_­Nursing­Mother | bit | 1 |
| @New\_­Nursing­Infant | bit | 1 |
| @New\_­Pregnant | bit | 1 |
| @New\_­Client­Status­ID | smallint | 2 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130506  -- Description: Stored Procedure to update new people records  -- =============================================  -- DROP PROCEDURE usp\_up­Person  CREATE PROCEDURE [dbo].[usp\_up­Person]  -- Add the parameters for the stored procedure here  @Person\_­ID int = NULL,  @New\_­First­Name varchar(50) = NULL,  @New\_­Middle­Name varchar(50) = NULL,  @New\_­Last­Name varchar(50) = NULL,  @New\_­Birth­Date date = NULL,  @New\_­Gender char(1) = NULL,  @New\_­Status­ID smallint = NULL,  @New\_­Foreign­Travel bit = NULL,  @New\_­Outof­Site bit = NULL,  @New\_­Eats­Foreign­Food bit = NULL,  @New\_­Email­Address varchar(320) = NULL,  @New\_­Retest­Date date = NULL,  @New\_­Moved bit = NULL,  @New\_­Moved­Date date = NULL,  @New\_is­Closed bit = 0,  @New\_is­Resolved bit = 0,  @New\_­Person­Notes varchar(3000) = NULL,  @New\_­Travel­Notes varchar(3000) = NULL,  @New\_­Hobby­Notes varchar(3000) = NULL,  @New\_­Release­Notes varchar(3000) = NULL,  @New\_­Guardian­ID int = NULL,  @New\_­Person­Code smallint = NULL,  @New\_is­Smoker bit = NULL,  @New\_is­Client bit = NULL,  @New\_­Nursing­Mother bit = NULL,  @New\_­Nursing­Infant bit = NULL,  @New\_­Pregnant bit = NULL,  @New\_­Client­Status­ID smallint = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int, @spupdate­Personsql­Str NVARCHAR(4000);  -- insert statements for procedure here  BEGIN TRY  -- Check if Person­ID is valid, if not return  IF NOT EXISTS (SELECT Person­ID from Person where Person­ID = @Person\_­ID)  BEGIN  RAISERROR(15000, -1,-1,'usp\_up­Person');  END    -- BUILD update statement  IF (@New\_­Last­Name IS NULL)  SELECT @New\_­Last­Name = Last­Name from Person where Person­ID = @Person\_­ID;    SELECT @spupdate­Personsql­Str = N'update Person set Lastname = @Last­Name'  IF (@New\_­First­Name IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', First­Name = @Firstname'  IF (@New\_­Middle­Name IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Middle­Name = @Middle­Name'  IF (@New\_­Birth­Date IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Birth­Date = @Birth­Date'  IF (@New\_­Gender IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Gender = @Gender'  IF (@New\_­Status­ID IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Status­ID = @Status­ID'  IF (@New\_­Foreign­Travel IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Foreign­Travel = @Foreign­Travel'  IF (@New\_­Outof­Site IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Outof­Site = @Outof­Site'  IF (@New\_­Eats­Foreign­Food IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Eats­Foreign­Food = @Eats­Foreign­Food'  IF (@New\_­Email­Address IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Email­Address = @Email­Address'  IF (@New\_­Retest­Date IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Retest­Date = @Retest­Date'  IF (@New\_­Moved IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Moved = @Moved'  IF (@New\_­Moved­Date IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Moved­Date = @Moved­Date'  IF (@New\_is­Closed IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', is­Closed = @is­Closed'  IF (@New\_is­Resolved IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', is­Resolved = @is­Resolved'    IF (@New\_­Guardian­ID IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Guardian­ID = @Guardian­ID'    IF (@New\_­Person­Code IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Person­Code = @Person­Code'  IF (@New\_is­Smoker IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', is­Smoker = @is­Smoker'  IF (@New\_is­Client IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', is­Client = @is­Client'  IF (@New\_­Nursing­Mother IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Nursing­Mother = @Nursing­Mother'  IF (@New\_­Nursing­Infant IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Nursing­Infant = @Nursing­Infant'  IF (@New\_­Pregnant IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Pregnant = @Pregnant'  IF (@New\_­Client­Status­ID IS NOT NULL)  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N', Client­Status­ID = @Client­Status­ID'  -- make sure to only update record for specified person  SELECT @spupdate­Personsql­Str = @spupdate­Personsql­Str + N' WHERE Person­ID = @Person­ID'  IF (@DEBUG = 1)  SELECT @spupdate­Personsql­Str, Last­Name = @New\_­Last­Name, First­Name = @New\_­First­Name, Middle­Name = @New\_­Middle­Name  , Birth­Date = @New\_­Birth­Date, Gender = @New\_­Gender, Status­ID = @New\_­Status­ID, Foreign­Travel = @New\_­Foreign­Travel  , Outof­Site = @New\_­Outof­Site, Eats­Foreign­Food = @New\_­Eats­Foreign­Food, Email­Address = @New\_­Email­Address, Retest­Date = @New\_­Retest­Date  , Moved = @New\_­Moved, Moved­Date = @New\_­Moved­Date, is­Closed = @New\_is­Closed, is­Resolved = @New\_is­Resolved  , Guardian­ID = @New\_­Guardian­ID, Person­Code = @New\_­Person­Code, is­Smoker = @New\_is­Smoker, is­Client = @New\_is­Client  , Nursing­Mother = @New\_­Nursing­Mother, Nursing­Infant = @New\_­Nursing­Infant, Pregnant = @New\_­Pregnant  , Client­Status­ID = @New\_­Client­Status­ID, Person­ID = @Person\_­ID  EXEC [sp\_executesql] @spupdate­Personsql­Str  , N'@Last­Name VARCHAR(50), @First­Name VARCHAR(50), @Middle­Name VARCHAR(50), @Birth­Date date, @Gender char(1)  , @Status­ID smallint, @Foreign­Travel BIT, @Outof­Site bit, @Eats­Foreign­Food bit, @Email­Address varchar(320), @Retest­Date date  , @Moved bit, @Moved­Date date, @is­Closed bit, @is­Resolved bit, @Guardian­ID int, @Person­Code smallint, @is­Smoker bit  , @is­Client bit, @Nursing­Mother bit, @Nursing­Infant bit, @Pregnant bit, @Client­Status­ID smallint, @Person­ID int'  , @Last­Name = @New\_­Last­Name  , @First­Name = @New\_­First­Name  , @Middle­Name = @New\_­Middle­Name  , @Birth­Date = @New\_­Birth­Date  , @Gender = @New\_­Gender  , @Status­ID = @New\_­Status­ID  , @Foreign­Travel = @New\_­Foreign­Travel  , @Outof­Site = @New\_­Outof­Site  , @Eats­Foreign­Food = @New\_­Eats­Foreign­Food  , @EMail­Address = @New\_­Email­Address  , @Retest­Date = @New\_­Retest­Date  , @Moved = @New\_­Moved  , @Moved­Date = @New\_­Moved­Date  , @is­Closed = @New\_is­Closed  , @is­Resolved = @New\_is­Resolved  , @Guardian­ID = @New\_­Guardian­ID  , @Person­Code = @New\_­Person­Code  , @is­Smoker = @New\_is­Smoker  , @is­Client = @New\_is­Client  , @Nursing­Mother = @New\_­Nursing­Mother  , @Nursing­Infant = @New\_­Nursing­Infant  , @Pregnant = @New\_­Pregnant  , @Client­Status­ID = @New\_­Client­Status­ID  , @Person­ID = @Person\_­ID  IF (@New\_­Person­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Notes]  @Person\_­ID = @Person\_­ID,  @Notes = @New\_­Person­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@New\_­Release­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Release­Notes]  @Person\_­ID = @Person\_­ID,  @Notes = @New\_­Travel­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@New\_­Travel­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Travel­Notes]  @Person\_­ID = @Person\_­ID,  @Notes = @New\_­Travel­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  IF (@New\_­Hobby­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Person­Hobby­Notes]  @Person\_­ID = @Person\_­ID,  @Notes = @New\_­Hobby­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

[[dbo].[usp\_­Insert­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes)

[[dbo].[usp\_­Insert­Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes)

[[dbo].[usp\_­Insert­Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewBloodLeadTestResultsWebScreen)

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_up­Blood­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Property] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Property­ID | int | 4 |
| @New\_­Construction­Type­ID | tinyint | 1 |
| @New\_­Area­ID | int | 4 |
| @New\_isin­Historic­District | bit | 1 |
| @New\_is­Remodeled | bit | 1 |
| @New\_­Remodel­Date | date | 3 |
| @New\_isin­City­Limits | bit | 1 |
| @New\_­Address­Line1 | varchar(100) | 100 |
| @New\_­Address­Line2 | varchar(100) | 100 |
| @New\_­City | varchar(50) | 50 |
| @New\_­State | char(2) | 2 |
| @New\_­Zipcode | varchar(12) | 12 |
| @New\_­Year­Built | date | 3 |
| @New\_­Ownerid | int | 4 |
| @New\_is­Owner­Occuppied | bit | 1 |
| @New\_­Replaced­Pipes­Faucets | tinyint | 1 |
| @New\_­Total­Remediation­Costs | money | 8 |
| @New\_­Property­Notes | varchar(3000) | 3000 |
| @New\_is­Residential | bit | 1 |
| @New\_is­Currently­Being­Remodeled | bit | 1 |
| @New\_has­Peeling­Chipping­Paint | bit | 1 |
| @New\_­County | varchar(50) | 50 |
| @New\_is­Rental | bit | 1 |
| @New\_­Owner­Contact­Information | varchar(1000) | 1000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20140817  -- Description: Stored Procedure to update property records  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Property] -- usp\_up­Property  -- Add the parameters for the stored procedure here  @Property­ID int = NULL,  @New\_­Construction­Type­ID tinyint = NULL,  @New\_­Area­ID int = NULL,  @New\_isin­Historic­District bit = NULL,  @New\_is­Remodeled bit = NULL,  @New\_­Remodel­Date date = NULL,  @New\_isin­City­Limits bit = NULL,  @New\_­Address­Line1 varchar(100) = NULL,  @New\_­Address­Line2 varchar(100) = NULL,  @New\_­City varchar(50) = NULL,  @New\_­State char(2) = NULL,  @New\_­Zipcode varchar(12) = NULL,  @New\_­Year­Built date = NULL,  @New\_­Ownerid int = NULL,  @New\_is­Owner­Occuppied bit = NULL,  @New\_­Replaced­Pipes­Faucets tinyint = 0,  @New\_­Total­Remediation­Costs money = NULL,  @New\_­Property­Notes varchar(3000) = NULL,  @New\_is­Residential bit = NULL,  @New\_is­Currently­Being­Remodeled bit = NULL,  @New\_has­Peeling­Chipping­Paint bit = NULL,  @New\_­County varchar(50) = NULL,  @New\_is­Rental bit = NULL,  @New\_­Owner­Contact­Information varchar(1000) = NULL,  @DEBUG bit = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int, @spupdate­Propertysql­Str nvarchar(4000);  -- Insert statements for procedure here  BEGIN TRY  if (@Property­ID i­S NULL)  BEGIN  DECLARE @Error­String VARCHAR(3000);  SET @Error­String = 'Property must be specified';  RAISERROR (@Error­String, 11, -1);  RETURN;  END  -- BUILD update statement  IF (@New\_isin­Historic­District IS NULL)  SELECT @New\_isin­Historic­District = isin­Historic­District from Property where Property­ID = @Property­ID;    SELECT @spupdate­Propertysql­Str = N'update Property set isin­Historic­District = @isin­Historic­District'  IF (@New\_­Construction­Type­ID IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Construction­Type­ID = @Construction­Type­ID'  IF (@New\_­Area­ID IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Area­ID = @Area­ID'  IF (@New\_is­Remodeled IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', is­Remodeled = @is­Remodeled'  IF (@New\_­Remodel­Date IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Remodel­Date = @Remodel­Date'  IF (@New\_isin­City­Limits IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', isin­City­Limits = @isin­City­Limits'  IF (@New\_­Address­Line1 IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Address­Line1 = @Address­Line1'  IF (@New\_­Address­Line2 IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Address­Line2 = @Address­Line2'    IF (@New\_­City IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', City = @City'  IF (@New\_­State IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', State = @State'  IF (@New\_­Zipcode IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Zip­Code = @Zip­Code'  IF (@New\_­Ownerid IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Owner­ID = @Owner­ID'    IF (@New\_is­Owner­Occuppied IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', is­Owner­Occuppied = @is­Owner­Occuppied'    IF (@New\_­Replaced­Pipes­Faucets IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Replaced­Pipes­Faucets = @Replaced­Pipes­Faucets'    IF (@New\_­Total­Remediation­Costs IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Total­Remediation­Costs = @Total­Remediation­Costs'    IF (@New\_is­Residential IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', is­Residential = @is­Residential'    IF (@New\_is­Currently­Being­Remodeled IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', is­Currently­Being­Remodeled = @is­Currently­Being­Remodeled'    IF (@New\_has­Peeling­Chipping­Paint IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', has­Peeling­Chipping­Paint = @has­Peeling­Chipping­Paint'    IF (@New\_­County IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', County = @County'    IF (@New\_is­Rental IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', is­Rental = @is­Rental'    IF (@New\_­Year­Built IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Year­Built = @Year­Built'  IF (@New\_­Owner­Contact­Information IS NOT NULL)  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N', Owner­Contact­Information = @Owner­Contact­Information'  SELECT @spupdate­Propertysql­Str = @spupdate­Propertysql­Str + N' WHERE Property­ID = @Property­ID'  -- update property table  IF @DEBUG = 1  SELECT @spupdate­Propertysql­Str, New\_­Construction­Type­ID = @New\_­Construction­Type­ID, New\_­Area­ID = @New\_­Area­ID  , New\_isin­Historic­District = @New\_isin­Historic­District, New\_is­Remodeled = @New\_is­Remodeled  , New\_­Remodel­Date = @New\_­Remodel­Date, New\_isin­City­Limits = @New\_isin­City­Limits  , New\_­Address­Line1 = @New\_­Address­Line1, New\_­Address­Line2 = @New\_­Address­Line2, New\_­City = @New\_­City  , New\_­State = @New\_­State, New\_­Zipcode = @New\_­Zipcode, New\_­Owner­ID = @New\_­Ownerid  , New\_is­Owner­Occuppied = @New\_is­Owner­Occuppied, New\_­Replaced­Pipes­Faucets = @New\_­Replaced­Pipes­Faucets  , New\_­Property­Notes = @New\_­Property­Notes, New\_­Total­Remediation­Costs = @New\_­Total­Remediation­Costs  , New\_is­Residential = @New\_is­Residential, New\_is­Currently­Being­Remodeled = @New\_is­Currently­Being­Remodeled  , New\_has­Peeling­Chipping­Paint = @New\_has­Peeling­Chipping­Paint, New\_­County = @New\_­County  , New\_is­Rental = @New\_is­Rental, New\_­Year­Built = @New\_­Year­Built  , New\_­Owner­Contact­Information = @New\_­Owner­Contact­Information, Property­ID = @Property­ID  EXEC [sp\_executesql] @spupdate­Propertysql­Str  , N'@Construction­Type­ID tinyint, @Area­ID int, @isin­Historic­District bit, @is­Remodeled bit, @Remodel­Date date  , @isin­City­Limits BIT, @Address­Line1 varchar(100), @Address­Line2 varchar(100), @City varchar(50), @State char(2)  , @Zipcode varchar(12), @Owner­ID int, @is­Owner­Occuppied bit, @Replaced­Pipes­Faucets tinyint, @Total­Remediation­Costs money  , @is­Residential bit, @is­Currently­Being­Remodeled bit, @has­Peeling­Chipping­Paint bit  , @County varchar(50), @is­Rental bit, @Year­Built date, @Owner­Contact­Information varchar(1000), @Property­ID int'  , @Construction­Type­ID = @New\_­Construction­Type­ID  , @Area­ID = @New\_­Area­ID  , @isin­Historic­District = @New\_isin­Historic­District  , @is­Remodeled = @New\_is­Remodeled  , @Remodel­Date = @New\_­Remodel­Date  , @isin­City­Limits = @New\_isin­City­Limits  , @Address­Line1 = @New\_­Address­Line1  , @Address­Line2 = @New\_­Address­Line2  , @City = @New\_­City  , @State = @New\_­State  , @Zipcode = @New\_­Zipcode  , @Owner­ID = @New\_­Ownerid  , @is­Owner­Occuppied = @New\_is­Owner­Occuppied  , @Replaced­Pipes­Faucets = @New\_­Replaced­Pipes­Faucets  , @Total­Remediation­Costs = @New\_­Total­Remediation­Costs  , @is­Residential = @New\_is­Residential  , @is­Currently­Being­Remodeled = @New\_is­Currently­Being­Remodeled  , @has­Peeling­Chipping­Paint = @New\_has­Peeling­Chipping­Paint  , @County = @New\_­County  , @is­Rental = @New\_is­Rental  , @Year­Built = @New\_­Year­Built  , @Owner­Contact­Information = @New\_­Owner­Contact­Information  , @Property­ID = @Property­ID  IF (@New\_­Property­Notes IS NOT NULL)  BEGIN  IF @DEBUG = 1  SELECT 'EXEC [dbo].[usp\_­Insert­Property­Notes] @Property\_­ID = @Property\_­ID, @Notes = @New\_­Property­Notes, @Inserted­Notes­ID = @Notes­ID OUTPUT '  , @Property­ID, @New\_­Property­Notes  EXEC [dbo].[usp\_­Insert­Property­Notes]  @Property\_­ID = @Property­ID,  @Notes = @New\_­Property­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH;  END  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

[[dbo].[usp\_­Insert­Property­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Questionnaire] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Questionnaire­ID | int | 4 |
| @New\_­Questionnaire­Date | date | 3 |
| @New\_­Questionnaire­Data­Source­ID | int | 4 |
| @New\_­Visit­Remodeled­Property | bit | 1 |
| @New\_­Paint­Date | date | 3 |
| @New\_­Remodel­Property­Date | date | 3 |
| @New\_is­Exposedto­Peeling­Paint | bit | 1 |
| @New\_is­Taking­Vitamins | bit | 1 |
| @New\_­Nursing­Mother | bit | 1 |
| @New\_­Nursing­Infant | bit | 1 |
| @New\_­Pregnant | bit | 1 |
| @New\_is­Using­Pacifier | bit | 1 |
| @New\_is­Using­Bottle | bit | 1 |
| @New\_­Bites­Nails | bit | 1 |
| @New\_­Non­Food­Eating | bit | 1 |
| @New\_­Non­Foodin­Mouth | bit | 1 |
| @New\_­Eat­Outside | bit | 1 |
| @New\_­Suckling | bit | 1 |
| @New\_­Mouthing | bit | 1 |
| @New\_­Frequent­Hand­Washing | bit | 1 |
| @New\_­Visits­Old­Homes | bit | 1 |
| @New\_­Daycare­ID | int | 4 |
| @New\_­Notes | varchar(3000) | 3000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: William Thier  -- Create date: 20130618  -- Description: Stored Procedure to update  -- questionnaire records  -- =============================================  -- DROP PROCEDURE usp\_up­Questionnaire  CREATE PROCEDURE [dbo].[usp\_up­Questionnaire]  -- Add the parameters for the stored procedure here  @Questionnaire­ID int = NULL,  @New\_­Questionnaire­Date date = NULL,  @New\_­Questionnaire­Data­Source­ID int = NULL,  @New\_­Visit­Remodeled­Property bit = NULL,  @New\_­Paint­Date date = NULL,  @New\_­Remodel­Property­Date date = NULL,  @New\_is­Exposedto­Peeling­Paint bit = NULL,  @New\_is­Taking­Vitamins bit = NULL,  @New\_­Nursing­Mother bit = NULL,  @New\_­Nursing­Infant bit = NULL,  @New\_­Pregnant bit = NULL,  @New\_is­Using­Pacifier bit = NULL,  @New\_is­Using­Bottle bit = NULL,  @New\_­Bites­Nails bit = NULL,  @New\_­Non­Food­Eating bit = NULL,  @New\_­Non­Foodin­Mouth bit = Null,  @New\_­Eat­Outside bit = NULL,  @New\_­Suckling bit = NULL,  @New\_­Mouthing bit = NULL,  @New\_­Frequent­Hand­Washing bit = NULL,  @New\_­Visits­Old­Homes bit = NULL,  @New\_­Daycare­ID int = NULL,  @New\_­Notes varchar(3000) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  DECLARE @Error­Log­ID int, @Notes­ID int, @spupdate­Questionnairesql­Str NVARCHAR(4000);  -- insert statements for procedure here  BEGIN TRY  -- Check if Questionnaire­ID is valid, if not return  IF NOT EXISTS (SELECT Questionnaire­ID from Questionnaire where Questionnaire­ID = @Questionnaire­ID)  BEGIN  RAISERROR('Questionnaire­ID must be specified and valid', 11,-1,'usp\_up­Questionnaire');  END    -- BUILD update statement  if (@New\_­Questionnaire­Date is null)  select @New\_­Questionnaire­Date = Questionnaire­Date from Questionnaire where Questionnaire­ID = @Questionnaire­ID    SELECT @spupdate­Questionnairesql­Str = N'update Questionnaire set Questionnaire­Date = @Questionnaire­Date'  IF (@New\_­Questionnaire­Data­Source­ID IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Questionnaire­Data­Source­ID = @Questionnaire­Data­Source­ID'  IF (@New\_­Visit­Remodeled­Property IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Visit­Remodeled­Property = @Visit­Remodeled­Property'  IF (@New\_­Paint­Date IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Paint­Date = @Paint­Date'  IF (@New\_­Remodel­Property­Date IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Remodel­Property­Date = @Remodel­Property­Date'  IF (@New\_is­Exposedto­Peeling­Paint IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', is­Exposedto­Peeling­Paint = @is­Exposedto­Peeling­Paint'  IF (@New\_is­Taking­Vitamins IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', is­Taking­Vitamins = @is­Taking­Vitamins'  IF (@New\_­Nursing­Mother IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Nursing­Mother = @Nursing­Mother'  IF (@New\_­Nursing­Infant IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Nursing­Infant = @Nursing­Infant'  IF (@New\_­Pregnant IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Pregnant = @Pregnant'  IF (@New\_is­Using­Pacifier IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', is­Using­Pacifier = @is­Using­Pacifier'  IF (@New\_is­Using­Bottle IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', is­Using­Bottle = @is­Using­Bottle'  IF (@New\_­Bites­Nails IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Bites­Nails = @Bites­Nails'  IF (@New\_­Non­Food­Eating IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Non­Food­Eating = @Non­Food­Eating'  IF (@New\_­Non­Foodin­Mouth IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Non­Foodin­Mouth = @Non­Foodin­Mouth'  IF (@New\_­Eat­Outside IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Eat­Outside = @Eat­Outside'  IF (@New\_­Suckling IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Suckling = @Suckling'  IF (@New\_­Mouthing IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Mouthing = @Mouthing'    IF (@New\_­Frequent­Hand­Washing IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Frequent­Hand­Washing = @Frequent­Hand­Washing'  IF (@New\_­Visits­Old­Homes IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Visits­Old­Homes = @Visits­Old­Homes'  IF (@New\_­Daycare­ID IS NOT NULL)  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N', Day­Care­ID = @Daycare­ID'  -- make sure to only update record for specified Questionnaire  SELECT @spupdate­Questionnairesql­Str = @spupdate­Questionnairesql­Str + N' WHERE Questionnaire­ID = @Questionnaire­ID'  IF (@DEBUG = 1)  SELECT @spupdate­Questionnairesql­Str, Questionnaire­Date = @New\_­Questionnaire­Date, Questionnaire­Data­Source­ID = @New\_­Questionnaire­Data­Source­ID  , Visit­Remodeled­Property = @New\_­Visit­Remodeled­Property, Paint­Date = @New\_­Paint­Date, Remodel­Property­Date = @New\_­Remodel­Property­Date  , is­Exposedto­Peeling­Paint = @New\_is­Exposedto­Peeling­Paint, is­Taking­Vitamins = @New\_is­Taking­Vitamins, Nursing­Mother = @New\_­Nursing­Mother  , Nursing­Infant = @New\_­Nursing­Infant, Pregnant = @New\_­Pregnant, is­Using­Pacifier = @New\_is­Using­Pacifier, is­Using­Bottle = @New\_is­Using­Bottle  , Bitesnails = @New\_­Bites­Nails, Non­Food­Eating = @New\_­Non­Food­Eating, Non­Foodin­Mouth = @New\_­Non­Foodin­Mouth, Eat­Outside = @New\_­Eat­Outside  , Suckling = @New\_­Suckling, Mouthing = @New\_­Mouthing, Frequent­Hand­Washing = @New\_­Frequent­Hand­Washing, Visits­Old­Homes = @New\_­Visits­Old­Homes  , Daycare­ID = @New\_­Daycare­ID, Questionnaire­ID = @Questionnaire­ID, DEBUG = @DEBUG  EXEC [sp\_executesql] @spupdate­Questionnairesql­Str  , N'@Questionnaire­Date date, @Questionnaire­Data­Source­ID int, @Visit­Remodeled­Property bit, @Paint­Date date, @Remodel­Property­Date date  , @is­Exposedto­Peeling­Paint bit, @is­Taking­Vitamins bit, @Nursing­Mother bit, @Nursing­Infant bit, @Pregnant bit, @is­Using­Pacifier bit  , @is­Using­Bottle bit, @Bites­Nails bit, @Non­Food­Eating bit, @Non­Foodin­Mouth bit, @Eatoutside bit, @Suckling bit, @Mouthing bit  , @Frequent­Hand­Washing bit , @Visits­Old­Homes bit, @Daycare­ID int, @Questionnaire­ID int'  , @Questionnaire­Date = @New\_­Questionnaire­Date  , @Questionnairedata­Source­ID = @New\_­Questionnaire­Data­Source­ID  , @Visit­Remodeled­Property = @New\_­Visit­Remodeled­Property  , @Paint­Date = @New\_­Paint­Date  , @Remodel­Property­Date = @New\_­Remodel­Property­Date  , @is­Exposedto­Peeling­Paint = @New\_is­Exposedto­Peeling­Paint  , @is­Taking­Vitamins = @New\_is­Taking­Vitamins  , @Nursing­Mother = @New\_­Nursing­Mother  , @Nursing­Infant = @New\_­Nursing­Infant  , @Pregnant = @New\_­Pregnant  , @is­Using­Pacifier = @New\_is­Using­Pacifier  , @is­Using­Bottle = @New\_is­Using­Bottle  , @Bites­Nails = @New\_­Bites­Nails  , @Non­Food­Eating = @New\_­Non­Food­Eating  , @Non­Foodin­Mouth = @New\_­Non­Foodin­Mouth  , @Eat­Outside = @New\_­Eat­Outside  , @Suckling = @New\_­Suckling  , @Mouthing = @New\_­Mouthing  , @Frequent­Hand­Washing = @New\_­Frequent­Hand­Washing  , @Visits­Old­Homes = @New\_­Visits­Old­Homes  , @Daycare­ID = @New\_­Daycare­ID  , @Questionnaire­ID = @Questionnaire­ID  IF (@New\_­Notes IS NOT NULL)  EXEC [dbo].[usp\_­Insert­Questionnaire­Notes]  @Questionnaire\_­ID = @Questionnaire­ID,  @Notes = @New\_­Notes,  @Inserted­Notes­ID = @Notes­ID OUTPUT  END TRY  BEGIN CATCH  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  THROW  END CATCH;  END  GO |

Uses

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp\_­Insert­Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[usp\_up­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaireWebScreen)

|  |
| --- |
| [dbo].[usp\_up­Questionnaire­Web­Screen] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Questionnaire­ID | int | 4 |
| @Questionnaire­Date | date | 3 |
| @Questionnaire­Data­Source­ID | int | 4 |
| @Visit­Remodeled­Property | bit | 1 |
| @Paint­Date | date | 3 |
| @Remodel­Property­Date | date | 3 |
| @is­Exposedto­Peeling­Paint | bit | 1 |
| @is­Taking­Vitamins | bit | 1 |
| @Nursing­Mother | bit | 1 |
| @Nursing­Infant | bit | 1 |
| @Pregnant | bit | 1 |
| @is­Using­Pacifier | bit | 1 |
| @is­Using­Bottle | bit | 1 |
| @Bites­Nails | bit | 1 |
| @Non­Food­Eating | bit | 1 |
| @Non­Foodin­Mouth | bit | 1 |
| @Eat­Outside | bit | 1 |
| @Suckling | bit | 1 |
| @Mouthing | bit | 1 |
| @Frequent­Hand­Washing | bit | 1 |
| @Visits­Old­Homes | bit | 1 |
| @Daycare­ID | int | 4 |
| @Notes | varchar(3000) | 3000 |
| @DEBUG | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150618  -- Description: stored procedure to update  -- questionnaire data  -- =============================================  CREATE PROCEDURE [dbo].[usp\_up­Questionnaire­Web­Screen]  -- Add the parameters for the stored procedure here  @Questionnaire­ID int = NULL,  @Questionnaire­Date date = NULL,  @Questionnaire­Data­Source­ID int = NULL,  @Visit­Remodeled­Property bit = NULL,  @Paint­Date date = NULL,  @Remodel­Property­Date date = NULL,  @is­Exposedto­Peeling­Paint bit = NULL,  @is­Taking­Vitamins bit = NULL,  @Nursing­Mother bit = NULL,  @Nursing­Infant bit = NULL,  @Pregnant bit = NULL,  @is­Using­Pacifier bit = NULL,  @is­Using­Bottle bit = NULL,  @Bites­Nails bit = NULL,  @Non­Food­Eating bit = NULL,  @Non­Foodin­Mouth bit = Null,  @Eat­Outside bit = NULL,  @Suckling bit = NULL,  @Mouthing bit = NULL,  @Frequent­Hand­Washing bit = NULL,  @Visits­Old­Homes bit = NULL,  @Daycare­ID int = NULL,  @Notes varchar(3000) = NULL,  @DEBUG BIT = 0  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  BEGIN  DECLARE @Error­Log­ID int, @update­Questionnaire­Return­Value int;    -- If no family ID was passed in exit  IF (@Questionnaire­ID IS NULL)  BEGIN  RAISERROR ('Questionnaire ID must be supplied', 11, -1);  RETURN;  END;  BEGIN TRY  -- update questionnaire  EXEC @update­Questionnaire­Return­Value = [dbo].[usp\_up­Questionnaire]  @Questionnaire­ID = @Questionnaire­ID,  @New\_­Questionnaire­Date = @Questionnaire­Date,  @New\_­Questionnaire­Data­Source­ID = @Questionnaire­Data­Source­ID,  @New\_­Visit­Remodeled­Property = @Visit­Remodeled­Property,  @New\_­Paint­Date = @Paint­Date,  @New\_­Remodel­Property­Date = @Remodel­Property­Date,  @New\_is­Exposedto­Peeling­Paint = @is­Exposedto­Peeling­Paint,  @New\_is­Taking­Vitamins = @is­Taking­Vitamins,  @New\_­Nursing­Mother = @Nursing­Mother,  @New\_­Nursing­Infant = @Nursing­Infant,  @New\_­Pregnant = @Pregnant,  @New\_is­Using­Pacifier = @is­Using­Pacifier,  @New\_is­Using­Bottle = @is­Using­Bottle,  @New\_­Bites­Nails = @Bites­Nails,  @New\_­Non­Food­Eating = @Non­Food­Eating,  @New\_­Non­Foodin­Mouth = @Non­Foodin­Mouth,  @New\_­Eat­Outside = @Eat­Outside,  @New\_­Suckling = @Suckling,  @New\_­Mouthing = @Mouthing,  @New\_­Frequent­Hand­Washing = @Frequent­Hand­Washing,  @New\_­Visits­Old­Homes = @Visits­Old­Homes,  @New\_­Daycare­ID = @Daycare­ID,  @New\_­Notes = @Notes,  @DEBUG = @DEBUG  END TRY  BEGIN CATCH -- insert person  -- Call procedure to print error information.  EXECUTE dbo.usp­Print­Error;  -- Roll back any active or uncommittable transactions before  -- inserting information in the Error­Log.  IF XACT\_­STATE() <> 0  BEGIN  ROLLBACK TRANSACTION;  END  EXECUTE dbo.usp­Log­Error @Error­Log­ID = @Error­Log­ID OUTPUT;  RETURN ERROR\_­NUMBER()  END CATCH; -- insert new person  END  END  GO |

Uses

[[dbo].[usp\_up­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

|  |
| --- |
| [dbo].[usp­Log­Error] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @Error­Log­ID | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Log­Error]  @Error­Log­ID [int] = 0 OUTPUT -- Contains the Error­Log­ID of the row inserted  -- by usp­Log­Error in the Error­Log table.  AS  BEGIN  SET NOCOUNT ON;  -- Output parameter value of 0 indicates that error  -- information was not logged.  SET @Error­Log­ID = 0;  BEGIN TRY  -- Return if there is no error information to log.  IF ERROR\_­NUMBER() IS NULL  RETURN;  -- Return if inside an uncommittable transaction.  -- Data insertion/modification is not allowed when  -- a transaction is in an uncommittable state.  IF XACT\_­STATE() = -1  BEGIN  PRINT 'Cannot log error since the current transaction is in an uncommittable state. '  + 'Rollback the transaction before executing usp­Log­Error in order to successfully log error information.';  RETURN;  END;  INSERT [dbo].[Error­Log]  (  [User­Name],  [Error­Number],  [Error­Severity],  [Error­State],  [Error­Procedure],  [Error­Line],  [Error­Message]  )  VALUES  (  CONVERT(sysname, CURRENT\_­USER),  ERROR\_­NUMBER(),  ERROR\_­SEVERITY(),  ERROR\_­STATE(),  ERROR\_­PROCEDURE(),  ERROR\_­LINE(),  ERROR\_­MESSAGE()  );  -- Pass back the Error­Log­ID of the row inserted  SELECT @Error­Log­ID = @@IDENTITY;  END TRY  BEGIN CATCH  PRINT 'An error occurred in stored procedure usp­Log­Error: ';  EXECUTE [dbo].[usp­Print­Error];  RETURN -1;  END CATCH  END;  GO |

Uses

[[dbo].[Error­Log]](#(local)/User_databases/LCCHPDev/Tables/ErrorLog)

[[dbo].[usp­Print­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspPrintError)

Used By

[[dbo].[DELETE\_usp\_­Insert­Personto­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_InsertPersontoStatus)

[[dbo].[DELETE\_usp\_­Sl­Count­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_SlCountClients)

[[dbo].[usp\_­Insert­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessAgreement)

[[dbo].[usp\_­Insert­Access­Purpose]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessPurpose)

[[dbo].[usp\_­Insert­Area]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertArea)

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_­Insert­Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes)

[[dbo].[usp\_­Insert­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCleanupStatus)

[[dbo].[usp\_­Insert­Construction­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertConstructionType)

[[dbo].[usp\_­Insert­Contractor]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractor)

[[dbo].[usp\_­Insert­Contractorto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoProperty)

[[dbo].[usp\_­Insert­Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediation)

[[dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediationActionPlan)

[[dbo].[usp\_­Insert­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCountry)

[[dbo].[usp\_­Insert­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycare)

[[dbo].[usp\_­Insert­Daycare­Primary­Contact]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycarePrimaryContact)

[[dbo].[usp\_­Insert­Daycareto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycaretoProperty)

[[dbo].[usp\_­Insert­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployer)

[[dbo].[usp\_­Insert­Employerto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployertoProperty)

[[dbo].[usp\_­Insert­Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEnvironmentalInvestigation)

[[dbo].[usp\_­Insert­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEthnicity)

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

[[dbo].[usp\_­Insert­Family­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes)

[[dbo].[usp\_­Insert­Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber)

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

[[dbo].[usp\_­Insert­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFood)

[[dbo].[usp\_­Insert­Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFoodtoCountry)

[[dbo].[usp\_­Insert­Gift­Card]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertGiftCard)

[[dbo].[usp\_­Insert­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHobby)

[[dbo].[usp\_­Insert­Home­Remedies]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHomeRemedies)

[[dbo].[usp\_­Insert­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHouseholdSourcesofLead)

[[dbo].[usp\_­Insert­Insurance­Provider]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertInsuranceProvider)

[[dbo].[usp\_­Insert­Lab]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLab)

[[dbo].[usp\_­Insert­Lab­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLabNotes)

[[dbo].[usp\_­Insert­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLanguage)

[[dbo].[usp\_­Insert­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMedium)

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

[[dbo].[usp\_­Insert­Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResultsNotes)

[[dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewBloodLeadTestResultsWebScreen)

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_­Insert­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertOccupation)

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

[[dbo].[usp\_­Insert­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes)

[[dbo].[usp\_­Insert­Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes)

[[dbo].[usp\_­Insert­Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoAccessAgreement)

[[dbo].[usp\_­Insert­Personto­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoDaycare)

[[dbo].[usp\_­Insert­Personto­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEmployer)

[[dbo].[usp\_­Insert­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity)

[[dbo].[usp\_­Insert­Personto­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily)

[[dbo].[usp\_­Insert­Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoForeignFood)

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_­Insert­Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHomeRemedy)

[[dbo].[usp\_­Insert­Personto­Insurance]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoInsurance)

[[dbo].[usp\_­Insert­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage)

[[dbo].[usp\_­Insert­Personto­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation)

[[dbo].[usp\_­Insert­Personto­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPerson)

[[dbo].[usp\_­Insert­Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPhoneNumber)

[[dbo].[usp\_­Insert­Personto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoProperty)

[[dbo].[usp\_­Insert­Personto­Travel­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoTravelCountry)

[[dbo].[usp\_­Insert­Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes)

[[dbo].[usp\_­Insert­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber)

[[dbo].[usp\_­Insert­Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumberType)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp\_­Insert­Property­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes)

[[dbo].[usp\_­Insert­Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResults)

[[dbo].[usp\_­Insert­Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResultsNotes)

[[dbo].[usp\_­Insert­Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoCleanupStatus)

[[dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoHouseholdSourcesofLead)

[[dbo].[usp\_­Insert­Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoMedium)

[[dbo].[usp\_­Insert­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire)

[[dbo].[usp\_­Insert­Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes)

[[dbo].[usp\_­Insert­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediation)

[[dbo].[usp\_­Insert­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationActionPlan)

[[dbo].[usp\_­Insert­Remediation­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationNotes)

[[dbo].[usp\_­Insert­Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleLevelCategory)

[[dbo].[usp\_­Insert­Sample­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleType)

[[dbo].[usp\_­Insert­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertStatus)

[[dbo].[usp\_­Insert­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertTravelNotes)

[[dbo].[usp\_­SLAll­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults)

[[dbo].[usp\_­SLAll­Blood­Test­Results2]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults2)

[[dbo].[usp\_­Sl­Client­Follow­Up]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlClientFollowUp)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Count­Blood­Lead­Levels]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodLeadLevels)

[[dbo].[usp\_­Sl­Count­Blood­Tests]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodTests)

[[dbo].[usp\_­Sl­Count­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountClients)

[[dbo].[usp\_­Sl­Count­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountFamilyMembers)

[[dbo].[usp\_­Sl­Count­Home­Visit­Soil­Sample]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountHomeVisitSoilSample)

[[dbo].[usp\_­Sl­Count­New­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewClients)

[[dbo].[usp\_­Sl­Count­New­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewPeople)

[[dbo].[usp\_­Sl­Count­Nursing­Infants]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingInfants)

[[dbo].[usp\_­Sl­Count­Nursing­Mothers]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingMothers)

[[dbo].[usp\_­Sl­Count­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeople)

[[dbo].[usp\_­Sl­Count­People­By­Last­Name]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByLastName)

[[dbo].[usp\_­Sl­Count­Pregnant­Women]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPregnantWomen)

[[dbo].[usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditBloodTestResultsWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditClientInfoWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditFamilyWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Property­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditPropertyWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Questionnaire­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditQuestionnaireWebScreenInformation)

[[dbo].[usp\_­Sl­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyMembers)

[[dbo].[usp\_­Sl­Family­Nameto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyNametoProperty)

[[dbo].[usp\_­SLInserted­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedData)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

[[dbo].[usp\_­SLList­All­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListAllFamilyMembers)

[[dbo].[usp\_­Sl­List­Clients­By­Createdate]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByCreatedate)

[[dbo].[usp\_­Sl­List­Clients­By­Modifieddate]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByModifieddate)

[[dbo].[usp\_­Sl­List­Families]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilies)

[[dbo].[usp\_­Sl­List­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilyMembers)

[[dbo].[usp\_­Sl­List­Nursing­Womenby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListNursingWomenbyCreateDateRange)

[[dbo].[usp\_­Sl­List­Peopleby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPeoplebyCreateDateRange)

[[dbo].[usp\_­SLList­Potential­Duplicate­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicatePeople)

[[dbo].[usp\_­SLList­Potential­Duplicate­Properties]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicateProperties)

[[dbo].[usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPregnantWomenbyCreateDateRange)

[[dbo].[usp\_­SLMost­Recent­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLMostRecentBloodTestResults)

[[dbo].[usp\_­Sl­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersonNotes)

[[dbo].[usp\_­Sl­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoEthnicity)

[[dbo].[usp\_­Sl­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoLanguage)

[[dbo].[usp\_­Sl­Relation­Ship­Types]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlRelationShipTypes)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

[[dbo].[usp\_­Sl­Summary­Report\_­Meta­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport_MetaData)

[[dbo].[usp\_up­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults)

[[dbo].[usp\_up­Blood­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen)

[[dbo].[usp\_up­Client­Flag]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

[[dbo].[usp\_up­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily)

[[dbo].[usp\_up­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilytoProperty)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

[[dbo].[usp\_up­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upOccupation)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp\_up­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty)

[[dbo].[usp\_up­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire)

[[dbo].[usp\_up­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaireWebScreen)

|  |
| --- |
| [dbo].[usp­Print­Error] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[usp­Print­Error]  AS  BEGIN  SET NOCOUNT ON;  -- Print error information.  PRINT 'Error ' + CONVERT(varchar(50), ERROR\_­NUMBER()) +  ', Severity ' + CONVERT(varchar(5), ERROR\_­SEVERITY()) +  ', State ' + CONVERT(varchar(5), ERROR\_­STATE()) +  ', Procedure ' + ISNULL(ERROR\_­PROCEDURE(), '-') +  ', Line ' + CONVERT(varchar(5), ERROR\_­LINE());  PRINT ERROR\_­MESSAGE();  END;  GO |

Used By

[[dbo].[DELETE\_usp\_­Insert­Personto­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_InsertPersontoStatus)

[[dbo].[DELETE\_usp\_­Sl­Count­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/DELETE_usp_SlCountClients)

[[dbo].[usp\_­Insert­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessAgreement)

[[dbo].[usp\_­Insert­Access­Purpose]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertAccessPurpose)

[[dbo].[usp\_­Insert­Area]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertArea)

[[dbo].[usp\_­Insert­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResults)

[[dbo].[usp\_­Insert­Blood­Test­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertBloodTestResultsNotes)

[[dbo].[usp\_­Insert­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCleanupStatus)

[[dbo].[usp\_­Insert­Construction­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertConstructionType)

[[dbo].[usp\_­Insert­Contractor]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractor)

[[dbo].[usp\_­Insert­Contractorto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoProperty)

[[dbo].[usp\_­Insert­Contractorto­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediation)

[[dbo].[usp\_­Insert­Contractorto­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertContractortoRemediationActionPlan)

[[dbo].[usp\_­Insert­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertCountry)

[[dbo].[usp\_­Insert­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycare)

[[dbo].[usp\_­Insert­Daycare­Primary­Contact]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycarePrimaryContact)

[[dbo].[usp\_­Insert­Daycareto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertDaycaretoProperty)

[[dbo].[usp\_­Insert­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployer)

[[dbo].[usp\_­Insert­Employerto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEmployertoProperty)

[[dbo].[usp\_­Insert­Environmental­Investigation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEnvironmentalInvestigation)

[[dbo].[usp\_­Insert­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertEthnicity)

[[dbo].[usp\_­Insert­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamily)

[[dbo].[usp\_­Insert­Family­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilyNotes)

[[dbo].[usp\_­Insert­Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoPhoneNumber)

[[dbo].[usp\_­Insert­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertFamilytoProperty)

[[dbo].[usp\_­Insert­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFood)

[[dbo].[usp\_­Insert­Foreign­Foodto­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertForeignFoodtoCountry)

[[dbo].[usp\_­Insert­Gift­Card]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertGiftCard)

[[dbo].[usp\_­Insert­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHobby)

[[dbo].[usp\_­Insert­Home­Remedies]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHomeRemedies)

[[dbo].[usp\_­Insert­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertHouseholdSourcesofLead)

[[dbo].[usp\_­Insert­Insurance­Provider]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertInsuranceProvider)

[[dbo].[usp\_­Insert­Lab]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLab)

[[dbo].[usp\_­Insert­Lab­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLabNotes)

[[dbo].[usp\_­Insert­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertLanguage)

[[dbo].[usp\_­Insert­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMedium)

[[dbo].[usp\_­Insert­Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResults)

[[dbo].[usp\_­Insert­Medium­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertMediumSampleResultsNotes)

[[dbo].[usp\_­Insert­New­Blood­Lead­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewBloodLeadTestResultsWebScreen)

[[dbo].[usp\_­Insert­New­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewClientWebScreen)

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_­Insert­New­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewQuestionnaireWebScreen)

[[dbo].[usp\_­Insert­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertOccupation)

[[dbo].[usp\_­Insert­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPerson)

[[dbo].[usp\_­Insert­Person­Hobby­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonHobbyNotes)

[[dbo].[usp\_­Insert­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonNotes)

[[dbo].[usp\_­Insert­Person­Release­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonReleaseNotes)

[[dbo].[usp\_­Insert­Personto­Access­Agreement]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoAccessAgreement)

[[dbo].[usp\_­Insert­Personto­Daycare]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoDaycare)

[[dbo].[usp\_­Insert­Personto­Employer]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEmployer)

[[dbo].[usp\_­Insert­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoEthnicity)

[[dbo].[usp\_­Insert­Personto­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoFamily)

[[dbo].[usp\_­Insert­Personto­Foreign­Food]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoForeignFood)

[[dbo].[usp\_­Insert­Personto­Hobby]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHobby)

[[dbo].[usp\_­Insert­Personto­Home­Remedy]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoHomeRemedy)

[[dbo].[usp\_­Insert­Personto­Insurance]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoInsurance)

[[dbo].[usp\_­Insert­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoLanguage)

[[dbo].[usp\_­Insert­Personto­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoOccupation)

[[dbo].[usp\_­Insert­Personto­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPerson)

[[dbo].[usp\_­Insert­Personto­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoPhoneNumber)

[[dbo].[usp\_­Insert­Personto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoProperty)

[[dbo].[usp\_­Insert­Personto­Travel­Country]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersontoTravelCountry)

[[dbo].[usp\_­Insert­Person­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPersonTravelNotes)

[[dbo].[usp\_­Insert­Phone­Number]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumber)

[[dbo].[usp\_­Insert­Phone­Number­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPhoneNumberType)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

[[dbo].[usp\_­Insert­Property­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertyNotes)

[[dbo].[usp\_­Insert­Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResults)

[[dbo].[usp\_­Insert­Property­Sample­Results­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertySampleResultsNotes)

[[dbo].[usp\_­Insert­Propertyto­Cleanup­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoCleanupStatus)

[[dbo].[usp\_­Insert­Propertyto­Household­Sourcesof­Lead]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoHouseholdSourcesofLead)

[[dbo].[usp\_­Insert­Propertyto­Medium]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertPropertytoMedium)

[[dbo].[usp\_­Insert­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaire)

[[dbo].[usp\_­Insert­Questionnaire­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertQuestionnaireNotes)

[[dbo].[usp\_­Insert­Remediation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediation)

[[dbo].[usp\_­Insert­Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationActionPlan)

[[dbo].[usp\_­Insert­Remediation­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertRemediationNotes)

[[dbo].[usp\_­Insert­Sample­Level­Category]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleLevelCategory)

[[dbo].[usp\_­Insert­Sample­Type]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertSampleType)

[[dbo].[usp\_­Insert­Status]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertStatus)

[[dbo].[usp\_­Insert­Travel­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertTravelNotes)

[[dbo].[usp\_­SLAll­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults)

[[dbo].[usp\_­SLAll­Blood­Test­Results2]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLAllBloodTestResults2)

[[dbo].[usp\_­Sl­Client­Follow­Up]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlClientFollowUp)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Count­Blood­Lead­Levels]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodLeadLevels)

[[dbo].[usp\_­Sl­Count­Blood­Tests]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountBloodTests)

[[dbo].[usp\_­Sl­Count­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountClients)

[[dbo].[usp\_­Sl­Count­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountFamilyMembers)

[[dbo].[usp\_­Sl­Count­Home­Visit­Soil­Sample]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountHomeVisitSoilSample)

[[dbo].[usp\_­Sl­Count­New­Clients]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewClients)

[[dbo].[usp\_­Sl­Count­New­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNewPeople)

[[dbo].[usp\_­Sl­Count­Nursing­Infants]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingInfants)

[[dbo].[usp\_­Sl­Count­Nursing­Mothers]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountNursingMothers)

[[dbo].[usp\_­Sl­Count­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeople)

[[dbo].[usp\_­Sl­Count­People­By­Last­Name]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPeopleByLastName)

[[dbo].[usp\_­Sl­Count­Pregnant­Women]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountPregnantWomen)

[[dbo].[usp\_­Sl­Edit­Blood­Test­Results­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditBloodTestResultsWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Client­Info­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditClientInfoWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditFamilyWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Property­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditPropertyWebScreenInformation)

[[dbo].[usp\_­Sl­Edit­Questionnaire­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditQuestionnaireWebScreenInformation)

[[dbo].[usp\_­Sl­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyMembers)

[[dbo].[usp\_­Sl­Family­Nameto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlFamilyNametoProperty)

[[dbo].[usp\_­SLInserted­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedData)

[[dbo].[usp\_­SLInserted­Data­Simplified]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLInsertedDataSimplified)

[[dbo].[usp\_­SLList­All­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListAllFamilyMembers)

[[dbo].[usp\_­Sl­List­Clients­By­Createdate]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByCreatedate)

[[dbo].[usp\_­Sl­List­Clients­By­Modifieddate]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListClientsByModifieddate)

[[dbo].[usp\_­Sl­List­Families]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilies)

[[dbo].[usp\_­Sl­List­Family­Members]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListFamilyMembers)

[[dbo].[usp\_­Sl­List­Nursing­Womenby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListNursingWomenbyCreateDateRange)

[[dbo].[usp\_­Sl­List­Peopleby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPeoplebyCreateDateRange)

[[dbo].[usp\_­SLList­Potential­Duplicate­People]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicatePeople)

[[dbo].[usp\_­SLList­Potential­Duplicate­Properties]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLListPotentialDuplicateProperties)

[[dbo].[usp\_­Sl­List­Pregnant­Womenby­Create­Date­Range]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlListPregnantWomenbyCreateDateRange)

[[dbo].[usp\_­SLMost­Recent­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SLMostRecentBloodTestResults)

[[dbo].[usp\_­Sl­Person­Notes]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersonNotes)

[[dbo].[usp\_­Sl­Personto­Ethnicity]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoEthnicity)

[[dbo].[usp\_­Sl­Personto­Language]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlPersontoLanguage)

[[dbo].[usp\_­Sl­Relation­Ship­Types]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlRelationShipTypes)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

[[dbo].[usp\_­Sl­Summary­Report\_­Meta­Data]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport_MetaData)

[[dbo].[usp\_up­Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResults)

[[dbo].[usp\_up­Blood­Test­Results­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upBloodTestResultsWebScreen)

[[dbo].[usp\_up­Client­Flag]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientFlag)

[[dbo].[usp\_up­Client­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upClientWebScreen)

[[dbo].[usp\_up­Family]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamily)

[[dbo].[usp\_up­Familyto­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilytoProperty)

[[dbo].[usp\_up­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upFamilyWebScreen)

[[dbo].[usp\_up­Occupation]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upOccupation)

[[dbo].[usp\_up­Person]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upPerson)

[[dbo].[usp\_up­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upProperty)

[[dbo].[usp\_up­Questionnaire]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaire)

[[dbo].[usp\_up­Questionnaire­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_upQuestionnaireWebScreen)

[[dbo].[usp­Log­Error]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/uspLogError)

|  |
| --- |
| Scalar-valued Functions |

Objects

|  |
| --- |
| Name |
| [dbo.Remove­Special­Chars](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/RemoveSpecialChars) |
| [dbo.udf\_­Calculate­Age](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_CalculateAge) |
| [dbo.udf\_­Date­In­The­Past](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DateInThePast) |
| [dbo.udf\_­Does­Property­Exist](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_DoesPropertyExist) |
| [dbo.udf\_­Sl­Family­Phone­Number](#(local)/User_databases/LCCHPDev/Programmability/Functions/Scalar-valued_Functions/udf_SlFamilyPhoneNumber) |

|  |
| --- |
| [dbo].[Remove­Special­Chars] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |
| Schema Bound | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @s | varchar(256) | 256 |

SQL Script

|  |
| --- |
| -- Removes special characters from a string value.  -- All characters except 0-9, a-z and A-Z are removed and  -- the remaining characters are returned.  -- Author: Christian d'Heureuse, www.source-code.biz  CREATE function [dbo].[Remove­Special­Chars] (@s varchar(256)) returns varchar(256)  with schemabinding  begin  if @s is null  return null  declare @s2 varchar(256)  set @s2 = ''  declare @l int  set @l = len(@s)  declare @p int  set @p = 1  while @p <= @l begin  declare @c int  set @c = ascii(substring(@s, @p, 1))  if @c between 48 and 57 or @c between 65 and 90 or @c between 97 and 122  set @s2 = @s2 + char(@c)  set @p = @p + 1  end  if len(@s2) = 0  return null  return @s2  end  GO |

|  |
| --- |
| [dbo].[udf\_­Calculate­Age] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Birth­Date | datetime | 8 |
| @Current­Date | datetime | 8 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[udf\_­Calculate­Age]  (  @Birth­Date datetime = NULL,  @Current­Date datetime = NULL  )  RETURNS int  AS  BEGIN  IF @Birth­Date IS NULL  RETURN -1;  IF @Current­Date IS NULL  SET @Current­Date = Get­Date();  IF @Birth­Date > @Current­Date  RETURN 0  DECLARE @Age int  SELECT @Age = DATEDIFF(YY, @Birth­Date, @Current­Date) -  CASE WHEN(  (MONTH(@Birth­Date)\*100 + DAY(@Birth­Date)) >  (MONTH(@Current­Date)\*100 + DAY(@Current­Date))  ) THEN 1 ELSE 0 END  RETURN @Age  END  GO |

Used By

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[usp\_­Sl­Count­Adults]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlCountAdults)

[[dbo].[usp\_­Sl­Summary­Report]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlSummaryReport)

|  |
| --- |
| [dbo].[udf\_­Date­In­The­Past] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Check­Date | date | 3 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150220  -- Description: function to ensure date is less than current date  -- =============================================  CREATE FUNCTION [dbo].[udf\_­Date­In­The­Past]  (  -- Add the parameters for the function here  @Check­Date date  )  RETURNS bit  AS  BEGIN  -- Declare the return variable here  DECLARE @Result bit  -- Add the T-SQL statements to compute the return value here  IF (@Check­Date < Get­Date())  SET @Result = 1;  ELSE  SET @Result = 0;  -- Return the result of the function  RETURN @Result  END  GO |

Used By

[[dbo].[Blood­Test­Results]](#(local)/User_databases/LCCHPDev/Tables/BloodTestResults)

[[dbo].[Medium­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/MediumSampleResults)

[[dbo].[Person]](#(local)/User_databases/LCCHPDev/Tables/Person)

[[dbo].[Property­Sample­Results]](#(local)/User_databases/LCCHPDev/Tables/PropertySampleResults)

[[dbo].[Questionnaire]](#(local)/User_databases/LCCHPDev/Tables/Questionnaire)

[[dbo].[Remediation]](#(local)/User_databases/LCCHPDev/Tables/Remediation)

[[dbo].[Remediation­Action­Plan]](#(local)/User_databases/LCCHPDev/Tables/RemediationActionPlan)

|  |
| --- |
| [dbo].[udf\_­Does­Property­Exist] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Address­Line1 | varchar(100) | 100 |
| @Address­Line2 | varchar(100) | 100 |
| @City | varchar(50) | 50 |
| @State | char(2) | 2 |
| @Zip­Code | varchar(12) | 12 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150322  -- Description: Function to check for duplicate property  -- =============================================  CREATE FUNCTION [dbo].[udf\_­Does­Property­Exist]  (  -- Add the parameters for the function here  @Address­Line1 varchar(100),  @Address­Line2 varchar(100) = NULL,  @City varchar(50),  @State char(2),  @Zip­Code varchar(12)  )  RETURNS int  AS  BEGIN  -- Declare the return variable here  DECLARE @Property­ID int    IF (@Address­Line2 IS NULL)  SELECT @Property­ID = Property­ID from Property where  -- (dbo.Remove­Special­Chars(Address­Line1) = dbo.Remove­Special­Chars(@Address­Line1))  Address­Line1 = @Address­Line1  AND (Address­Line2 = '')  AND (City = @City )  and ([State] = @State and Zipcode = @Zip­Code)  ELSE  SELECT @Property­ID = Property­ID from Property where  --(dbo.Remove­Special­Chars(Address­Line1) = dbo.Remove­Special­Chars(@Address­Line1))  Address­Line1 = @Address­Line1  --AND (dbo.Remove­Special­Chars(Address­Line2) = dbo.Remove­Special­Chars(@Address­Line2))  AND Address­Line2 = @Address­Line2  AND (City = @City )  and ([State] = @State and Zipcode = @Zip­Code)  -- Return the result of the function  RETURN @Property­ID  END  GO |

Uses

[[dbo].[Property]](#(local)/User_databases/LCCHPDev/Tables/Property)

Used By

[[dbo].[usp\_­Insert­New­Family­Web­Screen]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertNewFamilyWebScreen)

[[dbo].[usp\_­Insert­Property]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_InsertProperty)

|  |
| --- |
| [dbo].[udf\_­Sl­Family­Phone­Number] |

Properties

|  |  |
| --- | --- |
| Property | Value |
| ANSI Nulls On | True |
| Quoted Identifier On | True |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Family\_­ID | int | 4 |
| @Phone­Number­Type­ID | tinyint | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Liam Thier  -- Create date: 20150405  -- Description: select specific phone number of  -- specified type for specific family  -- =============================================  CREATE FUNCTION [dbo].[udf\_­Sl­Family­Phone­Number]  (  -- Add the parameters for the function here  @Family\_­ID int,  @Phone­Number­Type­ID tinyint  )  RETURNS bigint  AS  BEGIN  -- Declare the return variable here  DECLARE @Phone­Number bigint--, @Phone­Number­Type­ID tinyint; --, @Family\_­ID int, @Phone­Number­Type­ID tinyint;  --SET @Phone­Number­Type­ID = 1  -- Add the T-SQL statements to compute the return value here  -- SELECT @Phone­Number = @Family\_­ID  Select @Phone­Number = [P].[Phone­Number] from [Phone­Number] AS [P]  JOIN [Familyto­Phone­Number] AS [P2N] ON [P].[Phone­Number­ID] = [P2N].[Phone­Number­ID]  JOIN [Family] AS [F] ON [P2N].[Family­ID] = [F].[Family­ID]  WHERE [F].[Family­ID] = @Family\_­ID and [P2N].[Number­Priority] = @Phone­Number­Type­ID  -- Select @Phone­Number  -- Return the result of the function  RETURN @Phone­Number  END  GO |

Uses

[[dbo].[Family]](#(local)/User_databases/LCCHPDev/Tables/Family)

[[dbo].[Familyto­Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/FamilytoPhoneNumber)

[[dbo].[Phone­Number]](#(local)/User_databases/LCCHPDev/Tables/PhoneNumber)

Used By

[[dbo].[usp\_­Sl­Edit­Family­Web­Screen­Information]](#(local)/User_databases/LCCHPDev/Programmability/Stored_Procedures/usp_SlEditFamilyWebScreenInformation)

|  |
| --- |
| Users |

Objects

|  |
| --- |
| Name |
| [app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser) |
| [WIN-1M8NQQ69OEH\SQLMaintenenace](#(local)/User_databases/LCCHPDev/Security/Users/WIN-1M8NQQ69OEH_SQLMaintenenace) |

|  |
| --- |
| app­User |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Type | Sql­User |
| Login Name | app­User |
| Default Schema | dbo |

Database Level Permissions

|  |  |
| --- | --- |
| Type | Action |
| CONNECT | Grant |

SQL Script

|  |
| --- |
| IF NOT EXISTS (SELECT \* FROM master.dbo.syslogins WHERE loginname = N'app­User')  CREATE LOGIN [app­User] WITH PASSWORD = 'p@ssw0rd'  GO  CREATE USER [app­User] FOR LOGIN [app­User]  GO |

|  |
| --- |
| WIN-1M8NQQ69OEH\SQLMaintenenace |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Type | Windows­User |
| Login Name | WIN-1M8NQQ69OEH\SQLMaintenenace |
| Default Schema | dbo |

Database Level Permissions

|  |  |
| --- | --- |
| Type | Action |
| CONNECT | Grant |

SQL Script

|  |
| --- |
| IF NOT EXISTS (SELECT \* FROM master.dbo.syslogins WHERE loginname = N'WIN-1M8NQQ69OEH\SQLMaintenenace')  CREATE LOGIN [WIN-1M8NQQ69OEH\SQLMaintenenace] FROM WINDOWS  GO  CREATE USER [WIN-1M8NQQ69OEH\SQLMaintenenace] FOR LOGIN [WIN-1M8NQQ69OEH\SQLMaintenenace]  GO |

|  |
| --- |
| Database Roles |

Objects

|  |
| --- |
| Name |
| [db\_accessadmin](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_accessadmin) |
| [db\_backupoperator](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_backupoperator) |
| [db\_datareader](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_datareader) |
| [db\_datawriter](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_datawriter) |
| [db\_ddladmin](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_ddladmin) |
| [db\_denydatareader](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_denydatareader) |
| [db\_denydatawriter](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_denydatawriter) |
| [db\_owner](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_owner) |
| [db\_securityadmin](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/db_securityadmin) |
| [public](#(local)/User_databases/LCCHPDev/Security/Roles/Database_Roles/public) |

|  |
| --- |
| db\_accessadmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

|  |
| --- |
| db\_backupoperator |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

Members

* [WIN-1M8NQQ69OEH\SQLMaintenenace](#(local)/User_databases/LCCHPDev/Security/Users/WIN-1M8NQQ69OEH_SQLMaintenenace)

SQL Script

|  |
| --- |
| EXEC sp\_addrolemember N'db\_backupoperator', N'WIN-1M8NQQ69OEH\SQLMaintenenace'  GO |

Uses

[WIN-1M8NQQ69OEH\SQLMaintenenace](#(local)/User_databases/LCCHPDev/Security/Users/WIN-1M8NQQ69OEH_SQLMaintenenace)

|  |
| --- |
| db\_datareader |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

Members

* [app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

SQL Script

|  |
| --- |
| EXEC sp\_addrolemember N'db\_datareader', N'app­User'  GO |

Uses

[app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

|  |
| --- |
| db\_datawriter |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

Members

* [app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

SQL Script

|  |
| --- |
| EXEC sp\_addrolemember N'db\_datawriter', N'app­User'  GO |

Uses

[app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

|  |
| --- |
| db\_ddladmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

|  |
| --- |
| db\_denydatareader |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

|  |
| --- |
| db\_denydatawriter |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

|  |
| --- |
| db\_owner |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

Members

* [app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

SQL Script

|  |
| --- |
| EXEC sp\_addrolemember N'db\_owner', N'app­User'  GO |

Uses

[app­User](#(local)/User_databases/LCCHPDev/Security/Users/appUser)

|  |
| --- |
| db\_securityadmin |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |

|  |
| --- |
| public |

Properties

|  |  |
| --- | --- |
| Property | Value |
| Owner | dbo |