Collaboard Database Documentation

IBV.Database



|  |  |
| --- | --- |
| Server | Web.collaboard.app |
| Author | Dennis Vroegop |
|  |  |
|  |  |

# Table of Contents

[Table of Contents 2](#_Toc256000000)

[localhost 12](#_Toc256000001)



[User databases 14](#_Toc256000002)



[IBV.Database Database 15](#_Toc256000003)



[Tables 17](#_Toc256000004)



[[auth].[t\_role] 21](#_Toc256000005)



[[auth].[t\_user] 23](#_Toc256000006)



[[auth].[t\_user\_role] 25](#_Toc256000007)



[[conversion].[t\_conversion\_booking] 27](#_Toc256000008)



[[conversion].[t\_conversion\_booking\_history] 30](#_Toc256000009)



[[conversion].[t\_convertable\_file\_type] 32](#_Toc256000010)



[[conversion].[t\_image\_resizing\_booking] 34](#_Toc256000011)



[[conversion].[t\_image\_resizing\_booking\_history] 36](#_Toc256000012)



[[conversion].[t\_state\_machine\_status] 38](#_Toc256000013)



[[dbo].[t\_blob\_status] 40](#_Toc256000014)



[[dbo].[t\_caches] 41](#_Toc256000015)



[[dbo].[t\_canvasshotter\_booking] 43](#_Toc256000016)



[[dbo].[t\_canvasshotter\_booking\_history] 45](#_Toc256000017)



[[dbo].[t\_configuration] 47](#_Toc256000018)



[[dbo].[t\_events] 49](#_Toc256000019)



[[dbo].[t\_language] 50](#_Toc256000020)



[[dbo].[t\_newsletter\_subscriber] 52](#_Toc256000021)



[[dbo].[t\_online\_user] 54](#_Toc256000022)



[[dbo].[t\_project] 57](#_Toc256000023)



[[dbo].[t\_project\_copy\_booking] 60](#_Toc256000024)



[[dbo].[t\_project\_copy\_booking\_history] 63](#_Toc256000025)



[[dbo].[t\_project\_participant] 65](#_Toc256000026)



[[dbo].[t\_project\_tag] 67](#_Toc256000027)



[[dbo].[t\_project\_type] 69](#_Toc256000028)



[[dbo].[t\_project\_type\_translation] 71](#_Toc256000029)



[[dbo].[t\_project\_upgrade\_booking] 73](#_Toc256000030)



[[dbo].[t\_project\_upgrade\_booking\_history] 75](#_Toc256000031)



[[dbo].[t\_quicklink] 77](#_Toc256000032)



[[dbo].[t\_quicklink\_tile] 79](#_Toc256000033)



[[dbo].[t\_state\_machine\_status] 81](#_Toc256000034)



[[dbo].[t\_telemetrydata] 83](#_Toc256000035)



[[dbo].[t\_template\_translation] 84](#_Toc256000036)



[[dbo].[t\_tile\_relation] 86](#_Toc256000037)



[[dbo].[t\_tile\_relation\_anchor] 90](#_Toc256000038)



[[dbo].[t\_tile\_relation\_line] 91](#_Toc256000039)



[[dbo].[t\_tile\_relation\_line\_style] 92](#_Toc256000040)



[[dbo].[t\_tile\_relation\_startend\_symbol] 93](#_Toc256000041)



[[dbo].[t\_tile\_relation\_style] 94](#_Toc256000042)



[[dbo].[t\_tile\_relation\_type] 95](#_Toc256000043)



[[dbo].[t\_tile\_type] 96](#_Toc256000044)



[[dbo].[t\_tile\_voting] 97](#_Toc256000045)



[[dbo].[t\_tilecopy\_copiedrelation] 99](#_Toc256000046)



[[dbo].[t\_tilecopy\_copiedtile] 101](#_Toc256000047)



[[dbo].[t\_tilecopy\_sourcetile] 104](#_Toc256000048)



[[dbo].[t\_tilestatus] 106](#_Toc256000049)



[[dbo].[t\_tilestatus\_copy\_booking] 111](#_Toc256000050)



[[dbo].[t\_tilestatus\_copy\_booking\_history] 114](#_Toc256000051)



[[dbo].[t\_tilestatus\_stroke] 116](#_Toc256000052)



[[dbo].[t\_tilestatus\_upgrade\_booking] 118](#_Toc256000053)



[[dbo].[t\_traces] 121](#_Toc256000054)



[[dbo].[t\_user] 122](#_Toc256000055)



[[licensing].[t\_billing\_type] 125](#_Toc256000056)



[[licensing].[t\_customer] 127](#_Toc256000057)



[[licensing].[t\_import\_data] 129](#_Toc256000058)



[[licensing].[t\_license\_file\_history] 131](#_Toc256000059)



[[licensing].[t\_object\_type] 133](#_Toc256000060)



[[licensing].[t\_order] 135](#_Toc256000061)



[[licensing].[t\_orderline] 137](#_Toc256000062)



[[licensing].[t\_payment] 139](#_Toc256000063)



[[licensing].[t\_payment\_method] 141](#_Toc256000064)



[[licensing].[t\_price] 143](#_Toc256000065)



[[licensing].[t\_pricelist] 145](#_Toc256000066)



[[licensing].[t\_product] 147](#_Toc256000067)



[[licensing].[t\_promo\_code] 149](#_Toc256000068)



[[licensing].[t\_promo\_type] 151](#_Toc256000069)



[[licensing].[t\_status] 153](#_Toc256000070)



[[licensing].[t\_subscription] 155](#_Toc256000071)



[[licensing].[t\_subscription\_device] 158](#_Toc256000072)



[[licensing].[t\_subscription\_user] 160](#_Toc256000073)



[[mft].[t\_chunkingmachine] 162](#_Toc256000074)



[[mft].[t\_chunkingmachine\_history] 165](#_Toc256000075)



[[mft].[t\_chunkingmachine\_status] 168](#_Toc256000076)



[[mft].[t\_mergemachine] 170](#_Toc256000077)



[[mft].[t\_mergemachine\_history] 173](#_Toc256000078)



[[mft].[t\_mergemachine\_status] 175](#_Toc256000079)



[[mft].[t\_state\_machine\_status] 177](#_Toc256000080)



[[mft].[t\_storagemachine\_status] 178](#_Toc256000081)



[[mft].[t\_storageoperations] 180](#_Toc256000082)



[[mft].[t\_storageoperations\_history] 183](#_Toc256000083)



[[mft].[t\_storageoperationtype] 186](#_Toc256000084)



[Stored Procedures 188](#_Toc256000085)



[[auth].[p\_user\_add\_role] 196](#_Toc256000086)



[[auth].[p\_user\_delete] 197](#_Toc256000087)



[[auth].[p\_user\_get\_by\_username] 198](#_Toc256000088)



[[auth].[p\_user\_get\_password] 200](#_Toc256000089)



[[auth].[p\_user\_get\_roles] 201](#_Toc256000090)



[[auth].[p\_user\_insert] 202](#_Toc256000091)



[[auth].[p\_user\_remove\_role] 204](#_Toc256000092)



[[auth].[p\_user\_update] 205](#_Toc256000093)



[[auth].[p\_user\_update\_password] 207](#_Toc256000094)



[[conversion].[p\_conversion\_book] 208](#_Toc256000095)



[[conversion].[p\_conversion\_book\_fallback] 210](#_Toc256000096)



[[conversion].[p\_conversion\_book\_history] 211](#_Toc256000097)



[[conversion].[p\_conversion\_insert] 213](#_Toc256000098)



[[conversion].[p\_conversion\_unbook] 215](#_Toc256000099)



[[conversion].[p\_image\_resizing\_book] 216](#_Toc256000100)



[[conversion].[p\_image\_resizing\_book\_fallback] 217](#_Toc256000101)



[[conversion].[p\_image\_resizing\_book\_history] 218](#_Toc256000102)



[[conversion].[p\_image\_resizing\_booking\_insert] 220](#_Toc256000103)



[[conversion].[p\_image\_resizing\_unbook] 222](#_Toc256000104)



[[dbo].[p\_cache\_load] 223](#_Toc256000105)



[[dbo].[p\_cache\_publishid\_get] 224](#_Toc256000106)



[[dbo].[p\_canvasshotter\_book] 225](#_Toc256000107)



[[dbo].[p\_canvasshotter\_booking\_history\_search] 226](#_Toc256000108)



[[dbo].[p\_canvasshotter\_booking\_insert\_conditional] 228](#_Toc256000109)



[[dbo].[p\_canvasshotter\_fallback] 230](#_Toc256000110)



[[dbo].[p\_canvasshotter\_history] 231](#_Toc256000111)



[[dbo].[p\_canvasshotter\_unbook] 233](#_Toc256000112)



[[dbo].[p\_configuration\_get\_all] 234](#_Toc256000113)



[[dbo].[p\_configuration\_load] 235](#_Toc256000114)



[[dbo].[p\_configuration\_set] 236](#_Toc256000115)



[[dbo].[p\_get\_all\_template\_types] 237](#_Toc256000116)



[[dbo].[p\_insert\_event] 238](#_Toc256000117)



[[dbo].[p\_insert\_trace] 239](#_Toc256000118)



[[dbo].[p\_language\_get\_all] 240](#_Toc256000119)



[[dbo].[p\_language\_insert] 241](#_Toc256000120)



[[dbo].[p\_language\_is\_id\_valid] 242](#_Toc256000121)



[[dbo].[p\_newsletter\_subscriber\_insert] 243](#_Toc256000122)



[[dbo].[p\_online\_user\_book] 245](#_Toc256000123)



[[dbo].[p\_online\_user\_check\_connected] 246](#_Toc256000124)



[[dbo].[p\_online\_user\_delete] 247](#_Toc256000125)



[[dbo].[p\_online\_user\_get\_by\_connectionid] 248](#_Toc256000126)



[[dbo].[p\_online\_user\_get\_by\_username] 250](#_Toc256000127)



[[dbo].[p\_online\_user\_get\_first\_by\_project\_id] 252](#_Toc256000128)



[[dbo].[p\_online\_user\_get\_projectid] 253](#_Toc256000129)



[[dbo].[p\_online\_user\_insert] 254](#_Toc256000130)



[[dbo].[p\_online\_user\_search\_backend] 256](#_Toc256000131)



[[dbo].[p\_online\_user\_set\_disconnected] 258](#_Toc256000132)



[[dbo].[p\_online\_user\_unbook] 259](#_Toc256000133)



[[dbo].[p\_online\_user\_unbook\_forgot] 260](#_Toc256000134)



[[dbo].[p\_online\_user\_update\_keepalive] 261](#_Toc256000135)



[[dbo].[p\_project\_copy\_book] 262](#_Toc256000136)



[[dbo].[p\_project\_copy\_fallback] 263](#_Toc256000137)



[[dbo].[p\_project\_copy\_history] 264](#_Toc256000138)



[[dbo].[p\_project\_copy\_history\_search] 266](#_Toc256000139)



[[dbo].[p\_project\_copy\_insert] 268](#_Toc256000140)



[[dbo].[p\_project\_copy\_unbook] 270](#_Toc256000141)



[[dbo].[p\_project\_delete\_template] 271](#_Toc256000142)



[[dbo].[p\_project\_get\_by\_id] 272](#_Toc256000143)



[[dbo].[p\_project\_get\_by\_userid] 274](#_Toc256000144)



[[dbo].[p\_project\_get\_connectionid\_for\_presenter] 275](#_Toc256000145)



[[dbo].[p\_project\_get\_containeruri] 276](#_Toc256000146)



[[dbo].[p\_project\_get\_coparticipating\_projects] 277](#_Toc256000147)



[[dbo].[p\_project\_get\_coparticipating\_users] 279](#_Toc256000148)



[[dbo].[p\_project\_get\_licensed\_projects] 281](#_Toc256000149)



[[dbo].[p\_project\_get\_participating\_projects] 283](#_Toc256000150)



[[dbo].[p\_project\_get\_participating\_projects\_count] 286](#_Toc256000151)



[[dbo].[p\_project\_get\_templates] 287](#_Toc256000152)



[[dbo].[p\_project\_insert] 289](#_Toc256000153)



[[dbo].[p\_project\_participant\_delete] 291](#_Toc256000154)



[[dbo].[p\_project\_participant\_get\_by\_coparticipant] 292](#_Toc256000155)



[[dbo].[p\_project\_participant\_get\_by\_projectid] 293](#_Toc256000156)



[[dbo].[p\_project\_participant\_get\_by\_projectid\_and\_userid] 294](#_Toc256000157)



[[dbo].[p\_project\_participant\_get\_by\_userid] 295](#_Toc256000158)



[[dbo].[p\_project\_participant\_insert] 296](#_Toc256000159)



[[dbo].[p\_project\_participant\_update] 297](#_Toc256000160)



[[dbo].[p\_project\_remove\_user\_as\_presenter\_in\_all] 298](#_Toc256000161)



[[dbo].[p\_project\_search\_backend] 299](#_Toc256000162)



[[dbo].[p\_project\_set\_presenter] 301](#_Toc256000163)



[[dbo].[p\_project\_setlock] 302](#_Toc256000164)



[[dbo].[p\_project\_setversion] 303](#_Toc256000165)



[[dbo].[p\_project\_tag\_delete] 304](#_Toc256000166)



[[dbo].[p\_project\_tag\_get\_by\_projectid] 305](#_Toc256000167)



[[dbo].[p\_project\_tag\_get\_by\_projectids] 306](#_Toc256000168)



[[dbo].[p\_project\_tag\_insert] 308](#_Toc256000169)



[[dbo].[p\_project\_update] 309](#_Toc256000170)



[[dbo].[p\_project\_update\_background] 310](#_Toc256000171)



[[dbo].[p\_project\_update\_canvas\_size\_ratio] 311](#_Toc256000172)



[[dbo].[p\_project\_update\_deleted] 312](#_Toc256000173)



[[dbo].[p\_project\_update\_owner] 313](#_Toc256000174)



[[dbo].[p\_project\_update\_sas\_token] 315](#_Toc256000175)



[[dbo].[p\_project\_update\_template] 316](#_Toc256000176)



[[dbo].[p\_project\_update\_thumbnail] 318](#_Toc256000177)



[[dbo].[p\_project\_update\_zoom\_offset] 319](#_Toc256000178)



[[dbo].[p\_project\_upgrade\_book] 320](#_Toc256000179)



[[dbo].[p\_project\_upgrade\_fallback] 321](#_Toc256000180)



[[dbo].[p\_project\_upgrade\_history] 322](#_Toc256000181)



[[dbo].[p\_project\_upgrade\_insert] 324](#_Toc256000182)



[[dbo].[p\_project\_upgrade\_unbook] 326](#_Toc256000183)



[[dbo].[p\_projecttypes\_convert\_to\_template] 327](#_Toc256000184)



[[dbo].[p\_projecttypes\_delete] 329](#_Toc256000185)



[[dbo].[p\_projecttypes\_get\_all\_template\_types] 330](#_Toc256000186)



[[dbo].[p\_projecttypes\_insert\_new\_projecttype] 332](#_Toc256000187)



[[dbo].[p\_projecttypes\_update] 334](#_Toc256000188)



[[dbo].[p\_projecttypes\_verify\_if\_template] 336](#_Toc256000189)



[[dbo].[p\_quicklink\_delete] 337](#_Toc256000190)



[[dbo].[p\_quicklink\_get\_by\_id] 338](#_Toc256000191)



[[dbo].[p\_quicklink\_get\_by\_projectid] 339](#_Toc256000192)



[[dbo].[p\_quicklink\_get\_tiles\_for\_quicklink] 341](#_Toc256000193)



[[dbo].[p\_quicklink\_insert] 342](#_Toc256000194)



[[dbo].[p\_quicklink\_remove\_tile\_from\_quicklinks] 344](#_Toc256000195)



[[dbo].[p\_quicklink\_update] 345](#_Toc256000196)



[[dbo].[p\_storage\_eraser\_book] 347](#_Toc256000197)



[[dbo].[p\_storage\_eraser\_unbook] 348](#_Toc256000198)



[[dbo].[p\_telemetry\_insert] 349](#_Toc256000199)



[[dbo].[p\_tile\_relation\_by\_participating\_tile] 350](#_Toc256000200)



[[dbo].[p\_tile\_relation\_delete] 353](#_Toc256000201)



[[dbo].[p\_tile\_relation\_delete\_all\_by\_tileid] 354](#_Toc256000202)



[[dbo].[p\_tile\_relation\_get\_by\_projectid] 355](#_Toc256000203)



[[dbo].[p\_tile\_relation\_get\_by\_projectid\_paged] 356](#_Toc256000204)



[[dbo].[p\_tile\_relation\_insert] 358](#_Toc256000205)



[[dbo].[p\_tile\_relation\_update] 360](#_Toc256000206)



[[dbo].[p\_tile\_relation\_with\_tile] 362](#_Toc256000207)



[[dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id] 364](#_Toc256000208)



[[dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id\_paged] 365](#_Toc256000209)



[[dbo].[p\_tile\_voting\_update] 367](#_Toc256000210)



[[dbo].[p\_tilecopy\_cleanup] 369](#_Toc256000211)



[[dbo].[p\_tilecopy\_getcopiedtiles\_by\_token] 370](#_Toc256000212)



[[dbo].[p\_tilecopy\_getrelations\_by\_token] 372](#_Toc256000213)



[[dbo].[p\_tilecopy\_getsourcetiles\_by\_token] 374](#_Toc256000214)



[[dbo].[p\_tilecopy\_insertcopiedtile] 375](#_Toc256000215)



[[dbo].[p\_tilecopy\_insertrelation] 379](#_Toc256000216)



[[dbo].[p\_tilecopy\_insertsourcetile] 381](#_Toc256000217)



[[dbo].[p\_tilestatus\_copy\_book] 383](#_Toc256000218)



[[dbo].[p\_tilestatus\_copy\_fallback] 384](#_Toc256000219)



[[dbo].[p\_tilestatus\_copy\_history] 385](#_Toc256000220)



[[dbo].[p\_tilestatus\_copy\_history\_search] 387](#_Toc256000221)



[[dbo].[p\_tilestatus\_copy\_insert] 389](#_Toc256000222)



[[dbo].[p\_tilestatus\_copy\_unbook] 391](#_Toc256000223)



[[dbo].[p\_tilestatus\_delete] 392](#_Toc256000224)



[[dbo].[p\_tilestatus\_get\_by\_id] 394](#_Toc256000225)



[[dbo].[p\_tilestatus\_get\_by\_parenttile] 396](#_Toc256000226)



[[dbo].[p\_tilestatus\_get\_by\_projectid] 398](#_Toc256000227)



[[dbo].[p\_tilestatus\_get\_by\_projectid\_paged] 400](#_Toc256000228)



[[dbo].[p\_tilestatus\_get\_by\_projectid\_with\_thumbnails] 402](#_Toc256000229)



[[dbo].[p\_tilestatus\_get\_max\_zindex] 404](#_Toc256000230)



[[dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile] 405](#_Toc256000231)



[[dbo].[p\_tilestatus\_get\_toplevel\_by\_projectid] 407](#_Toc256000232)



[[dbo].[p\_tilestatus\_get\_upgradable\_for\_project] 408](#_Toc256000233)



[[dbo].[p\_tilestatus\_gettree\_by\_id] 410](#_Toc256000234)



[[dbo].[p\_tilestatus\_insert] 413](#_Toc256000235)



[[dbo].[p\_tilestatus\_stroke\_by\_tileid] 416](#_Toc256000236)



[[dbo].[p\_tilestatus\_stroke\_insert] 417](#_Toc256000237)



[[dbo].[p\_tilestatus\_stroke\_update] 418](#_Toc256000238)



[[dbo].[p\_tilestatus\_unlock\_by\_ids] 419](#_Toc256000239)



[[dbo].[p\_tilestatus\_unlock\_by\_lockedby] 420](#_Toc256000240)



[[dbo].[p\_tilestatus\_unlock\_by\_project\_lockedby] 421](#_Toc256000241)



[[dbo].[p\_tilestatus\_unlock\_forgot] 423](#_Toc256000242)



[[dbo].[p\_tilestatus\_update] 424](#_Toc256000243)



[[dbo].[p\_tilestatus\_update\_blobstatusid] 426](#_Toc256000244)



[[dbo].[p\_tilestatus\_update\_containertype] 428](#_Toc256000245)



[[dbo].[p\_tilestatus\_update\_content] 430](#_Toc256000246)



[[dbo].[p\_tilestatus\_update\_grouped] 431](#_Toc256000247)



[[dbo].[p\_tilestatus\_update\_lock] 433](#_Toc256000248)



[[dbo].[p\_tilestatus\_update\_pinning] 435](#_Toc256000249)



[[dbo].[p\_tilestatus\_update\_position] 436](#_Toc256000250)



[[dbo].[p\_tilestatus\_update\_voting] 438](#_Toc256000251)



[[dbo].[p\_tilestatus\_update\_zindex] 439](#_Toc256000252)



[[dbo].[p\_tilestatus\_upgrade\_book] 440](#_Toc256000253)



[[dbo].[p\_tilestatus\_upgrade\_file] 441](#_Toc256000254)



[[dbo].[p\_tilestatus\_upgrade\_insert] 442](#_Toc256000255)



[[dbo].[p\_tilestatus\_upgrade\_unbook] 444](#_Toc256000256)



[[dbo].[p\_update\_blobstatus\_for\_thumbnails] 445](#_Toc256000257)



[[dbo].[p\_user\_get\_all] 446](#_Toc256000258)



[[dbo].[p\_user\_get\_by\_project] 447](#_Toc256000259)



[[dbo].[p\_user\_get\_by\_project\_count] 448](#_Toc256000260)



[[dbo].[p\_user\_get\_by\_userid] 449](#_Toc256000261)



[[dbo].[p\_user\_get\_by\_username] 450](#_Toc256000262)



[[dbo].[p\_user\_insert] 452](#_Toc256000263)



[[dbo].[p\_user\_update] 454](#_Toc256000264)



[[licensing].[p\_billing\_type\_delete] 456](#_Toc256000265)



[[licensing].[p\_billing\_type\_get\_all] 457](#_Toc256000266)



[[licensing].[p\_billing\_type\_get\_by\_id] 458](#_Toc256000267)



[[licensing].[p\_billing\_type\_insert] 459](#_Toc256000268)



[[licensing].[p\_billing\_type\_update] 460](#_Toc256000269)



[[licensing].[p\_customer\_delete] 461](#_Toc256000270)



[[licensing].[p\_customer\_get\_all] 462](#_Toc256000271)



[[licensing].[p\_customer\_get\_by\_id] 463](#_Toc256000272)



[[licensing].[p\_customer\_get\_by\_username] 464](#_Toc256000273)



[[licensing].[p\_customer\_insert] 465](#_Toc256000274)



[[licensing].[p\_customer\_search] 466](#_Toc256000275)



[[licensing].[p\_customer\_update] 467](#_Toc256000276)



[[licensing].[p\_import\_data\_book] 468](#_Toc256000277)



[[licensing].[p\_import\_data\_delete] 470](#_Toc256000278)



[[licensing].[p\_import\_data\_insert] 471](#_Toc256000279)



[[licensing].[p\_import\_data\_unbook] 473](#_Toc256000280)



[[licensing].[p\_license\_file\_history\_get\_by\_id] 474](#_Toc256000281)



[[licensing].[p\_license\_file\_history\_insert] 475](#_Toc256000282)



[[licensing].[p\_license\_file\_history\_search] 476](#_Toc256000283)



[[licensing].[p\_order\_delete] 478](#_Toc256000284)



[[licensing].[p\_order\_get\_all] 479](#_Toc256000285)



[[licensing].[p\_order\_get\_by\_id] 480](#_Toc256000286)



[[licensing].[p\_order\_insert] 481](#_Toc256000287)



[[licensing].[p\_order\_search] 483](#_Toc256000288)



[[licensing].[p\_order\_update] 485](#_Toc256000289)



[[licensing].[p\_orderline\_delete] 487](#_Toc256000290)



[[licensing].[p\_orderline\_get\_all] 488](#_Toc256000291)



[[licensing].[p\_orderline\_get\_by\_id] 489](#_Toc256000292)



[[licensing].[p\_orderline\_get\_by\_orderid] 490](#_Toc256000293)



[[licensing].[p\_orderline\_insert] 491](#_Toc256000294)



[[licensing].[p\_orderline\_update] 493](#_Toc256000295)



[[licensing].[p\_payment\_delete] 495](#_Toc256000296)



[[licensing].[p\_payment\_get\_all] 496](#_Toc256000297)



[[licensing].[p\_payment\_get\_by\_id] 497](#_Toc256000298)



[[licensing].[p\_payment\_insert] 498](#_Toc256000299)



[[licensing].[p\_payment\_method\_delete] 500](#_Toc256000300)



[[licensing].[p\_payment\_method\_get\_all] 501](#_Toc256000301)



[[licensing].[p\_payment\_method\_get\_by\_id] 502](#_Toc256000302)



[[licensing].[p\_payment\_method\_insert] 503](#_Toc256000303)



[[licensing].[p\_payment\_method\_update] 504](#_Toc256000304)



[[licensing].[p\_payment\_search] 505](#_Toc256000305)



[[licensing].[p\_payment\_update] 507](#_Toc256000306)



[[licensing].[p\_price\_delete] 508](#_Toc256000307)



[[licensing].[p\_price\_get\_all] 509](#_Toc256000308)



[[licensing].[p\_price\_get\_by\_id] 510](#_Toc256000309)



[[licensing].[p\_price\_get\_by\_pricelistid] 511](#_Toc256000310)



[[licensing].[p\_price\_insert] 512](#_Toc256000311)



[[licensing].[p\_price\_update] 514](#_Toc256000312)



[[licensing].[p\_pricelist\_delete] 515](#_Toc256000313)



[[licensing].[p\_pricelist\_get\_all] 516](#_Toc256000314)



[[licensing].[p\_pricelist\_get\_by\_id] 517](#_Toc256000315)



[[licensing].[p\_pricelist\_insert] 518](#_Toc256000316)



[[licensing].[p\_pricelist\_update] 520](#_Toc256000317)



[[licensing].[p\_product\_delete] 521](#_Toc256000318)



[[licensing].[p\_product\_get\_all] 522](#_Toc256000319)



[[licensing].[p\_product\_get\_by\_id] 523](#_Toc256000320)



[[licensing].[p\_product\_get\_by\_productfamily] 524](#_Toc256000321)



[[licensing].[p\_product\_insert] 525](#_Toc256000322)



[[licensing].[p\_product\_update] 527](#_Toc256000323)



[[licensing].[p\_promo\_code\_delete] 529](#_Toc256000324)



[[licensing].[p\_promo\_code\_get\_all] 530](#_Toc256000325)



[[licensing].[p\_promo\_code\_get\_by\_code] 531](#_Toc256000326)



[[licensing].[p\_promo\_code\_get\_by\_id] 532](#_Toc256000327)



[[licensing].[p\_promo\_code\_get\_by\_promotypeid] 533](#_Toc256000328)



[[licensing].[p\_promo\_code\_insert] 535](#_Toc256000329)



[[licensing].[p\_promo\_code\_update] 537](#_Toc256000330)



[[licensing].[p\_promo\_type\_delete] 539](#_Toc256000331)



[[licensing].[p\_promo\_type\_get\_all] 540](#_Toc256000332)



[[licensing].[p\_promo\_type\_get\_by\_id] 541](#_Toc256000333)



[[licensing].[p\_promo\_type\_insert] 542](#_Toc256000334)



[[licensing].[p\_promo\_type\_update] 543](#_Toc256000335)



[[licensing].[p\_subscription\_delete] 544](#_Toc256000336)



[[licensing].[p\_subscription\_device\_delete] 545](#_Toc256000337)



[[licensing].[p\_subscription\_device\_get\_all] 546](#_Toc256000338)



[[licensing].[p\_subscription\_device\_get\_by\_id] 547](#_Toc256000339)



[[licensing].[p\_subscription\_device\_get\_by\_subscriptionid] 548](#_Toc256000340)



[[licensing].[p\_subscription\_device\_get\_by\_subscriptionid\_devicecode] 549](#_Toc256000341)



[[licensing].[p\_subscription\_device\_insert] 550](#_Toc256000342)



[[licensing].[p\_subscription\_device\_update] 552](#_Toc256000343)



[[licensing].[p\_subscription\_get\_all] 553](#_Toc256000344)



[[licensing].[p\_subscription\_get\_by\_customerid] 554](#_Toc256000345)



[[licensing].[p\_subscription\_get\_by\_customerid\_productfamilycode] 555](#_Toc256000346)



[[licensing].[p\_subscription\_get\_by\_id] 557](#_Toc256000347)



[[licensing].[p\_subscription\_get\_by\_subscriptiondevice] 558](#_Toc256000348)



[[licensing].[p\_subscription\_get\_by\_subscriptionuser] 560](#_Toc256000349)



[[licensing].[p\_subscription\_insert] 562](#_Toc256000350)



[[licensing].[p\_subscription\_search] 564](#_Toc256000351)



[[licensing].[p\_subscription\_update] 566](#_Toc256000352)



[[licensing].[p\_subscription\_user\_delete] 568](#_Toc256000353)



[[licensing].[p\_subscription\_user\_get\_all] 569](#_Toc256000354)



[[licensing].[p\_subscription\_user\_get\_by\_id] 570](#_Toc256000355)



[[licensing].[p\_subscription\_user\_get\_by\_subscriptionid\_username] 571](#_Toc256000356)



[[licensing].[p\_subscription\_user\_insert] 572](#_Toc256000357)



[[licensing].[p\_subscription\_user\_search] 573](#_Toc256000358)



[[licensing].[p\_subscription\_user\_update] 575](#_Toc256000359)



[[mft].[p\_check\_download\_available] 576](#_Toc256000360)



[[mft].[p\_check\_storageoperation\_finished] 577](#_Toc256000361)



[[mft].[p\_check\_upload\_finished] 578](#_Toc256000362)



[[mft].[p\_chunkingmachine\_book] 579](#_Toc256000363)



[[mft].[p\_chunkingmachine\_completed] 581](#_Toc256000364)



[[mft].[p\_chunkingmachine\_insert] 582](#_Toc256000365)



[[mft].[p\_chunkingmachine\_unbook] 584](#_Toc256000366)



[[mft].[p\_cleaningmachine\_bookchunking] 586](#_Toc256000367)



[[mft].[p\_cleaningmachine\_bookmergemoving] 588](#_Toc256000368)



[[mft].[p\_cleaningmachine\_bookmerging] 590](#_Toc256000369)



[[mft].[p\_cleaningmachine\_move\_chunkinghistory] 592](#_Toc256000370)



[[mft].[p\_cleaningmachine\_move\_mergehistory] 593](#_Toc256000371)



[[mft].[p\_cleaningmachine\_unbookchunking] 594](#_Toc256000372)



[[mft].[p\_cleaningmachine\_unbookmerging] 596](#_Toc256000373)



[[mft].[p\_cleaningmachinenotcompleted\_bookchunking] 597](#_Toc256000374)



[[mft].[p\_cleaningmachinenotcompleted\_bookchunkingmoving] 599](#_Toc256000375)



[[mft].[p\_cleaningmachinenotcompleted\_bookmerging] 601](#_Toc256000376)



[[mft].[p\_configuration\_load\_documentexplorer] 603](#_Toc256000377)



[[mft].[p\_get\_activity\_from\_chunking] 604](#_Toc256000378)



[[mft].[p\_get\_activity\_from\_merge] 605](#_Toc256000379)



[[mft].[p\_get\_activity\_from\_storage] 606](#_Toc256000380)



[[mft].[p\_mergemachine\_book] 607](#_Toc256000381)



[[mft].[p\_mergemachine\_insert] 609](#_Toc256000382)



[[mft].[p\_mergemachine\_ready\_to\_process] 611](#_Toc256000383)



[[mft].[p\_mergemachine\_unbook] 612](#_Toc256000384)



[[mft].[p\_storagemachine\_book] 613](#_Toc256000385)



[[mft].[p\_storagemachine\_insert] 614](#_Toc256000386)



[[mft].[p\_storagemachine\_unbook] 616](#_Toc256000387)



[[mft].[p\_storageoperations\_move\_history] 617](#_Toc256000388)



[Scalar-valued Functions 618](#_Toc256000389)



[[conversion].[fn\_­Get­Convertable­File­Type­Id] 619](#_Toc256000390)



[[conversion].[fn\_­Get­State­Machine­Status­Id] 620](#_Toc256000391)



[[dbo].[fn\_­Get­Blob­Status] 621](#_Toc256000392)



[[dbo].[fn\_­Get­Cache­Id] 622](#_Toc256000393)



[[dbo].[fn\_­Get­Officeconverter­Machine­Status­Id] 623](#_Toc256000394)



[[dbo].[fn\_­Get­State­Machine­Status­Id] 624](#_Toc256000395)



[[dbo].[fn\_­Get­Tile­Type] 626](#_Toc256000396)



[[licensing].[fn\_­Apply­Filter] 628](#_Toc256000397)



[[mft].[fn\_­Get­Chunking­Machine­Status­Id] 629](#_Toc256000398)



[[mft].[fn\_­Get­Merge­Machine­Status­Id] 631](#_Toc256000399)



[[mft].[fn\_­Get­State­Machine­Status­Id] 633](#_Toc256000400)



[[mft].[fn\_­Get­Storage­Machine­Status­Id] 634](#_Toc256000401)



|  |
| --- |
| localhost |

Databases (1)

* [IBV.Database](#DgGLlFQTVNBJ0yLmcZ68QltSdNE=)



Server Properties

|  |  |
| --- | --- |
| Property | Value |
| Product | Microsoft SQL Server |
| Version | 15.0.4123.1 |
| Language | English |
| Platform | NT x64 |
| Edition | Developer Edition (64-bit) |
| Engine Edition | 3 (Enterprise) |
| Processors | 24 |
| OS Version | 6.2 (9200) |
| Physical Memory | 41082 |
| Is Clustered | False |
| Root Directory | C:\ |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |

Server Settings

|  |  |
| --- | --- |
| Property | Value |
| Default data file path | /var/opt/mssql/data/ |
| Default backup file path | /var/opt/mssql/data |
| Default log file path | /var/opt/mssql/data/ |
| Recovery Interval (minutes) | 0 |
| Default index fill factor | 0 |
| Default backup media retention | 0 |
| Compress Backup | False |

Advanced Server Settings

|  |  |
| --- | --- |
| Property | Value |
| 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 | 5 |
| Max degree of parallelism | 0 |
| Min server memory | 16 |
| Max server memory | 2147483647 |
| Scan for startup procs | False |
| Transform noise words | False |
| CLR enabled | False |
| Blocked process threshold | 0 |
| Filestream access level | False |
| Optimize for ad hoc workloads | False |
| CLR strict security | True |

|  |
| --- |
| User databases |

Databases (1)

* [IBV.Database](#DgGLlFQTVNBJ0yLmcZ68QltSdNE=)



|  |
| --- |
| IBV.Database Database |

Database Properties

|  |  |
| --- | --- |
| Property | Value |
| SQL Server Version | Max |
| Compatibility Level | Max |
| Last backup time | - |
| Last log backup time | - |
| Creation date | May 14 2021 |
| Users | 4 |
| Database Encryption Enabled | False |
| Database Encryption Algorithm | None |
| Database size | 80.00 MB |
| Unallocated space | 62.36 MB |

Database Options

|  |  |
| --- | --- |
| Property | Value |
| Compatibility Level | 150 |
| 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 | ee0fd386-a58c-49c6-8264-7625d43833e4 |
| Service broker enabled | True |
| Log reuse wait | OLDEST\_­PAGE |
| 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 | 60 |
| Database owner | sa |

Files

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size | Maxsize | Autogrowth | File Name |
| IBV.Database | Data | 72,00 MB | unlimited | 64,00 MB | /var/opt/mssql/data/IBV.Database\_­Primary.mdf |
| IBV.Database\_log | Log | 8,00 MB | 2048,00 GB | 64,00 MB | /var/opt/mssql/data/IBV.Database\_­Primary.ldf |

|  |
| --- |
| Tables |

Objects

|  |
| --- |
| Name |
| [auth.t\_role](#AweQXMLX6KvUKdT9zh9dn14odbs=) All available roles for authentication |
| [auth.t\_user](#DO0Gsi+pRv3WrPujaZIDUG5kUes=) All users, used by authentication |
| [auth.t\_user\_role](#z0oD8EiDELcVSPbOHQev9pusO10=) Authenticated users and their roles |
| [conversion.t\_conversion\_booking](#uTiDswOkhkFY12t13BlyECo1WqA=) Office Conversion state machine |
| [conversion.t\_conversion\_booking\_history](#Y0UXxUluDiZBREfu3Qg07kR0UW8=) Office Conversion state machine history |
| [conversion.t\_convertable\_file\_type](#tVkeUMaUYGiFb1qPFkvmeYUVCII=) Types of files that can be converted |
| [conversion.t\_image\_resizing\_booking](#PFy0f4aYM/KVhD40Db4VdEF3FDk=) Image resizer state machine |
| [conversion.t\_image\_resizing\_booking\_history](#ZTGqNCVGVuaAWpCIqJJZokIlpuM=) Image resizer state machine history |
| [conversion.t\_state\_machine\_status](#iBHbgvgyd9MrMv4uPQSI2vbyeGY=) State of state machine |
| [dbo.t\_blob\_status](#Wh7xQn2/X4EcCdSoo4kUljq/Gp8=) State of blob |
| [dbo.t\_caches](#n+xoB6o4S9rPTyl8cBF9APNvuGs=) Configuration Cache |
| [dbo.t\_canvasshotter\_booking](#jU82GEddQUIqor8nUoFVYesFEEo=) Screenshot of project state machine |
| [dbo.t\_canvasshotter\_booking\_history](#AdXuTn3PyjKlMrK2nZ5poeksshY=) Screenshot of project state machine history |
| [dbo.t\_configuration](#SlYHTPBSmXSvh/WEaW+L97YJRd8=) Configuration data |
| [dbo.t\_events](#SAhKe3M7QyyIog5mlg0zn7hNQIg=) Events |
| [dbo.t\_language](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=) Languages used for translating templates |
| [dbo.t\_newsletter\_subscriber](#AWHNaP7PiJsuDf+PCHEERpzAUn4=) Subcsribers to the newsletter |
| [dbo.t\_online\_user](#oTKipXz9q0i7/ho1w/MFw/f36nA=) A user who is currently in a project |
| [dbo.t\_project](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) All projects |
| [dbo.t\_project\_copy\_booking](#xK10ySnNmS3vqciPJwVKT7p8JzQ=) Copy project state machine |
| [dbo.t\_project\_copy\_booking\_history](#8yKLMTMt2d+nVkiLPu6HrcDuBEE=) Copy project state machine history |
| [dbo.t\_project\_participant](#20iVIf+vwX3+XobN2ZpMOACBq4o=) Users who have access to a project |
| [dbo.t\_project\_tag](#NVtxRSVjLeXCqup7XcDqnAEmXn0=) Tags for the project |
| [dbo.t\_project\_type](#GyGD46URFdCG64D0BvNZhMpw4IA=) The type of project (user project or template) |
| [dbo.t\_project\_type\_translation](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=) The project template type translation |
| [dbo.t\_project\_upgrade\_booking](#gfuU7fskjWXvIyRe4VWGy2an6Kw=) Upgrade mechanism state machine |
| [dbo.t\_project\_upgrade\_booking\_history](#lorYCHQjcbwJVwleNpEFauH2JOY=) Upgrade mechanism state machine history |
| [dbo.t\_quicklink](#8R5txRxbFoGoF2i7BOacoCDM29c=) Quicklinks in a project |
| [dbo.t\_quicklink\_tile](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=) Linking table between quicklinks and the tiles |
| [dbo.t\_state\_machine\_status](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) Status of the state machine |
| [dbo.t\_telemetrydata](#cjQE5M9B1IDxcJRkAEXqhevcxLc=) Data for telemetry |
| [dbo.t\_template\_translation](#BpdWJknsRmdVD81ROP738LoU6EU=) Translated entries for the templates |
| [dbo.t\_tile\_relation](#078YtCFOxZtrTr+ji99XjRUc40A=) Relation between tiles |
| [dbo.t\_tile\_relation\_anchor](#FcUpACp1BApta+S3+HnXkpQMt4I=) The type of anchor for a relation |
| [dbo.t\_tile\_relation\_line](#+qMkPlre1TQOnP2h8pcGrG1QxDo=) The type of line between relation |
| [dbo.t\_tile\_relation\_line\_style](#0h0UAGOGrfB/LY7u8AJHKD5ERYA=) The line styles for a relation |
| [dbo.t\_tile\_relation\_startend\_symbol](#20FtsGWM3tQuQsgV56yVep7A5B8=) The symbol used for the relation line start |
| [dbo.t\_tile\_relation\_style](#uaXVDUif5OUkaINcFrMH2xZwemo=) The style of the relation |
| [dbo.t\_tile\_relation\_type](#t9KwwwpGyY9rRoGxVre804qNj4A=) The type of the relation |
| [dbo.t\_tile\_type](#8YaY5k86JZ8/My8nxSV+Md7QE6s=) The type of the tile |
| [dbo.t\_tile\_voting](#apFAn+5wzEjWRUBbGPwi/lWybWQ=) Voting for the tile |
| [dbo.t\_tilecopy\_copiedrelation](#TpEVDYGh4J/aV/9zW2Qgqvq2Ku8=) All copied relations for a copy command |
| [dbo.t\_tilecopy\_copiedtile](#z32gliNZ8xOJvfNpb2w8q4TnC08=) All copied tiles for a copy command |
| [dbo.t\_tilecopy\_sourcetile](#Q2UYJrfAzp+CQwZGWFDTwEseGMI=) All source tiles used in a copy command |
| [dbo.t\_tilestatus](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) The tiles |
| [dbo.t\_tilestatus\_copy\_booking](#RVjh+ATj4D3aPXIlnTnzWVjMatE=) Copy of a tile state machine |
| [dbo.t\_tilestatus\_copy\_booking\_history](#c+tNoP0kvrC9ukkhGNCEvWTioxY=) Copy of a tile state machine history |
| [dbo.t\_tilestatus\_stroke](#eNq8iC4zx70GGM6EXbHSL++uzlU=) The stroke for inking in a tile |
| [dbo.t\_tilestatus\_upgrade\_booking](#NfeAlTE7NimcNyx/T33qC/risAg=) Tile update mechanism state machine |
| [dbo.t\_traces](#KsOYXNDwJr+c7gsoWx5SwI5231k=) Debug / Trace information |
| [dbo.t\_user](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) The user as used in tiles and projects |
| [licensing.t\_billing\_type](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=) The type of billing |
| [licensing.t\_customer](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=) The registered customer |
| [licensing.t\_import\_data](#yOjz4MRQxXPPTp9qgVyGmTRcv5k=) Licensing data that has been imported from an external file |
| [licensing.t\_license\_file\_history](#SIlmTMR49Mk3ZHwNuyVJLwgmdAI=) History of all licensing files |
| [licensing.t\_object\_type](#iVCrlYdN90WWswSpfUGP/SVaY7w=) Type of object used in look-up tables for licensing |
| [licensing.t\_order](#Abm+/exewd8b+kfSgBGP3mREH2k=) The orders |
| [licensing.t\_orderline](#gW/8v+TiudxKbw4411ArMKtXy2Y=) The line in an order |
| [licensing.t\_payment](#hYxAzoyBQRnrW5Nz12ue3oquKm4=) The payments made |
| [licensing.t\_payment\_method](#bq8EjbGuD1o8wQd83KAB4kZD7RE=) The methods of payment |
| [licensing.t\_price](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=) The price |
| [licensing.t\_pricelist](#lfqnc4nKM7PvrMGyuhtCjbwwf44=) The pricelist |
| [licensing.t\_product](#6EBiFkvfbRTw3F555fk44DFA3ks=) The products available for licensing |
| [licensing.t\_promo\_code](#6oqplzrZY3UacWTAEhXbFt60GSQ=) Promo codes |
| [licensing.t\_promo\_type](#XWhkhxJf9mvneWNL4gIOGe4ENts=) The types of promo code |
| [licensing.t\_status](#9aWO1xuBeb9B/Yli1IzB1MagQIA=) The status of the licensing related objects |
| [licensing.t\_subscription](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=) Subscriptions |
| [licensing.t\_subscription\_device](#lJFbLWHid2xndcdtbsATRo54g/w=) The devices allowed to use a subscription |
| [licensing.t\_subscription\_user](#jrEJsKBnLMDw32yaeZrtVt+09KU=) The users allowed to use a subscription |
| [mft.t\_chunkingmachine](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=) State machine for the chunking service (download Blobs) |
| [mft.t\_chunkingmachine\_history](#aU/tVVaTKU0VGvpN8lOEMo7OrWQ=) State machine for the chunking service history |
| [mft.t\_chunkingmachine\_status](#orTfJWER9t2qKsaqUNd9g7WUDiQ=) Status of the chunking service state machine |
| [mft.t\_mergemachine](#bu7q7+v0YyUsCxS8oi40DUowbds=) State machine for the merging service (upload Blobs) |
| [mft.t\_mergemachine\_history](#A4S2F2HPlFcxaD6FNedLE5yGu/A=) State machine for the merging service history |
| [mft.t\_mergemachine\_status](#fAT3wGUeZiRwTusB4E+KrU+Trcc=) Status of the merging service state machine |
| [mft.t\_state\_machine\_status](#Wncuy+RLAj/fIK2ZMwRnfCiyyOY=) All states for the MFT state machines |
| [mft.t\_storagemachine\_status](#fzTCNIvpwKpEL9bwvmEgtn5jvxk=) State of the Storage state machine |
| [mft.t\_storageoperations](#Y7425KmgHwq7Czxr/cHhDctuQW8=) State machine for the Storage service (file operations) |
| [mft.t\_storageoperations\_history](#Wikc4aMSZrWQB6JETj0eWwnYvU0=) State machine for the Storage service history |
| [mft.t\_storageoperationtype](#qO3E1mf5h5nTxmZrP6eixHVA/6k=) Storage state machine operation types |

|  |
| --- |
| [auth].[t\_role] |

MS\_­Description

All available roles for authentication

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:21 vrijdag 14 mei 2021 |
| Last Modified | 06:53:21 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | id | int | 4 | NOT NULL |
|  | name | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_role | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [auth].[t\_role]  (  [id] [int] NOT NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [auth].[t\_role] ADD CONSTRAINT [PK\_t\_role] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All available roles for authentication', 'SCHEMA', N'auth', 'TABLE', N't\_role', NULL, NULL  GO |

Uses

[auth]

Used By

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[[auth].[p\_user\_add\_role]](#ck5ZcQ+YNnKJyGtrky4Tlwi1VHY=)

[[auth].[p\_user\_get\_roles]](#mYS79/VnbJ6BW6+5ChFc46J5CPY=)

[[auth].[p\_user\_remove\_role]](#prGivRZpVkEHaarCD80Rh2XIA6E=)

|  |
| --- |
| [auth].[t\_user] |

MS\_­Description

All users, used by authentication

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:21 vrijdag 14 mei 2021 |
| Last Modified | 06:53:21 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | id | uniqueidentifier | 16 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | firstname | nvarchar(250) | 500 | NULL allowed |  |
|  | lastname | nvarchar(250) | 500 | NULL allowed |  |
|  | email | nvarchar(250) | 500 | NULL allowed |  |
|  | phonenumber | nvarchar(50) | 100 | NULL allowed |  |
|  | language | nvarchar(50) | 100 | NULL allowed |  |
|  | photourl | nvarchar(500) | 1000 | NULL allowed |  |
|  | creationdate | datetime | 8 | NULL allowed |  |
|  | lastlogindate | datetime | 8 | NULL allowed |  |
|  | terms\_accepted | datetime | 8 | NULL allowed |  |
|  | isverified | bit | 1 | NOT NULL | ((0)) |
|  | passwordhash | nvarchar(250) | 500 | NULL allowed |  |
|  | lockout\_enabled | bit | 1 | NOT NULL | ((1)) |
|  | lockout\_end\_date | datetime | 8 | NULL allowed |  |
|  | otp\_security\_key | nvarchar(250) | 500 | NULL allowed |  |
|  | external\_provider | nvarchar(50) | 100 | NULL allowed |  |
|  | authentication\_mode | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_user | id | True |
|  | IX\_t\_user\_username | username | True |

SQL Script

|  |
| --- |
| CREATE TABLE [auth].[t\_user]  (  [id] [uniqueidentifier] NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [firstname] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lastname] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [email] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [phonenumber] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [language] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [photourl] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NULL,  [lastlogindate] [datetime] NULL,  [terms\_accepted] [datetime] NULL,  [isverified] [bit] NOT NULL CONSTRAINT [DF\_t\_user\_isverified] DEFAULT ((0)),  [passwordhash] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockout\_enabled] [bit] NOT NULL CONSTRAINT [DF\_t\_user\_lockout\_enabled] DEFAULT ((1)),  [lockout\_end\_date] [datetime] NULL,  [otp\_security\_key] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [external\_provider] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [authentication\_mode] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [auth].[t\_user] ADD CONSTRAINT [PK\_t\_user] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [IX\_t\_user\_username] ON [auth].[t\_user] ([username]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All users, used by authentication', 'SCHEMA', N'auth', 'TABLE', N't\_user', NULL, NULL  GO |

Uses

[auth]

Used By

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[[auth].[p\_user\_add\_role]](#ck5ZcQ+YNnKJyGtrky4Tlwi1VHY=)

[[auth].[p\_user\_delete]](#KPWoxg6bGn3VOhpJdtEAD584I3I=)

[[auth].[p\_user\_get\_by\_username]](#2PVxAKdNDzeBHG1fDzq92gVGxKw=)

[[auth].[p\_user\_get\_password]](#yIEMH4Nl3SYhz4iZlZJIOvqJinM=)

[[auth].[p\_user\_get\_roles]](#mYS79/VnbJ6BW6+5ChFc46J5CPY=)

[[auth].[p\_user\_insert]](#lHytnjCFoXFrDfVlt1SmfGVhBH0=)

[[auth].[p\_user\_remove\_role]](#prGivRZpVkEHaarCD80Rh2XIA6E=)

[[auth].[p\_user\_update]](#y/hmn19lbyzRoHunXWnYBxvPeaU=)

[[auth].[p\_user\_update\_password]](#yMG4qktZRlbNCOQri9AyYQaYh4Y=)

|  |
| --- |
| [auth].[t\_user\_role] |

MS\_­Description

Authenticated users and their roles

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 1 |
| Created | 06:53:21 vrijdag 14 mei 2021 |
| Last Modified | 06:53:21 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | userid | uniqueidentifier | 16 | NOT NULL |
|  | roleid | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_user\_role | userid, roleid | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_user\_role\_t\_role | roleid->[[auth].[t\_role].[id]](#AweQXMLX6KvUKdT9zh9dn14odbs=) |
| FK\_t\_user\_role\_t\_user | userid->[[auth].[t\_user].[id]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=) |

SQL Script

|  |
| --- |
| CREATE TABLE [auth].[t\_user\_role]  (  [userid] [uniqueidentifier] NOT NULL,  [roleid] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [auth].[t\_user\_role] ADD CONSTRAINT [PK\_t\_user\_role] PRIMARY KEY CLUSTERED ([userid], [roleid]) ON [PRIMARY]  GO  ALTER TABLE [auth].[t\_user\_role] ADD CONSTRAINT [FK\_t\_user\_role\_t\_role] FOREIGN KEY ([roleid]) REFERENCES [auth].[t\_role] ([id])  GO  ALTER TABLE [auth].[t\_user\_role] ADD CONSTRAINT [FK\_t\_user\_role\_t\_user] FOREIGN KEY ([userid]) REFERENCES [auth].[t\_user] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Authenticated users and their roles', 'SCHEMA', N'auth', 'TABLE', N't\_user\_role', NULL, NULL  GO |

Uses

[[auth].[t\_role]](#AweQXMLX6KvUKdT9zh9dn14odbs=)

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

Used By

[[auth].[p\_user\_add\_role]](#ck5ZcQ+YNnKJyGtrky4Tlwi1VHY=)

[[auth].[p\_user\_delete]](#KPWoxg6bGn3VOhpJdtEAD584I3I=)

[[auth].[p\_user\_get\_roles]](#mYS79/VnbJ6BW6+5ChFc46J5CPY=)

[[auth].[p\_user\_remove\_role]](#prGivRZpVkEHaarCD80Rh2XIA6E=)

|  |
| --- |
| [conversion].[t\_conversion\_booking] |

MS\_­Description

Office Conversion state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | id | int | 4 | NOT NULL | 1 - 1 |  |
|  | entityid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | updatedby | nvarchar(200) | 400 | NULL allowed |  |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | domain | nvarchar(200) | 400 | NULL allowed |  |  |
|  | username | nvarchar(200) | 400 | NOT NULL |  |  |
|  | filename | nvarchar(250) | 500 | NOT NULL |  |  |
|  | destinationfolder | nvarchar(250) | 500 | NULL allowed |  |  |
|  | convertablefiletypeid | int | 4 | NOT NULL |  | ((1)) |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | state­Machinestatusid | int | 4 | NOT NULL |  | ((0)) |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_conver\_\_3213E83F8EEC23CD | id | True |
|  | IX\_t\_conversion\_booking\_statemachinestatusid | state­Machinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| t\_conversion\_booking\_t\_convertable\_file\_type | convertablefiletypeid->[[conversion].[t\_convertable\_file\_type].[convertablefiletypeid]](#tVkeUMaUYGiFb1qPFkvmeYUVCII=) |
| t\_conversion\_booking\_t\_state\_machine\_status | state­Machinestatusid->[[conversion].[t\_state\_machine\_status].[statemachinestatusid]](#iBHbgvgyd9MrMv4uPQSI2vbyeGY=) |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_conversion\_booking]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [entityid] [uniqueidentifier] NOT NULL,  [updatedby] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_conversion\_booking\_creationdate] DEFAULT (getutcdate()),  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_\_t\_convers\_\_retry\_\_00AA174D] DEFAULT ((0)),  [domain] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [destinationfolder] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [convertablefiletypeid] [int] NOT NULL CONSTRAINT [DF\_t\_conversion\_booking\_convertablefiletype] DEFAULT ((1)),  [projectid] [int] NOT NULL,  [state­Machinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_conversion\_booking\_statemachinestatusid] DEFAULT ((0)),  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_conversion\_booking] ADD CONSTRAINT [PK\_\_t\_conver\_\_3213E83F8EEC23CD] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_conversion\_booking\_statemachinestatusid] ON [conversion].[t\_conversion\_booking] ([state­Machinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_conversion\_booking] ADD CONSTRAINT [t\_conversion\_booking\_t\_convertable\_file\_type] FOREIGN KEY ([convertablefiletypeid]) REFERENCES [conversion].[t\_convertable\_file\_type] ([convertablefiletypeid])  GO  ALTER TABLE [conversion].[t\_conversion\_booking] ADD CONSTRAINT [t\_conversion\_booking\_t\_state\_machine\_status] FOREIGN KEY ([state­Machinestatusid]) REFERENCES [conversion].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Office Conversion state machine', 'SCHEMA', N'conversion', 'TABLE', N't\_conversion\_booking', NULL, NULL  GO |

Uses

[[conversion].[t\_convertable\_file\_type]](#tVkeUMaUYGiFb1qPFkvmeYUVCII=)

[[conversion].[t\_state\_machine\_status]](#iBHbgvgyd9MrMv4uPQSI2vbyeGY=)

[conversion]

Used By

[[conversion].[p\_conversion\_book]](#2ky7nZ3brsF0UI6fkxfmgT8GZqI=)

[[conversion].[p\_conversion\_book\_fallback]](#N/g2rojWTtdssTZ449KkIIx/4B8=)

[[conversion].[p\_conversion\_book\_history]](#Fndejxz7Ohz6SURqC1Fm5gQZiYQ=)

[[conversion].[p\_conversion\_insert]](#ZAAI0TdWV+PMPe0e2gKthXBu6VY=)

[[conversion].[p\_conversion\_unbook]](#VOMFu7g1zPWBppeP3UenCTegvVk=)

|  |
| --- |
| [conversion].[t\_conversion\_booking\_history] |

MS\_­Description

Office Conversion state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | id | int | 4 | NOT NULL |  |
|  | entityid | uniqueidentifier | 16 | NOT NULL |  |
|  | updatedby | nvarchar(200) | 400 | NULL allowed |  |
|  | creationdate | datetime | 8 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
|  | domain | nvarchar(200) | 400 | NULL allowed |  |
|  | username | nvarchar(200) | 400 | NOT NULL |  |
|  | filename | nvarchar(250) | 500 | NOT NULL |  |
|  | destinationfolder | nvarchar(250) | 500 | NULL allowed |  |
|  | convertablefiletypeid | int | 4 | NOT NULL |  |
|  | projectid | int | 4 | NOT NULL |  |
|  | state­Machinestatusid | int | 4 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_conver\_\_3213E83F06B78952 | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_conversion\_booking\_history]  (  [id] [int] NOT NULL,  [entityid] [uniqueidentifier] NOT NULL,  [updatedby] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NOT NULL,  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_\_t\_convers\_\_retry\_\_056ECC6A] DEFAULT ((0)),  [domain] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [destinationfolder] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [convertablefiletypeid] [int] NOT NULL,  [projectid] [int] NOT NULL,  [state­Machinestatusid] [int] NOT NULL,  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_conversion\_booking\_history] ADD CONSTRAINT [PK\_\_t\_conver\_\_3213E83F06B78952] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Office Conversion state machine history', 'SCHEMA', N'conversion', 'TABLE', N't\_conversion\_booking\_history', NULL, NULL  GO |

Uses

[conversion]

Used By

[[conversion].[p\_conversion\_book\_history]](#Fndejxz7Ohz6SURqC1Fm5gQZiYQ=)

|  |
| --- |
| [conversion].[t\_convertable\_file\_type] |

MS\_­Description

Types of files that can be converted

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 5 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | convertablefiletypeid | int | 4 | NOT NULL |
|  | description | nvarchar(100) | 200 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_convertable\_file\_type | convertablefiletypeid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_convertable\_file\_type]  (  [convertablefiletypeid] [int] NOT NULL,  [description] [nvarchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_convertable\_file\_type] ADD CONSTRAINT [PK\_t\_convertable\_file\_type] PRIMARY KEY CLUSTERED ([convertablefiletypeid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Types of files that can be converted', 'SCHEMA', N'conversion', 'TABLE', N't\_convertable\_file\_type', NULL, NULL  GO |

Uses

[conversion]

Used By

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

|  |
| --- |
| [conversion].[t\_image\_resizing\_booking] |

MS\_­Description

Image resizer state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | imageresizingbookingid | int | 4 | NOT NULL | 1 - 1 |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | updatedby | nvarchar(200) | 400 | NULL allowed |  |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | file­Name | nvarchar(250) | 500 | NULL allowed |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | state­Machinestatusid | int | 4 | NOT NULL |  | ((0)) |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_image\_resizing\_booking | imageresizingbookingid | True |
|  | IX\_t\_image\_resizing\_booking\_statemachinestatusid | state­Machinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_image\_resizing\_booking\_statemachinestatusid | state­Machinestatusid->[[conversion].[t\_state\_machine\_status].[statemachinestatusid]](#iBHbgvgyd9MrMv4uPQSI2vbyeGY=) |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_image\_resizing\_booking]  (  [imageresizingbookingid] [int] NOT NULL IDENTITY(1, 1),  [tileid] [uniqueidentifier] NOT NULL,  [updatedby] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_image\_resizing\_creationdate] DEFAULT (getutcdate()),  [lastupdate] [datetime] NULL,  [file­Name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_\_t\_image\_r\_\_retry\_\_7CD98669] DEFAULT ((0)),  [projectid] [int] NOT NULL,  [state­Machinestatusid] [int] NOT NULL CONSTRAINT [DF\_\_t\_image\_r\_\_state\_\_7DCDAAA2] DEFAULT ((0)),  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_image\_resizing\_booking] ADD CONSTRAINT [PK\_t\_image\_resizing\_booking] PRIMARY KEY CLUSTERED ([imageresizingbookingid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_image\_resizing\_booking\_statemachinestatusid] ON [conversion].[t\_image\_resizing\_booking] ([state­Machinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_image\_resizing\_booking] ADD CONSTRAINT [FK\_t\_image\_resizing\_booking\_statemachinestatusid] FOREIGN KEY ([state­Machinestatusid]) REFERENCES [conversion].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Image resizer state machine', 'SCHEMA', N'conversion', 'TABLE', N't\_image\_resizing\_booking', NULL, NULL  GO |

Uses

[[conversion].[t\_state\_machine\_status]](#iBHbgvgyd9MrMv4uPQSI2vbyeGY=)

[conversion]

Used By

[[conversion].[p\_image\_resizing\_book]](#8H9bGc3/Fy7+H4uFbIgZhvy1agY=)

[[conversion].[p\_image\_resizing\_book\_fallback]](#s8FUIxTi+6LBh8mYurqnSND9xLQ=)

[[conversion].[p\_image\_resizing\_book\_history]](#1xpeiE5gtXIQzKAuzLeH5MY+gHc=)

[[conversion].[p\_image\_resizing\_booking\_insert]](#lJxGVVKhAjqYDYdtS1TeWbGi0Ko=)

[[conversion].[p\_image\_resizing\_unbook]](#RlPaP7QJJSbXEeDJQW4D5ItTc0Q=)

|  |
| --- |
| [conversion].[t\_image\_resizing\_booking\_history] |

MS\_­Description

Image resizer state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | imageresizingbookingid | int | 4 | NOT NULL |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | updatedby | nvarchar(200) | 400 | NULL allowed |  |
|  | creationdate | datetime | 8 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | file­Name | nvarchar(250) | 500 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
|  | projectid | int | 4 | NOT NULL |  |
|  | state­Machinestatusid | int | 4 | NOT NULL | ((0)) |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_image\_\_\_1D6DA3FD05CCFB3B | imageresizingbookingid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_image\_resizing\_booking\_history]  (  [imageresizingbookingid] [int] NOT NULL,  [tileid] [uniqueidentifier] NOT NULL,  [updatedby] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NOT NULL,  [lastupdate] [datetime] NULL,  [file­Name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_\_t\_image\_r\_\_retry\_\_038683F8] DEFAULT ((0)),  [projectid] [int] NOT NULL,  [state­Machinestatusid] [int] NOT NULL CONSTRAINT [DF\_\_t\_image\_r\_\_state\_\_047AA831] DEFAULT ((0)),  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_image\_resizing\_booking\_history] ADD CONSTRAINT [PK\_\_t\_image\_\_\_1D6DA3FD05CCFB3B] PRIMARY KEY CLUSTERED ([imageresizingbookingid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Image resizer state machine history', 'SCHEMA', N'conversion', 'TABLE', N't\_image\_resizing\_booking\_history', NULL, NULL  GO |

Uses

[conversion]

Used By

[[conversion].[p\_image\_resizing\_book\_history]](#1xpeiE5gtXIQzKAuzLeH5MY+gHc=)

|  |
| --- |
| [conversion].[t\_state\_machine\_status] |

MS\_­Description

State of state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 5 |
| Created | 06:53:15 vrijdag 14 mei 2021 |
| Last Modified | 06:53:15 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | statemachinestatusid | int | 4 | NOT NULL |
|  | statemachinestatusdescription | nvarchar(50) | 100 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_state\_machine\_status | statemachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [conversion].[t\_state\_machine\_status]  (  [statemachinestatusid] [int] NOT NULL,  [statemachinestatusdescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [conversion].[t\_state\_machine\_status] ADD CONSTRAINT [PK\_t\_state\_machine\_status] PRIMARY KEY CLUSTERED ([statemachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State of state machine', 'SCHEMA', N'conversion', 'TABLE', N't\_state\_machine\_status', NULL, NULL  GO |

Uses

[conversion]

Used By

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

|  |
| --- |
| [dbo].[t\_blob\_status] |

MS\_­Description

State of blob

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 7 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | blobstatusid | int | 4 | NOT NULL |
|  | blobdescription | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_blob\_status | blobstatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_blob\_status]  (  [blobstatusid] [int] NOT NULL,  [blobdescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_blob\_status] ADD CONSTRAINT [PK\_t\_blob\_status] PRIMARY KEY CLUSTERED ([blobstatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State of blob ', 'SCHEMA', N'dbo', 'TABLE', N't\_blob\_status', NULL, NULL  GO |

Used By

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[t\_caches] |

MS\_­Description

Configuration Cache

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 2 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | cacheid | smallint | 2 | NOT NULL |
|  | publishid | uniqueidentifier | 16 | NOT NULL |
|  | pollingtimeinseconds | smallint | 2 | NOT NULL |
|  | cachename | nvarchar(50) | 100 | NOT NULL |
|  | cachedescription | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_caches\_\_59913613F81D6554 | cacheid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_caches]  (  [cacheid] [smallint] NOT NULL,  [publishid] [uniqueidentifier] NOT NULL,  [pollingtimeinseconds] [smallint] NOT NULL,  [cachename] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [cachedescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_caches] ADD CONSTRAINT [PK\_\_t\_caches\_\_59913613F81D6554] PRIMARY KEY CLUSTERED ([cacheid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Configuration Cache', 'SCHEMA', N'dbo', 'TABLE', N't\_caches', NULL, NULL  GO |

Used By

[[dbo].[p\_cache\_load]](#x5NpJBEk3hMDw/mO6nOObbXG3r4=)

[[dbo].[p\_cache\_publishid\_get]](#1PcgqfvGzy3UkcWN7LQKIzudc+Y=)

[[dbo].[p\_configuration\_load]](#4A8WPG9EXmbdkLYr/w11wlL7oSw=)

[[dbo].[p\_configuration\_set]](#DeIdr/PSElcRt6bzhuI+IlR1rlU=)

|  |
| --- |
| [dbo].[t\_canvasshotter\_booking] |

MS\_­Description

Screenshot of project state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | canvasshotterid | int | 4 | NOT NULL | 1 - 1 |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | userid | int | 4 | NOT NULL |  |  |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | lastupdate | datetime | 8 | NOT NULL |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL |  | ((1)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_canvasshotter\_booking | canvasshotterid | True |
|  | nci\_wi\_t\_project\_7ADAB38961944CA62EFA69AF2981E8A3 | bookedby, bookingdate |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_canvasshotter\_booking\_projectid | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| FK\_t\_canvasshotter\_booking\_statemachinestatusid | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |
| FK\_t\_canvasshotter\_booking\_userid | userid->[[dbo].[t\_user].[userid]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_canvasshotter\_booking]  (  [canvasshotterid] [int] NOT NULL IDENTITY(1, 1),  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_canvasshotter\_booking\_creationdate] DEFAULT (getutcdate()),  [userid] [int] NOT NULL,  [projectid] [int] NOT NULL,  [lastupdate] [datetime] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_canvasshotter\_booking\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_canvasshotter\_booking\_statemachinestatusid] DEFAULT ((1))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_canvasshotter\_booking] ADD CONSTRAINT [PK\_t\_canvasshotter\_booking] PRIMARY KEY CLUSTERED ([canvasshotterid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [nci\_wi\_t\_project\_7ADAB38961944CA62EFA69AF2981E8A3] ON [dbo].[t\_canvasshotter\_booking] ([bookedby], [bookingdate]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_canvasshotter\_booking] ADD CONSTRAINT [FK\_t\_canvasshotter\_booking\_projectid] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_canvasshotter\_booking] ADD CONSTRAINT [FK\_t\_canvasshotter\_booking\_statemachinestatusid] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  ALTER TABLE [dbo].[t\_canvasshotter\_booking] ADD CONSTRAINT [FK\_t\_canvasshotter\_booking\_userid] FOREIGN KEY ([userid]) REFERENCES [dbo].[t\_user] ([userid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Screenshot of project state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_canvasshotter\_booking', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

Used By

[[dbo].[p\_canvasshotter\_book]](#GBecoS5NNIUwnYY10IG98Y4ntws=)

[[dbo].[p\_canvasshotter\_booking\_insert\_conditional]](#5iHeTvzxxnH/YtLAt0wKOhWh8Ls=)

[[dbo].[p\_canvasshotter\_fallback]](#w7g5EjCSqGzhgN064FP5YdNO6GM=)

[[dbo].[p\_canvasshotter\_history]](#N8r5uy8daNeo8TkSqrrpdiwTXaU=)

[[dbo].[p\_canvasshotter\_unbook]](#L/BBCL4V4hSpLE86zujRBjY+MW8=)

|  |
| --- |
| [dbo].[t\_canvasshotter\_booking\_history] |

MS\_­Description

Screenshot of project state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Heap | True |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Nullability |
| canvasshotterid | int | 4 | NOT NULL |
| creationdate | datetime | 8 | NULL allowed |
| userid | int | 4 | NOT NULL |
| projectid | int | 4 | NOT NULL |
| lastupdate | datetime | 8 | NOT NULL |
| bookedby | uniqueidentifier | 16 | NULL allowed |
| bookingdate | datetime | 8 | NULL allowed |
| retry | int | 4 | NOT NULL |
| statemachinestatusid | int | 4 | NOT NULL |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_canvasshotter\_booking\_history]  (  [canvasshotterid] [int] NOT NULL,  [creationdate] [datetime] NULL,  [userid] [int] NOT NULL,  [projectid] [int] NOT NULL,  [lastupdate] [datetime] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL,  [statemachinestatusid] [int] NOT NULL  ) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Screenshot of project state machine history', 'SCHEMA', N'dbo', 'TABLE', N't\_canvasshotter\_booking\_history', NULL, NULL  GO |

Used By

[[dbo].[p\_canvasshotter\_booking\_history\_search]](#94xtpURKbcm+rjwpqeLkew5J9lc=)

[[dbo].[p\_canvasshotter\_history]](#N8r5uy8daNeo8TkSqrrpdiwTXaU=)

|  |
| --- |
| [dbo].[t\_configuration] |

MS\_­Description

Configuration data

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 129 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | key | nvarchar(100) | 200 | NOT NULL |  |
|  | value | nvarchar(max) | max | NOT NULL |  |
|  | description | nvarchar(250) | 500 | NOT NULL |  |
|  | App­Domain | nvarchar(50) | 100 | NOT NULL | (N'\*') |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_tmp\_ms\_x\_\_­DFD83CAE83521136 | key | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_configuration]  (  [key] [nvarchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [value] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [App­Domain] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL CONSTRAINT [DF\_t\_configuration\_­App­Domain] DEFAULT (N'\*')  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_configuration] ADD CONSTRAINT [PK\_\_tmp\_ms\_x\_\_­DFD83CAE83521136] PRIMARY KEY CLUSTERED ([key]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Configuration data', 'SCHEMA', N'dbo', 'TABLE', N't\_configuration', NULL, NULL  GO |

Used By

[[dbo].[p\_configuration\_get\_all]](#EsJYab4euWPOJA6iZ0wTBFk/Gm8=)

[[dbo].[p\_configuration\_load]](#4A8WPG9EXmbdkLYr/w11wlL7oSw=)

[[dbo].[p\_configuration\_set]](#DeIdr/PSElcRt6bzhuI+IlR1rlU=)

|  |
| --- |
| [dbo].[t\_events] |

MS\_­Description

Events

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Heap | True |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
| ID | int | 4 | NOT NULL | 1 - 1 |  |
| Message | nvarchar(max) | max | NULL allowed |  |  |
| Params | nvarchar(max) | max | NULL allowed |  |  |
| Last­Update | datetime | 8 | NULL allowed |  | (getutcdate()) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_events]  (  [ID] [int] NOT NULL IDENTITY(1, 1),  [Message] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Params] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Last­Update] [datetime] NULL CONSTRAINT [DF\_t\_events\_­Last­Update] DEFAULT (getutcdate())  ) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Events', 'SCHEMA', N'dbo', 'TABLE', N't\_events', NULL, NULL  GO |

Used By

[[dbo].[p\_insert\_event]](#xN4F4KQcQp+aOyu0EvbP2rFSyLE=)

|  |
| --- |
| [dbo].[t\_language] |

MS\_­Description

Languages used for translating templates

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 1 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | language­Id | int | 4 | NOT NULL | 1 - 1 |  |
|  | language­Code | nvarchar(25) | 50 | NULL allowed |  | (N'en/US') |
|  | lcid | int | 4 | NULL allowed |  |  |
|  | name\_english | nvarchar(250) | 500 | NULL allowed |  |  |
|  | name\_native | nvarchar(250) | 500 | NULL allowed |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_language | language­Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_language]  (  [language­Id] [int] NOT NULL IDENTITY(1, 1),  [language­Code] [nvarchar] (25) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL CONSTRAINT [DF\_t\_language\_language­Code] DEFAULT (N'en/US'),  [lcid] [int] NULL,  [name\_english] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [name\_native] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_language] ADD CONSTRAINT [PK\_t\_language] PRIMARY KEY CLUSTERED ([language­Id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Languages used for translating templates', 'SCHEMA', N'dbo', 'TABLE', N't\_language', NULL, NULL  GO |

Used By

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

[[dbo].[p\_language\_get\_all]](#X9O6KKdbCG9O6uoQedQsHiv8C/E=)

[[dbo].[p\_language\_insert]](#IT7Hy/qMDQo++95jr0UXosszJys=)

[[dbo].[p\_language\_is\_id\_valid]](#HBZCyP1MDpjLjZZ3HMhxiyNqAKA=)

|  |
| --- |
| [dbo].[t\_newsletter\_subscriber] |

MS\_­Description

Subcsribers to the newsletter

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | email | nvarchar(250) | 500 | NOT NULL |  |
|  | firstname | nvarchar(50) | 100 | NULL allowed |  |
|  | lastname | nvarchar(50) | 100 | NULL allowed |  |
|  | language | nvarchar(50) | 100 | NULL allowed |  |
|  | creationdate | datetime | 8 | NOT NULL |  |
|  | is\_active | bit | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_newsletter\_subscriber | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_newsletter\_subscriber]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [email] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [firstname] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lastname] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [language] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NOT NULL,  [is\_active] [bit] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_newsletter\_subscriber] ADD CONSTRAINT [PK\_t\_newsletter\_subscriber] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Subcsribers to the newsletter', 'SCHEMA', N'dbo', 'TABLE', N't\_newsletter\_subscriber', NULL, NULL  GO |

Used By

[[dbo].[p\_newsletter\_subscriber\_insert]](#34hs/68I0Kq5v/3iyTP3SkEzeXQ=)

|  |
| --- |
| [dbo].[t\_online\_user] |

MS\_­Description

A user who is currently in a project

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | connectionid | nvarchar(50) | 100 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NOT NULL |  |
|  | creationdate | datetime | 8 | NOT NULL | (getutcdate()) |
|  | userid | int | 4 | NOT NULL |  |
|  | projectid | int | 4 | NULL allowed |  |
|  | logindate | datetime | 8 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NOT NULL |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL | ((1)) |
|  | ispresenting | bit | 1 | NOT NULL | ((0)) |
|  | isdisconnected | bit | 1 | NOT NULL | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_online\_user | connectionid | True |
|  | IX\_­FK\_t\_online\_user\_t\_project | projectid |  |
|  | IX\_­FK\_t\_online\_user\_t\_state\_machine\_status | statemachinestatusid |  |
|  | IX\_­FK\_t\_online\_user\_t\_user | userid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_online\_user\_t\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| FK\_t\_online\_user\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |
| FK\_t\_online\_user\_t\_user | userid->[[dbo].[t\_user].[userid]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_online\_user]  (  [connectionid] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [machineid] [uniqueidentifier] NOT NULL,  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_online\_user\_creationdate] DEFAULT (getutcdate()),  [userid] [int] NOT NULL,  [projectid] [int] NULL,  [logindate] [datetime] NOT NULL,  [lastupdate] [datetime] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_online\_user\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_online\_user\_statemachinestatusid] DEFAULT ((1)),  [ispresenting] [bit] NOT NULL CONSTRAINT [DF\_\_t\_online\_\_\_ispre\_\_787EE5A0] DEFAULT ((0)),  [isdisconnected] [bit] NOT NULL CONSTRAINT [DF\_\_t\_online\_\_\_isdis\_\_797309D9] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_online\_user] ADD CONSTRAINT [PK\_t\_online\_user] PRIMARY KEY CLUSTERED ([connectionid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_online\_user\_t\_project] ON [dbo].[t\_online\_user] ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_online\_user\_t\_state\_machine\_status] ON [dbo].[t\_online\_user] ([statemachinestatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_online\_user\_t\_user] ON [dbo].[t\_online\_user] ([userid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_online\_user] ADD CONSTRAINT [FK\_t\_online\_user\_t\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_online\_user] ADD CONSTRAINT [FK\_t\_online\_user\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  ALTER TABLE [dbo].[t\_online\_user] ADD CONSTRAINT [FK\_t\_online\_user\_t\_user] FOREIGN KEY ([userid]) REFERENCES [dbo].[t\_user] ([userid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'A user who is currently in a project', 'SCHEMA', N'dbo', 'TABLE', N't\_online\_user', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

Used By

[[dbo].[p\_online\_user\_book]](#c5iw4Jj06eWGv8RAQS1V5t4MAv8=)

[[dbo].[p\_online\_user\_check\_connected]](#TKm9+V+71Fc5TxTKUdeoAk8SXOw=)

[[dbo].[p\_online\_user\_delete]](#TBplcYBoKypjbhjavP5K5dOlL1c=)

[[dbo].[p\_online\_user\_get\_by\_connectionid]](#3hdddy7qEOm97dR0hNzxuENEE04=)

[[dbo].[p\_online\_user\_get\_by\_username]](#LGVJsJlf1/SngrX4ixAa0Bh5h/s=)

[[dbo].[p\_online\_user\_get\_first\_by\_project\_id]](#g/xQegeHFdihIcAOJ+yFPZuRE+o=)

[[dbo].[p\_online\_user\_get\_projectid]](#NL9mm0bUGUv+swcXYJoRiYj6zW8=)

[[dbo].[p\_online\_user\_insert]](#qv2u/8StiVLEVqLtKYLw5mU2W1s=)

[[dbo].[p\_online\_user\_search\_backend]](#Djf8Cny5kYj9llD1X8X5SouWFZI=)

[[dbo].[p\_online\_user\_set\_disconnected]](#6aV4Lug75izZ/3woIWF+ktDiCm4=)

[[dbo].[p\_online\_user\_unbook]](#m8zhIuLkW2eQZExBU+OrYX9Q4P8=)

[[dbo].[p\_online\_user\_unbook\_forgot]](#CR8dfDUDNElJfvYHsUNNh1QwKbk=)

[[dbo].[p\_online\_user\_update\_keepalive]](#RGffZq+MHbZHzhLGgmYB7+RxXLQ=)

[[dbo].[p\_project\_get\_connectionid\_for\_presenter]](#9ossGhPV6Hp0q1Wns3apYI72crQ=)

[[dbo].[p\_tilestatus\_unlock\_forgot]](#8SPblYSof1I8LItXwQ2VeLi22IY=)

[[dbo].[p\_tilestatus\_update]](#rEGjz6syz+nr2U/s/JA4/83VTuE=)

[[dbo].[p\_tilestatus\_update\_content]](#ovOnybcjSuIEeGh3AadnsW3fAak=)

[[dbo].[p\_tilestatus\_update\_lock]](#aczDQAYDZF2HlDSzivZSrAuFtOc=)

[[dbo].[p\_tilestatus\_update\_position]](#rVRoekoodnhEE3EEZ9teUKHN6pM=)

|  |
| --- |
| [dbo].[t\_project] |

MS\_­Description

All projects

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | projectid | int | 4 | NOT NULL | 1 - 1 |  |
|  | containeruri | nvarchar(500) | 1000 | NOT NULL |  |  |
|  | createdby | int | 4 | NOT NULL |  |  |
|  | creationdate | datetime | 8 | NOT NULL |  |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | presenter | nvarchar(250) | 500 | NULL allowed |  |  |
|  | description | nvarchar(250) | 500 | NOT NULL |  | (' ') |
|  | createdbyuniquemachineid | uniqueidentifier | 16 | NOT NULL |  | ('00000000-0000-0000-0000-000000000000') |
|  | updatedby | int | 4 | NULL allowed |  |  |
|  | zoomfactor | float | 8 | NOT NULL |  | ((0)) |
|  | offsetx | float | 8 | NOT NULL |  | ((0)) |
|  | offsety | float | 8 | NOT NULL |  | ((0)) |
|  | deleted | bit | 1 | NOT NULL |  | ((0)) |
|  | projecttypeid | int | 4 | NOT NULL |  | ((0)) |
|  | thumbnail | varbinary(max) | max | NULL allowed |  |  |
|  | backgroundcolor | nvarchar(7) | 14 | NULL allowed |  |  |
|  | canvas\_size\_ratio | int | 4 | NULL allowed |  |  |
|  | upgradeversion | int | 4 | NOT NULL |  | ((1)) |
|  | islocked | bit | 1 | NOT NULL |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_project | projectid | True |
|  | nci\_wi\_t\_project\_9ADAB38961944CA62EFA69AF2981E8A3 | createdby, deleted |  |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project]  (  [projectid] [int] NOT NULL IDENTITY(1, 1),  [containeruri] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [createdby] [int] NOT NULL,  [creationdate] [datetime] NOT NULL,  [lastupdate] [datetime] NULL,  [presenter] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL CONSTRAINT [DF\_t\_project\_description] DEFAULT (' '),  [createdbyuniquemachineid] [uniqueidentifier] NOT NULL CONSTRAINT [DF\_t\_project\_createdbyuniquemachineid] DEFAULT ('00000000-0000-0000-0000-000000000000'),  [updatedby] [int] NULL,  [zoomfactor] [float] NOT NULL CONSTRAINT [DF\_t\_project\_zoomfactor] DEFAULT ((0)),  [offsetx] [float] NOT NULL CONSTRAINT [DF\_t\_project\_offsetx] DEFAULT ((0)),  [offsety] [float] NOT NULL CONSTRAINT [DF\_t\_project\_offsety] DEFAULT ((0)),  [deleted] [bit] NOT NULL CONSTRAINT [DF\_t\_project\_deleted\_1] DEFAULT ((0)),  [projecttypeid] [int] NOT NULL CONSTRAINT [DF\_t\_project\_projecttypeid] DEFAULT ((0)),  [thumbnail] [varbinary] (max) NULL,  [backgroundcolor] [nvarchar] (7) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [canvas\_size\_ratio] [int] NULL,  [upgradeversion] [int] NOT NULL CONSTRAINT [DF\_\_t\_project\_\_upgra\_\_7E37BEF6] DEFAULT ((1)),  [islocked] [bit] NOT NULL CONSTRAINT [DF\_\_t\_project\_\_isloc\_\_7F2BE32F] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project] ADD CONSTRAINT [PK\_t\_project] PRIMARY KEY CLUSTERED ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [nci\_wi\_t\_project\_9ADAB38961944CA62EFA69AF2981E8A3] ON [dbo].[t\_project] ([createdby], [deleted]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All projects', 'SCHEMA', N'dbo', 'TABLE', N't\_project', NULL, NULL  GO |

Used By

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[p\_online\_user\_search\_backend]](#Djf8Cny5kYj9llD1X8X5SouWFZI=)

[[dbo].[p\_project\_delete\_template]](#h6ZHMHXodjZQ7/r28dA/xoJe5Ic=)

[[dbo].[p\_project\_get\_by\_id]](#KGz8phqiKp5J5UrsWlI+RqfLiH8=)

[[dbo].[p\_project\_get\_by\_userid]](#f+k/6T5SoQt87MBYuQ+KyD5PD58=)

[[dbo].[p\_project\_get\_connectionid\_for\_presenter]](#9ossGhPV6Hp0q1Wns3apYI72crQ=)

[[dbo].[p\_project\_get\_containeruri]](#wHz01VXthrjWCG2k8L5EJx+O8+U=)

[[dbo].[p\_project\_get\_coparticipating\_projects]](#D8T2UsmAddF87lP7WVAHYzS213o=)

[[dbo].[p\_project\_get\_coparticipating\_users]](#Qw+YCRyiWNeg8ihfRXKeBlbOdso=)

[[dbo].[p\_project\_get\_licensed\_projects]](#ceaCreTDTNvU0ICOTDx6kd9f8/Q=)

[[dbo].[p\_project\_get\_participating\_projects]](#y6U32mvf6Ydonzjyz9Alrl4pM2g=)

[[dbo].[p\_project\_get\_participating\_projects\_count]](#zIPBmWH0udwZ+u/O1fQlR71fyDk=)

[[dbo].[p\_project\_get\_templates]](#Kx+UI3EG+DPsEGR/IXReULy0pqM=)

[[dbo].[p\_project\_insert]](#DF2GFjYrkWYMFByvqV+nQcUlbKI=)

[[dbo].[p\_project\_remove\_user\_as\_presenter\_in\_all]](#eeHk7j2e7OVwoMFeLX3kgTe63Vk=)

[[dbo].[p\_project\_search\_backend]](#xL4aGoOVsBGhh2gahaLWnst60v4=)

[[dbo].[p\_project\_set\_presenter]](#gacANbG/B4xV95rDVlQFT5dl+wQ=)

[[dbo].[p\_project\_setlock]](#XoZhwvaT1d/nS9CAEnKSYKp2IH4=)

[[dbo].[p\_project\_setversion]](#gylQ5y5AlOemxenGqzVrVAeMMUc=)

[[dbo].[p\_project\_update]](#HnARsWtIToI2HkIeehqGO8xLPYU=)

[[dbo].[p\_project\_update\_background]](#tVCiNBHhV9nbMY5h705o5/e2aTY=)

[[dbo].[p\_project\_update\_canvas\_size\_ratio]](#X9EZEEciQlSU00elRen/dNjE7SA=)

[[dbo].[p\_project\_update\_deleted]](#nb5QQNpPcSK5w5nOBulGHodgG1w=)

[[dbo].[p\_project\_update\_owner]](#OfB+9BtUWlve8RbDyeNuNPzX3yc=)

[[dbo].[p\_project\_update\_sas\_token]](#RKyUApgnw8W6pXmnH6bJzqDSX5Y=)

[[dbo].[p\_project\_update\_template]](#puatJhq48ml0PZR3M/9OUxn1CLY=)

[[dbo].[p\_project\_update\_thumbnail]](#QUvLEBPm045EqTjq3BhBwvvubbY=)

[[dbo].[p\_project\_update\_zoom\_offset]](#nYA8gp7ufhzJqRDzS9UMHz+um64=)

[[dbo].[p\_projecttypes\_convert\_to\_template]](#JwgwHPiObJe/EYScukAUlCcGImA=)

|  |
| --- |
| [dbo].[t\_project\_copy\_booking] |

MS\_­Description

Copy project state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 07:45:07 maandag 17 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | Id | int | 4 | NOT NULL | 1 - 1 |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | description | nvarchar(250) | 500 | NULL allowed |  |  |
|  | isfromtemplate | bit | 1 | NOT NULL |  | ((0)) |
|  | clientconnectionid | nvarchar(50) | 100 | NOT NULL |  |  |
|  | username | nvarchar(200) | 400 | NOT NULL |  |  |
|  | appver | nvarchar(50) | 100 | NOT NULL |  |  |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL |  | ((0)) |
|  | xoffset | float | 8 | NULL allowed |  | ((0.0)) |
|  | yoffset | float | 8 | NULL allowed |  | ((0.0)) |
|  | templateid | int | 4 | NULL allowed |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_projec\_\_3214EC07E8388610 | Id | True |
|  | IX\_t\_project\_copy\_booking\_statemachinestatusid | statemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| t\_project\_copy\_booking\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_copy\_booking]  (  [Id] [int] NOT NULL IDENTITY(1, 1),  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_project\_copy\_booking\_creationdate] DEFAULT (getutcdate()),  [projectid] [int] NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [isfromtemplate] [bit] NOT NULL CONSTRAINT [DF\_t\_project\_copy\_booking\_isfromtemplate] DEFAULT ((0)),  [clientconnectionid] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [appver] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_project\_copy\_booking\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_project\_copy\_booking\_statemachinestatusid] DEFAULT ((0)),  [xoffset] [float] NULL CONSTRAINT [DF\_\_t\_project\_\_xoffs\_\_06CD04F7] DEFAULT ((0.0)),  [yoffset] [float] NULL CONSTRAINT [DF\_\_t\_project\_\_yoffs\_\_07C12930] DEFAULT ((0.0)),  [templateid] [int] NULL CONSTRAINT [DF\_\_t\_project\_\_templ\_\_2A363CC5] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_copy\_booking] ADD CONSTRAINT [PK\_\_t\_projec\_\_3214EC07E8388610] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_project\_copy\_booking\_statemachinestatusid] ON [dbo].[t\_project\_copy\_booking] ([statemachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_copy\_booking] ADD CONSTRAINT [t\_project\_copy\_booking\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Copy project state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_copy\_booking', NULL, NULL  GO |

Uses

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

Used By

[[dbo].[p\_project\_copy\_book]](#Xc2uZUUHsCqQzKVu6toSxfLcKL8=)

[[dbo].[p\_project\_copy\_fallback]](#SZSN716jO+FTCfcstEifkqpd04k=)

[[dbo].[p\_project\_copy\_history]](#iP9nGEt2cxagal93V/gZLHBLNeA=)

[[dbo].[p\_project\_copy\_insert]](#IElOCneHQ59gypn3QPnkEKG6pSo=)

[[dbo].[p\_project\_copy\_unbook]](#OIWAcYC3Mg2Um97d273ICxc0pwc=)

|  |
| --- |
| [dbo].[t\_project\_copy\_booking\_history] |

MS\_­Description

Copy project state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | Id | int | 4 | NOT NULL |  |
|  | creationdate | datetime | 8 | NOT NULL |  |
|  | projectid | int | 4 | NOT NULL |  |
|  | description | nvarchar(250) | 500 | NULL allowed |  |
|  | isfromtemplate | bit | 1 | NOT NULL |  |
|  | clientconnectionid | nvarchar(50) | 100 | NOT NULL |  |
|  | username | nvarchar(200) | 400 | NOT NULL |  |
|  | appver | nvarchar(50) | 100 | NOT NULL |  |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL |  |
|  | statemachinestatusid | int | 4 | NOT NULL |  |
|  | xoffset | float | 8 | NULL allowed | ((0.0)) |
|  | yoffset | float | 8 | NULL allowed | ((0.0)) |
|  | templateid | int | 4 | NULL allowed | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_projec\_\_3214EC07FBB3D762 | Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_copy\_booking\_history]  (  [Id] [int] NOT NULL,  [creationdate] [datetime] NOT NULL,  [projectid] [int] NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [isfromtemplate] [bit] NOT NULL,  [clientconnectionid] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [appver] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL,  [statemachinestatusid] [int] NOT NULL,  [xoffset] [float] NULL CONSTRAINT [DF\_\_t\_project\_\_xoffs\_\_0B91BA14] DEFAULT ((0.0)),  [yoffset] [float] NULL CONSTRAINT [DF\_\_t\_project\_\_yoffs\_\_0D7A0286] DEFAULT ((0.0)),  [templateid] [int] NULL CONSTRAINT [DF\_\_t\_project\_\_templ\_\_0C85DE4D] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_copy\_booking\_history] ADD CONSTRAINT [PK\_\_t\_projec\_\_3214EC07FBB3D762] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Copy project state machine history', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_copy\_booking\_history', NULL, NULL  GO |

Used By

[[dbo].[p\_project\_copy\_history]](#iP9nGEt2cxagal93V/gZLHBLNeA=)

[[dbo].[p\_project\_copy\_history\_search]](#SaV7osn7TKY5pGBFnNs0lyjeMxw=)

|  |
| --- |
| [dbo].[t\_project\_participant] |

MS\_­Description

Users who have access to a project

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 2 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | projectid | int | 4 | NOT NULL |  |
|  | userid | int | 4 | NOT NULL |  |
|  | permission | tinyint | 1 | NOT NULL |  |
|  | lastaccessdate | datetime | 8 | NOT NULL | (getutcdate()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_project\_participant\_1 | projectid, userid | True |
|  | IX\_­FK\_t\_project\_participant\_projectid | projectid |  |
|  | IX\_t\_project\_participant\_userid | userid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_project\_participant\_projectid | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| t\_project\_participant\_userid | userid->[[dbo].[t\_user].[userid]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_participant]  (  [projectid] [int] NOT NULL,  [userid] [int] NOT NULL,  [permission] [tinyint] NOT NULL,  [lastaccessdate] [datetime] NOT NULL CONSTRAINT [DF\_\_t\_project\_\_lasta\_\_0E6E26BF] DEFAULT (getutcdate())  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_participant] ADD CONSTRAINT [PK\_t\_project\_participant\_1] PRIMARY KEY CLUSTERED ([projectid], [userid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_project\_participant\_projectid] ON [dbo].[t\_project\_participant] ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_project\_participant\_userid] ON [dbo].[t\_project\_participant] ([userid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_participant] ADD CONSTRAINT [FK\_t\_project\_participant\_projectid] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_project\_participant] ADD CONSTRAINT [t\_project\_participant\_userid] FOREIGN KEY ([userid]) REFERENCES [dbo].[t\_user] ([userid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Users who have access to a project', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_participant', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

Used By

[[dbo].[p\_project\_get\_coparticipating\_projects]](#D8T2UsmAddF87lP7WVAHYzS213o=)

[[dbo].[p\_project\_get\_coparticipating\_users]](#Qw+YCRyiWNeg8ihfRXKeBlbOdso=)

[[dbo].[p\_project\_get\_licensed\_projects]](#ceaCreTDTNvU0ICOTDx6kd9f8/Q=)

[[dbo].[p\_project\_get\_participating\_projects]](#y6U32mvf6Ydonzjyz9Alrl4pM2g=)

[[dbo].[p\_project\_get\_participating\_projects\_count]](#zIPBmWH0udwZ+u/O1fQlR71fyDk=)

[[dbo].[p\_project\_participant\_delete]](#qLGpKwd5+XtoBnbBN/kM9Pz0ZpI=)

[[dbo].[p\_project\_participant\_get\_by\_coparticipant]](#JCJRheGtspnhunwKoYA/h8irV3I=)

[[dbo].[p\_project\_participant\_get\_by\_projectid]](#E+DU3Nso4w1mgOq/VwtiREaUIrI=)

[[dbo].[p\_project\_participant\_get\_by\_projectid\_and\_userid]](#kxWIu4dGjs334k0/sp7ExWmQERg=)

[[dbo].[p\_project\_participant\_get\_by\_userid]](#Cvz0fhSf+WGHZxztYeigtKrc5qg=)

[[dbo].[p\_project\_participant\_insert]](#P+CWh+/pskzvWIPXLquLWExDBaY=)

[[dbo].[p\_project\_participant\_update]](#/Qukp870xZ16Zg9oI3zYOVxn2fw=)

[[dbo].[p\_project\_search\_backend]](#xL4aGoOVsBGhh2gahaLWnst60v4=)

[[dbo].[p\_project\_update\_owner]](#OfB+9BtUWlve8RbDyeNuNPzX3yc=)

[[dbo].[p\_projecttypes\_convert\_to\_template]](#JwgwHPiObJe/EYScukAUlCcGImA=)

[[dbo].[p\_user\_get\_by\_project]](#pLBDyYqwrvyK2QW29bKsZG0rbnY=)

[[dbo].[p\_user\_get\_by\_project\_count]](#NPhxaTlIkusKkdZ6XY0kKAydQWs=)

|  |
| --- |
| [dbo].[t\_project\_tag] |

MS\_­Description

Tags for the project

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | projectid | int | 4 | NOT NULL |
|  | tag | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_project\_tag | projectid, tag | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_project\_tag\_t\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_tag]  (  [projectid] [int] NOT NULL,  [tag] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_tag] ADD CONSTRAINT [PK\_t\_project\_tag] PRIMARY KEY CLUSTERED ([projectid], [tag]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_tag] ADD CONSTRAINT [FK\_t\_project\_tag\_t\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tags for the project', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_tag', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

Used By

[[dbo].[p\_project\_get\_participating\_projects]](#y6U32mvf6Ydonzjyz9Alrl4pM2g=)

[[dbo].[p\_project\_tag\_delete]](#3z8R9RcW5Wjwzu5hfm2oJLVRbqc=)

[[dbo].[p\_project\_tag\_get\_by\_projectid]](#3W7c4wEtBdaF/TMWjiRDP1vahwg=)

[[dbo].[p\_project\_tag\_get\_by\_projectids]](#x3lNVjzNt+wEIiEXQvAI2D8+6QQ=)

[[dbo].[p\_project\_tag\_insert]](#3vvsElx+lZXKCjcNsf3X5c+aQpY=)

|  |
| --- |
| [dbo].[t\_project\_type] |

MS\_­Description

The type of project (user project or template)

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | projecttypeid | int | 4 | NOT NULL |
|  | description | nvarchar(250) | 500 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_project\_type | projecttypeid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_type]  (  [projecttypeid] [int] NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_type] ADD CONSTRAINT [PK\_t\_project\_type] PRIMARY KEY CLUSTERED ([projecttypeid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of project (user project or template)', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_type', NULL, NULL  GO |

Used By

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

[[dbo].[p\_get\_all\_template\_types]](#QKiRgmfSorkRF8B3TsazZeJPlDI=)

[[dbo].[p\_projecttypes\_delete]](#70ejMUHyGW7VE9h4zbtOWktjRas=)

[[dbo].[p\_projecttypes\_get\_all\_template\_types]](#Hqsb1aBo2hQhIDxbDm0pmRoxiog=)

[[dbo].[p\_projecttypes\_insert\_new\_projecttype]](#6m8B+ayGYiWnCRtFav0XhiYLSOQ=)

[[dbo].[p\_projecttypes\_update]](#r8wawLDNZrHLDtPZFcORekZDs4M=)

[[dbo].[p\_projecttypes\_verify\_if\_template]](#10qW7rShvQlXAgL1Fh1vVBvRPVQ=)

|  |
| --- |
| [dbo].[t\_project\_type\_translation] |

MS\_­Description

The project template type translation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 1 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | translationid | int | 4 | NOT NULL | 1 - 1 |
|  | projecttypeid | int | 4 | NOT NULL |  |
|  | languageid | int | 4 | NOT NULL |  |
|  | description | nvarchar(250) | 500 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_project\_type\_translation | translationid | True |
|  | IX\_t\_project\_type\_translation\_language | languageid |  |
|  | IX\_t\_project\_type\_translation\_project\_type | projecttypeid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_project\_type\_translation\_t\_language | languageid->[[dbo].[t\_language].[language­Id]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=) |
| FK\_t\_project\_type\_translation\_t\_project\_type | projecttypeid->[[dbo].[t\_project\_type].[projecttypeid]](#GyGD46URFdCG64D0BvNZhMpw4IA=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_type\_translation]  (  [translationid] [int] NOT NULL IDENTITY(1, 1),  [projecttypeid] [int] NOT NULL,  [languageid] [int] NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_type\_translation] ADD CONSTRAINT [PK\_t\_project\_type\_translation] PRIMARY KEY CLUSTERED ([translationid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_project\_type\_translation\_language] ON [dbo].[t\_project\_type\_translation] ([languageid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_project\_type\_translation\_project\_type] ON [dbo].[t\_project\_type\_translation] ([projecttypeid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_type\_translation] ADD CONSTRAINT [FK\_t\_project\_type\_translation\_t\_language] FOREIGN KEY ([languageid]) REFERENCES [dbo].[t\_language] ([language­Id])  GO  ALTER TABLE [dbo].[t\_project\_type\_translation] ADD CONSTRAINT [FK\_t\_project\_type\_translation\_t\_project\_type] FOREIGN KEY ([projecttypeid]) REFERENCES [dbo].[t\_project\_type] ([projecttypeid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The project template type translation', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_type\_translation', NULL, NULL  GO |

Uses

[[dbo].[t\_language]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=)

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

Used By

[[dbo].[p\_projecttypes\_delete]](#70ejMUHyGW7VE9h4zbtOWktjRas=)

[[dbo].[p\_projecttypes\_get\_all\_template\_types]](#Hqsb1aBo2hQhIDxbDm0pmRoxiog=)

[[dbo].[p\_projecttypes\_insert\_new\_projecttype]](#6m8B+ayGYiWnCRtFav0XhiYLSOQ=)

[[dbo].[p\_projecttypes\_update]](#r8wawLDNZrHLDtPZFcORekZDs4M=)

|  |
| --- |
| [dbo].[t\_project\_upgrade\_booking] |

MS\_­Description

Upgrade mechanism state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | Id | int | 4 | NOT NULL | 1 - 1 |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | username | nvarchar(200) | 400 | NOT NULL |  |  |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | currentversion | int | 4 | NOT NULL |  |  |
|  | desiredversion | int | 4 | NOT NULL |  |  |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_projec\_\_3214EC07606140C7 | Id | True |
|  | IX\_t\_project\_upgrade\_booking\_statemachinestatusid | statemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| t\_project\_upgrade\_booking\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_upgrade\_booking]  (  [Id] [int] NOT NULL IDENTITY(1, 1),  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_project\_upgrade\_booking\_creationdate] DEFAULT (getutcdate()),  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [projectid] [int] NOT NULL,  [currentversion] [int] NOT NULL,  [desiredversion] [int] NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [lastupdate] [datetime] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_project\_upgrade\_booking\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_project\_upgrade\_booking\_statemachinestatusid] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_upgrade\_booking] ADD CONSTRAINT [PK\_\_t\_projec\_\_3214EC07606140C7] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_project\_upgrade\_booking\_statemachinestatusid] ON [dbo].[t\_project\_upgrade\_booking] ([statemachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_upgrade\_booking] ADD CONSTRAINT [t\_project\_upgrade\_booking\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Upgrade mechanism state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_upgrade\_booking', NULL, NULL  GO |

Uses

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

Used By

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

[[dbo].[p\_project\_upgrade\_book]](#ZtIHMxzkIaQqTAum/wUN5RfUzwk=)

[[dbo].[p\_project\_upgrade\_fallback]](#eyFMA0mMdoB7kA1BgRbWM2aeqpE=)

[[dbo].[p\_project\_upgrade\_history]](#TRfRFOGkYYT9dyRx+1BlOHqMJnk=)

[[dbo].[p\_project\_upgrade\_insert]](#2XDAj09j2hETAOl2Ohb7ZFRhJUU=)

[[dbo].[p\_project\_upgrade\_unbook]](#9/vewVcF04vLmiCKTfJh/bhSZrU=)

|  |
| --- |
| [dbo].[t\_project\_upgrade\_booking\_history] |

MS\_­Description

Upgrade mechanism state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Id | int | 4 | NOT NULL |
|  | creationdate | datetime | 8 | NOT NULL |
|  | username | nvarchar(200) | 400 | NOT NULL |
|  | projectid | int | 4 | NOT NULL |
|  | currentversion | int | 4 | NOT NULL |
|  | desiredversion | int | 4 | NOT NULL |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |
|  | lastupdate | datetime | 8 | NULL allowed |
|  | bookingdate | datetime | 8 | NULL allowed |
|  | retry | int | 4 | NOT NULL |
|  | statemachinestatusid | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_projec\_\_3214EC07BF041967 | Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_project\_upgrade\_booking\_history]  (  [Id] [int] NOT NULL,  [creationdate] [datetime] NOT NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [projectid] [int] NOT NULL,  [currentversion] [int] NOT NULL,  [desiredversion] [int] NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [lastupdate] [datetime] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL,  [statemachinestatusid] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_project\_upgrade\_booking\_history] ADD CONSTRAINT [PK\_\_t\_projec\_\_3214EC07BF041967] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Upgrade mechanism state machine history', 'SCHEMA', N'dbo', 'TABLE', N't\_project\_upgrade\_booking\_history', NULL, NULL  GO |

Used By

[[dbo].[p\_project\_upgrade\_history]](#TRfRFOGkYYT9dyRx+1BlOHqMJnk=)

|  |
| --- |
| [dbo].[t\_quicklink] |

MS\_­Description

Quicklinks in a project

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | quicklinkid | int | 4 | NOT NULL | 1 - 1 |  |
|  | projectid | int | 4 | NOT NULL |  |  |
|  | createdby | int | 4 | NOT NULL |  |  |
|  | creationdate | datetime | 8 | NOT NULL |  |  |
|  | lastupdate | datetime | 8 | NOT NULL |  |  |
|  | lastupdatedby | int | 4 | NOT NULL |  |  |
|  | thumbnail | varbinary(max) | max | NULL allowed |  |  |
|  | name | nvarchar(250) | 500 | NULL allowed |  |  |
|  | createdbyuniquemachineid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | isdeleted | bit | 1 | NOT NULL |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_quickl\_\_­C76188A2EB00B22B | quicklinkid | True |
|  | IX\_t\_quicklink\_project | projectid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_quicklink\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| FK\_t\_quicklink\_user | createdby->[[dbo].[t\_user].[userid]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_quicklink]  (  [quicklinkid] [int] NOT NULL IDENTITY(1, 1),  [projectid] [int] NOT NULL,  [createdby] [int] NOT NULL,  [creationdate] [datetime] NOT NULL,  [lastupdate] [datetime] NOT NULL,  [lastupdatedby] [int] NOT NULL,  [thumbnail] [varbinary] (max) NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [createdbyuniquemachineid] [uniqueidentifier] NOT NULL,  [isdeleted] [bit] NOT NULL CONSTRAINT [DF\_\_t\_quickli\_\_isdel\_\_123EB7A3] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_quicklink] ADD CONSTRAINT [PK\_\_t\_quickl\_\_­C76188A2EB00B22B] PRIMARY KEY CLUSTERED ([quicklinkid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_quicklink\_project] ON [dbo].[t\_quicklink] ([projectid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_quicklink] ADD CONSTRAINT [FK\_t\_quicklink\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_quicklink] ADD CONSTRAINT [FK\_t\_quicklink\_user] FOREIGN KEY ([createdby]) REFERENCES [dbo].[t\_user] ([userid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Quicklinks in a project', 'SCHEMA', N'dbo', 'TABLE', N't\_quicklink', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

Used By

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

[[dbo].[p\_quicklink\_delete]](#hSJmCOcnSP/0EQyXqFjk0TGGCrU=)

[[dbo].[p\_quicklink\_get\_by\_id]](#20MjjZnKch28Y9YLDxEDS3FGmTo=)

[[dbo].[p\_quicklink\_get\_by\_projectid]](#CjFA3Ut9sPwMnJs7sFx+eBX6Pf8=)

[[dbo].[p\_quicklink\_insert]](#7L+mI5DHWSS1FBBxmtHfmfPa4Mc=)

[[dbo].[p\_quicklink\_remove\_tile\_from\_quicklinks]](#kIJK7kTX7Jb4WDa8+PQQ1AGu1T8=)

[[dbo].[p\_quicklink\_update]](#dCF41uDCmFp5QJyNe7l1hO5erx8=)

|  |
| --- |
| [dbo].[t\_quicklink\_tile] |

MS\_­Description

Linking table between quicklinks and the tiles

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | quicklink\_tile\_id | int | 4 | NOT NULL | 1 - 1 |
|  | quicklinkid | int | 4 | NOT NULL |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_quickl\_\_55D4B02AD02744FD | quicklink\_tile\_id | True |
|  | IX\_t\_quicklink\_tile\_quicklinkid | quicklinkid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_quicklink\_tile\_quicklinks | quicklinkid->[[dbo].[t\_quicklink].[quicklinkid]](#8R5txRxbFoGoF2i7BOacoCDM29c=) |
| FK\_t\_quicklink\_tile\_tileid | tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_quicklink\_tile]  (  [quicklink\_tile\_id] [int] NOT NULL IDENTITY(1, 1),  [quicklinkid] [int] NOT NULL,  [tileid] [uniqueidentifier] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_quicklink\_tile] ADD CONSTRAINT [PK\_\_t\_quickl\_\_55D4B02AD02744FD] PRIMARY KEY CLUSTERED ([quicklink\_tile\_id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_quicklink\_tile\_quicklinkid] ON [dbo].[t\_quicklink\_tile] ([quicklinkid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_quicklink\_tile] ADD CONSTRAINT [FK\_t\_quicklink\_tile\_quicklinks] FOREIGN KEY ([quicklinkid]) REFERENCES [dbo].[t\_quicklink] ([quicklinkid])  GO  ALTER TABLE [dbo].[t\_quicklink\_tile] ADD CONSTRAINT [FK\_t\_quicklink\_tile\_tileid] FOREIGN KEY ([tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Linking table between quicklinks and the tiles', 'SCHEMA', N'dbo', 'TABLE', N't\_quicklink\_tile', NULL, NULL  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

Used By

[[dbo].[p\_quicklink\_get\_tiles\_for\_quicklink]](#Cucj13pAuEzWnCUMIObApOpoAn0=)

[[dbo].[p\_quicklink\_insert]](#7L+mI5DHWSS1FBBxmtHfmfPa4Mc=)

[[dbo].[p\_quicklink\_remove\_tile\_from\_quicklinks]](#kIJK7kTX7Jb4WDa8+PQQ1AGu1T8=)

[[dbo].[p\_quicklink\_update]](#dCF41uDCmFp5QJyNe7l1hO5erx8=)

|  |
| --- |
| [dbo].[t\_state\_machine\_status] |

MS\_­Description

Status of the state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 7 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | statemachinestatusid | int | 4 | NOT NULL |
|  | statemachinestatusdescription | nvarchar(50) | 100 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_state\_machine\_status | statemachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_state\_machine\_status]  (  [statemachinestatusid] [int] NOT NULL,  [statemachinestatusdescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_state\_machine\_status] ADD CONSTRAINT [PK\_t\_state\_machine\_status] PRIMARY KEY CLUSTERED ([statemachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Status of the state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_state\_machine\_status', NULL, NULL  GO |

Used By

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

|  |
| --- |
| [dbo].[t\_telemetrydata] |

MS\_­Description

Data for telemetry

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | Log­Entry­Id | int | 4 | NOT NULL | 1 - 1 |
|  | Creation­Date­Time | datetime | 8 | NOT NULL |  |
|  | Log­Entry | nvarchar(1024) | 2048 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_telemetrydata | Log­Entry­Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_telemetrydata]  (  [Log­Entry­Id] [int] NOT NULL IDENTITY(1, 1),  [Creation­Date­Time] [datetime] NOT NULL,  [Log­Entry] [nvarchar] (1024) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_telemetrydata] ADD CONSTRAINT [PK\_t\_telemetrydata] PRIMARY KEY CLUSTERED ([Log­Entry­Id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Data for telemetry', 'SCHEMA', N'dbo', 'TABLE', N't\_telemetrydata', NULL, NULL  GO |

Used By

[[dbo].[p\_telemetry\_insert]](#K+D3VDXeBNU758yas8f/5w0zA+0=)

|  |
| --- |
| [dbo].[t\_template\_translation] |

MS\_­Description

Translated entries for the templates

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | translationid | int | 4 | NOT NULL | 1 - 1 |
| (2) | projectid | int | 4 | NOT NULL |  |
| (2) | languageid | int | 4 | NOT NULL |  |
|  | description | nvarchar(250) | 500 | NULL allowed |  |
|  | extendeddescription | nvarchar(max) | max | NULL allowed |  |
|  | moreinfourl | nvarchar(250) | 500 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_template\_translation | translationid | True |
|  | UK\_t\_template\_translation | projectid, languageid | True |
|  | IX\_t\_template\_translation\_language | languageid |  |
|  | IX\_t\_template\_translation\_project | projectid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_template\_translation\_t\_language | languageid->[[dbo].[t\_language].[language­Id]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=) |
| FK\_t\_template\_translation\_t\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_template\_translation]  (  [translationid] [int] NOT NULL IDENTITY(1, 1),  [projectid] [int] NOT NULL,  [languageid] [int] NOT NULL,  [description] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [extendeddescription] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [moreinfourl] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_template\_translation] ADD CONSTRAINT [PK\_t\_template\_translation] PRIMARY KEY CLUSTERED ([translationid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_template\_translation\_language] ON [dbo].[t\_template\_translation] ([languageid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_template\_translation\_project] ON [dbo].[t\_template\_translation] ([projectid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_template\_translation] ADD CONSTRAINT [UK\_t\_template\_translation] UNIQUE NONCLUSTERED ([projectid], [languageid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_template\_translation] ADD CONSTRAINT [FK\_t\_template\_translation\_t\_language] FOREIGN KEY ([languageid]) REFERENCES [dbo].[t\_language] ([language­Id])  GO  ALTER TABLE [dbo].[t\_template\_translation] ADD CONSTRAINT [FK\_t\_template\_translation\_t\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Translated entries for the templates', 'SCHEMA', N'dbo', 'TABLE', N't\_template\_translation', NULL, NULL  GO |

Uses

[[dbo].[t\_language]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=)

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

Used By

[[dbo].[p\_project\_delete\_template]](#h6ZHMHXodjZQ7/r28dA/xoJe5Ic=)

[[dbo].[p\_project\_get\_templates]](#Kx+UI3EG+DPsEGR/IXReULy0pqM=)

[[dbo].[p\_project\_update\_template]](#puatJhq48ml0PZR3M/9OUxn1CLY=)

[[dbo].[p\_projecttypes\_convert\_to\_template]](#JwgwHPiObJe/EYScukAUlCcGImA=)

|  |
| --- |
| [dbo].[t\_tile\_relation] |

MS\_­Description

Relation between tiles

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | tilerelationid | int | 4 | NOT NULL | 1 - 1 |
|  | projectid | int | 4 | NOT NULL |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | related\_tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | tilerelationstyleid | int | 4 | NOT NULL |  |
|  | tilerelationanchoridorigin | int | 4 | NOT NULL |  |
|  | tilerelationanchoriddestination | int | 4 | NOT NULL |  |
|  | tilerelationstartendsymbolorigin | int | 4 | NOT NULL |  |
|  | tilerelationstartendsymboldestination | int | 4 | NOT NULL |  |
|  | tilerelationlineid | int | 4 | NOT NULL |  |
|  | color | nvarchar(7) | 14 | NOT NULL |  |
|  | thickness | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_1 | tilerelationid | True |
|  | IX\_­FK\_\_t\_tile\_re\_\_tiler\_\_5E74FADA | tilerelationstartendsymbolorigin |  |
|  | IX\_­FK\_\_t\_tile\_re\_\_tiler\_\_5F691F13 | tilerelationstartendsymboldestination |  |
|  | IX\_­FK\_t\_tile\_relation\_t\_project | projectid |  |
|  | IX\_­FK\_tilerelationanchor | tilerelationanchoridorigin |  |
|  | IX\_­FK\_tilerelationanchoriddestination | tilerelationanchoriddestination |  |
|  | IX\_­FK\_tilerelationstyle | tilerelationstyleid |  |
|  | IX\_t\_tile\_relation\_related\_tileid\_t\_tilestatus | related\_tileid |  |
|  | IX\_t\_tile\_relation\_tileid\_t\_tilestatus | tileid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_\_t\_tile\_re\_\_tiler\_\_5E74FADA | tilerelationstartendsymbolorigin->[[dbo].[t\_tile\_relation\_startend\_symbol].[tilerelationstartendsymbol]](#20FtsGWM3tQuQsgV56yVep7A5B8=) |
| FK\_\_t\_tile\_re\_\_tiler\_\_5F691F13 | tilerelationstartendsymboldestination->[[dbo].[t\_tile\_relation\_startend\_symbol].[tilerelationstartendsymbol]](#20FtsGWM3tQuQsgV56yVep7A5B8=) |
| FK\_t\_tile\_relation\_t\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| FK\_t\_tile\_relation\_t\_tile\_relation\_line | tilerelationlineid->[[dbo].[t\_tile\_relation\_line].[tilerelationlineid]](#+qMkPlre1TQOnP2h8pcGrG1QxDo=) |
| FK\_tilerelationanchor | tilerelationanchoridorigin->[[dbo].[t\_tile\_relation\_anchor].[tilerelationanchorid]](#FcUpACp1BApta+S3+HnXkpQMt4I=) |
| FK\_tilerelationanchoriddestination | tilerelationanchoriddestination->[[dbo].[t\_tile\_relation\_anchor].[tilerelationanchorid]](#FcUpACp1BApta+S3+HnXkpQMt4I=) |
| FK\_tilerelationstyle | tilerelationstyleid->[[dbo].[t\_tile\_relation\_style].[tilerelationstyleid]](#uaXVDUif5OUkaINcFrMH2xZwemo=) |
| t\_tile\_relation\_related\_tileid\_t\_tilestatus | related\_tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |
| t\_tile\_relation\_tileid\_t\_tilestatus | tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation]  (  [tilerelationid] [int] NOT NULL IDENTITY(1, 1),  [projectid] [int] NOT NULL,  [tileid] [uniqueidentifier] NOT NULL,  [related\_tileid] [uniqueidentifier] NOT NULL,  [tilerelationstyleid] [int] NOT NULL,  [tilerelationanchoridorigin] [int] NOT NULL,  [tilerelationanchoriddestination] [int] NOT NULL,  [tilerelationstartendsymbolorigin] [int] NOT NULL,  [tilerelationstartendsymboldestination] [int] NOT NULL,  [tilerelationlineid] [int] NOT NULL,  [color] [nvarchar] (7) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [thickness] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [PK\_t\_tile\_relation\_1] PRIMARY KEY CLUSTERED ([tilerelationid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_tile\_relation\_t\_project] ON [dbo].[t\_tile\_relation] ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tile\_relation\_related\_tileid\_t\_tilestatus] ON [dbo].[t\_tile\_relation] ([related\_tileid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tile\_relation\_tileid\_t\_tilestatus] ON [dbo].[t\_tile\_relation] ([tileid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_tilerelationanchoriddestination] ON [dbo].[t\_tile\_relation] ([tilerelationanchoriddestination]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_tilerelationanchor] ON [dbo].[t\_tile\_relation] ([tilerelationanchoridorigin]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_\_t\_tile\_re\_\_tiler\_\_5F691F13] ON [dbo].[t\_tile\_relation] ([tilerelationstartendsymboldestination]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_\_t\_tile\_re\_\_tiler\_\_5E74FADA] ON [dbo].[t\_tile\_relation] ([tilerelationstartendsymbolorigin]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_tilerelationstyle] ON [dbo].[t\_tile\_relation] ([tilerelationstyleid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_\_t\_tile\_re\_\_tiler\_\_5E74FADA] FOREIGN KEY ([tilerelationstartendsymbolorigin]) REFERENCES [dbo].[t\_tile\_relation\_startend\_symbol] ([tilerelationstartendsymbol])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_\_t\_tile\_re\_\_tiler\_\_5F691F13] FOREIGN KEY ([tilerelationstartendsymboldestination]) REFERENCES [dbo].[t\_tile\_relation\_startend\_symbol] ([tilerelationstartendsymbol])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_t\_tile\_relation\_t\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_t\_tile\_relation\_t\_tile\_relation\_line] FOREIGN KEY ([tilerelationlineid]) REFERENCES [dbo].[t\_tile\_relation\_line] ([tilerelationlineid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_tilerelationanchor] FOREIGN KEY ([tilerelationanchoridorigin]) REFERENCES [dbo].[t\_tile\_relation\_anchor] ([tilerelationanchorid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_tilerelationanchoriddestination] FOREIGN KEY ([tilerelationanchoriddestination]) REFERENCES [dbo].[t\_tile\_relation\_anchor] ([tilerelationanchorid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [FK\_tilerelationstyle] FOREIGN KEY ([tilerelationstyleid]) REFERENCES [dbo].[t\_tile\_relation\_style] ([tilerelationstyleid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [t\_tile\_relation\_related\_tileid\_t\_tilestatus] FOREIGN KEY ([related\_tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  ALTER TABLE [dbo].[t\_tile\_relation] ADD CONSTRAINT [t\_tile\_relation\_tileid\_t\_tilestatus] FOREIGN KEY ([tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Relation between tiles', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation', NULL, NULL  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_tile\_relation\_anchor]](#FcUpACp1BApta+S3+HnXkpQMt4I=)

[[dbo].[t\_tile\_relation\_line]](#+qMkPlre1TQOnP2h8pcGrG1QxDo=)

[[dbo].[t\_tile\_relation\_startend\_symbol]](#20FtsGWM3tQuQsgV56yVep7A5B8=)

[[dbo].[t\_tile\_relation\_style]](#uaXVDUif5OUkaINcFrMH2xZwemo=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

Used By

[[dbo].[p\_tile\_relation\_by\_participating\_tile]](#4nTl9Mym0JSY27XlS6qwx+oO8Ww=)

[[dbo].[p\_tile\_relation\_delete]](#qaxYLU/+tQD5a8bg+OzEjVFLvOA=)

[[dbo].[p\_tile\_relation\_delete\_all\_by\_tileid]](#fos8gM3ClGfCxOrzgmfdOF2zYyw=)

[[dbo].[p\_tile\_relation\_get\_by\_projectid]](#PjHeRYIUeUzVAfG8OgmofXHPZwY=)

[[dbo].[p\_tile\_relation\_get\_by\_projectid\_paged]](#4tHkV/GK4O7uKFVRAnsUzM5KpJA=)

[[dbo].[p\_tile\_relation\_insert]](#ATmtljARCqxJIPGy8/izryEv/y4=)

[[dbo].[p\_tile\_relation\_update]](#OPU9oKDp8YMJ9WjYBAoaPsfOSRY=)

[[dbo].[p\_tile\_relation\_with\_tile]](#TFrhkYoEzh9+cYTxXnhVxpekoBg=)

[[dbo].[p\_tilestatus\_delete]](#g8XrmtIP1eDM2T1aBQtWCyFzoHA=)

|  |
| --- |
| [dbo].[t\_tile\_relation\_anchor] |

MS\_­Description

The type of anchor for a relation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 5 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationanchorid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_anchor | tilerelationanchorid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_anchor]  (  [tilerelationanchorid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_anchor] ADD CONSTRAINT [PK\_t\_tile\_relation\_anchor] PRIMARY KEY CLUSTERED ([tilerelationanchorid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of anchor for a relation', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_anchor', NULL, NULL  GO |

Used By

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[t\_tile\_relation\_line] |

MS\_­Description

The type of line between relation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationlineid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_line | tilerelationlineid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_line]  (  [tilerelationlineid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_line] ADD CONSTRAINT [PK\_t\_tile\_relation\_line] PRIMARY KEY CLUSTERED ([tilerelationlineid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of line between relation', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_line', NULL, NULL  GO |

Used By

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[t\_tile\_relation\_line\_style] |

MS\_­Description

The line styles for a relation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationlinestyleid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_line\_style | tilerelationlinestyleid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_line\_style]  (  [tilerelationlinestyleid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_line\_style] ADD CONSTRAINT [PK\_t\_tile\_relation\_line\_style] PRIMARY KEY CLUSTERED ([tilerelationlinestyleid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The line styles for a relation', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_line\_style', NULL, NULL  GO |

|  |
| --- |
| [dbo].[t\_tile\_relation\_startend\_symbol] |

MS\_­Description

The symbol used for the relation line start

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 7 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationstartendsymbol | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_startend\_symbol | tilerelationstartendsymbol | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_startend\_symbol]  (  [tilerelationstartendsymbol] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_startend\_symbol] ADD CONSTRAINT [PK\_t\_tile\_relation\_startend\_symbol] PRIMARY KEY CLUSTERED ([tilerelationstartendsymbol]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The symbol used for the relation line start', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_startend\_symbol', NULL, NULL  GO |

Used By

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[t\_tile\_relation\_style] |

MS\_­Description

The style of the relation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationstyleid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_style | tilerelationstyleid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_style]  (  [tilerelationstyleid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_style] ADD CONSTRAINT [PK\_t\_tile\_relation\_style] PRIMARY KEY CLUSTERED ([tilerelationstyleid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The style of the relation ', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_style', NULL, NULL  GO |

Used By

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[t\_tile\_relation\_type] |

MS\_­Description

The type of the relation

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 4 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilerelationtypeid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_relation\_type | tilerelationtypeid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_relation\_type]  (  [tilerelationtypeid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_relation\_type] ADD CONSTRAINT [PK\_t\_tile\_relation\_type] PRIMARY KEY CLUSTERED ([tilerelationtypeid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of the relation', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_relation\_type', NULL, NULL  GO |

|  |
| --- |
| [dbo].[t\_tile\_type] |

MS\_­Description

The type of the tile

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 17 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tiletypeid | int | 4 | NOT NULL |
|  | tiledescription | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_tile\_type | tiletypeid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_type]  (  [tiletypeid] [int] NOT NULL,  [tiledescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_type] ADD CONSTRAINT [PK\_tile\_type] PRIMARY KEY CLUSTERED ([tiletypeid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of the tile', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_type', NULL, NULL  GO |

Used By

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[t\_tile\_voting] |

MS\_­Description

Voting for the tile

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | userid | int | 4 | NOT NULL |  |
|  | projectid | int | 4 | NOT NULL |  |
|  | rate | int | 4 | NOT NULL | ((0)) |
|  | thumb 0.1 thumb down, 0.0 no thub, 1.0 thumb up | decimal(18,9) | 9 | NOT NULL | ((0.0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tile\_thumb | tileid, userid | True |
|  | IX\_­FK\_­Projectid | projectid |  |
|  | IX\_­FK\_\_t\_tile\_th\_\_tilei\_\_75586032 | tileid |  |
|  | IX\_­FK\_\_t\_tile\_th\_\_useri\_\_764C846B | userid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_\_t\_tile\_th\_\_tilei\_\_75586032 | tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |
| FK\_\_t\_tile\_th\_\_useri\_\_764C846B | userid->[[dbo].[t\_user].[userid]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=) |
| FK\_­Projectid | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tile\_voting]  (  [tileid] [uniqueidentifier] NOT NULL,  [userid] [int] NOT NULL,  [projectid] [int] NOT NULL,  [rate] [int] NOT NULL CONSTRAINT [DF\_t\_tile\_thumb\_rate] DEFAULT ((0)),  [thumb] [decimal] (18, 9) NOT NULL CONSTRAINT [DF\_t\_tile\_voting\_thumb] DEFAULT ((0.0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_voting] ADD CONSTRAINT [PK\_t\_tile\_thumb] PRIMARY KEY CLUSTERED ([tileid], [userid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_­Projectid] ON [dbo].[t\_tile\_voting] ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_\_t\_tile\_th\_\_tilei\_\_75586032] ON [dbo].[t\_tile\_voting] ([tileid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_\_t\_tile\_th\_\_useri\_\_764C846B] ON [dbo].[t\_tile\_voting] ([userid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tile\_voting] ADD CONSTRAINT [FK\_\_t\_tile\_th\_\_tilei\_\_75586032] FOREIGN KEY ([tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  ALTER TABLE [dbo].[t\_tile\_voting] ADD CONSTRAINT [FK\_\_t\_tile\_th\_\_useri\_\_764C846B] FOREIGN KEY ([userid]) REFERENCES [dbo].[t\_user] ([userid])  GO  ALTER TABLE [dbo].[t\_tile\_voting] ADD CONSTRAINT [FK\_­Projectid] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Voting for the tile', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_voting', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'0.1 thumb down, 0.0 no thub, 1.0 thumb up', 'SCHEMA', N'dbo', 'TABLE', N't\_tile\_voting', 'COLUMN', N'thumb'  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

Used By

[[dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id]](#pM1zrS3UXx8O9xap0QOnGAXEIP0=)

[[dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id\_paged]](#uORfngnLeXLGViltwAI1sKjEVXs=)

[[dbo].[p\_tile\_voting\_update]](#Oy2PprjsLacUoOpGiBn/tfpZFfQ=)

[[dbo].[p\_tilestatus\_delete]](#g8XrmtIP1eDM2T1aBQtWCyFzoHA=)

[[dbo].[p\_tilestatus\_update\_blobstatusid]](#rKXiHuqjFMw5RE7XRuNRz7uwpBs=)

|  |
| --- |
| [dbo].[t\_tilecopy\_copiedrelation] |

MS\_­Description

All copied relations for a copy command

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | token | uniqueidentifier | 16 | NOT NULL |  |
|  | copyfinishedtimestamp | datetime | 8 | NOT NULL |  |
|  | usernamecreator | nvarchar(50) | 100 | NOT NULL |  |
|  | sourceprojectid | int | 4 | NOT NULL |  |
|  | targetprojectid | int | 4 | NULL allowed |  |
|  | tilerelationid | int | 4 | NOT NULL |  |
|  | projectid | int | 4 | NOT NULL |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | related\_tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | tilerelationstyleid | int | 4 | NOT NULL |  |
|  | tilerelationanchoridorigin | int | 4 | NOT NULL |  |
|  | tilerelationanchoriddestination | int | 4 | NOT NULL |  |
|  | tilerelationstartendsymbolorigin | int | 4 | NOT NULL |  |
|  | tilerelationstartendsymboldestination | int | 4 | NOT NULL |  |
|  | tilerelationlineid | int | 4 | NOT NULL |  |
|  | color | nvarchar(7) | 14 | NOT NULL |  |
|  | thickness | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tilecopy\_copiedrelatoin | id | True |
|  | IX\_tilecopy\_copiedrelation\_token | token |  |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilecopy\_copiedrelation]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [token] [uniqueidentifier] NOT NULL,  [copyfinishedtimestamp] [datetime] NOT NULL,  [usernamecreator] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [sourceprojectid] [int] NOT NULL,  [targetprojectid] [int] NULL,  [tilerelationid] [int] NOT NULL,  [projectid] [int] NOT NULL,  [tileid] [uniqueidentifier] NOT NULL,  [related\_tileid] [uniqueidentifier] NOT NULL,  [tilerelationstyleid] [int] NOT NULL,  [tilerelationanchoridorigin] [int] NOT NULL,  [tilerelationanchoriddestination] [int] NOT NULL,  [tilerelationstartendsymbolorigin] [int] NOT NULL,  [tilerelationstartendsymboldestination] [int] NOT NULL,  [tilerelationlineid] [int] NOT NULL,  [color] [nvarchar] (7) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [thickness] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilecopy\_copiedrelation] ADD CONSTRAINT [PK\_t\_tilecopy\_copiedrelatoin] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_tilecopy\_copiedrelation\_token] ON [dbo].[t\_tilecopy\_copiedrelation] ([token]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All copied relations for a copy command', 'SCHEMA', N'dbo', 'TABLE', N't\_tilecopy\_copiedrelation', NULL, NULL  GO |

Used By

[[dbo].[p\_tilecopy\_cleanup]](#B8DCD5XXKduW2LGzKEpheITcMD0=)

[[dbo].[p\_tilecopy\_getrelations\_by\_token]](#tAWK7qE6vmmZvtMeEfSUlTEvQgU=)

[[dbo].[p\_tilecopy\_insertrelation]](#rZoqGf5JuiqmeFlO6/jHRu4hBwg=)

|  |
| --- |
| [dbo].[t\_tilecopy\_copiedtile] |

MS\_­Description

All copied tiles for a copy command

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | token | uniqueidentifier | 16 | NOT NULL |  |
|  | copyfinishedtimestamp | datetime | 8 | NOT NULL |  |
|  | usernamecreator | nvarchar(50) | 100 | NOT NULL |  |
|  | sourceprojectid | int | 4 | NOT NULL |  |
|  | targetprojectid | int | 4 | NULL allowed |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
|  | projectid | int | 4 | NOT NULL |  |
|  | parentid | uniqueidentifier | 16 | NULL allowed |  |
|  | tiletype | int | 4 | NOT NULL |  |
|  | tilecontent | nvarchar(max) | max | NULL allowed |  |
|  | positionx | float | 8 | NOT NULL |  |
|  | positiony | float | 8 | NOT NULL |  |
|  | height | float | 8 | NOT NULL |  |
|  | width | float | 8 | NOT NULL |  |
|  | zindex | int | 4 | NOT NULL |  |
|  | scalex | float | 8 | NOT NULL |  |
|  | scaley | float | 8 | NOT NULL |  |
|  | rotation | float | 8 | NOT NULL |  |
|  | totalthumbup | int | 4 | NOT NULL |  |
|  | totlathumbdown | int | 4 | NOT NULL |  |
|  | overallrating | decimal(2,1) | 5 | NOT NULL |  |
|  | totalrates | int | 4 | NOT NULL |  |
|  | ispinned | bit | 1 | NOT NULL |  |
|  | backgroundcolor | nvarchar(7) | 14 | NULL allowed |  |
|  | originalfilename | nvarchar(255) | 510 | NULL allowed |  |
|  | lockeduser | nvarchar(50) | 100 | NULL allowed |  |
|  | lockedby | nvarchar(50) | 100 | NULL allowed |  |
|  | lockeddate | datetime | 8 | NULL allowed |  |
|  | lastupdate | datetime | 8 | NOT NULL |  |
|  | blobstatusid | int | 4 | NOT NULL |  |
|  | old­Id | uniqueidentifier | 16 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL |  |
|  | statemachinestatusid | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tilecopy\_copiedtile | id | True |
|  | IX\_tilecopy\_copiedtile\_token | token |  |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilecopy\_copiedtile]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [token] [uniqueidentifier] NOT NULL,  [copyfinishedtimestamp] [datetime] NOT NULL,  [usernamecreator] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [sourceprojectid] [int] NOT NULL,  [targetprojectid] [int] NULL,  [tileid] [uniqueidentifier] NOT NULL,  [projectid] [int] NOT NULL,  [parentid] [uniqueidentifier] NULL,  [tiletype] [int] NOT NULL,  [tilecontent] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [positionx] [float] NOT NULL,  [positiony] [float] NOT NULL,  [height] [float] NOT NULL,  [width] [float] NOT NULL,  [zindex] [int] NOT NULL,  [scalex] [float] NOT NULL,  [scaley] [float] NOT NULL,  [rotation] [float] NOT NULL,  [totalthumbup] [int] NOT NULL,  [totlathumbdown] [int] NOT NULL,  [overallrating] [decimal] (2, 1) NOT NULL,  [totalrates] [int] NOT NULL,  [ispinned] [bit] NOT NULL,  [backgroundcolor] [nvarchar] (7) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [originalfilename] [nvarchar] (255) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockeduser] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockedby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockeddate] [datetime] NULL,  [lastupdate] [datetime] NOT NULL,  [blobstatusid] [int] NOT NULL,  [old­Id] [uniqueidentifier] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL,  [statemachinestatusid] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilecopy\_copiedtile] ADD CONSTRAINT [PK\_t\_tilecopy\_copiedtile] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_tilecopy\_copiedtile\_token] ON [dbo].[t\_tilecopy\_copiedtile] ([token]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All copied tiles for a copy command', 'SCHEMA', N'dbo', 'TABLE', N't\_tilecopy\_copiedtile', NULL, NULL  GO |

Used By

[[dbo].[p\_tilecopy\_cleanup]](#B8DCD5XXKduW2LGzKEpheITcMD0=)

[[dbo].[p\_tilecopy\_getcopiedtiles\_by\_token]](#GApIlX6Y9pIlbETFFMYFvJHAKTU=)

[[dbo].[p\_tilecopy\_insertcopiedtile]](#HDv12qJtmkCLjWHumxpbZOTBz8g=)

|  |
| --- |
| [dbo].[t\_tilecopy\_sourcetile] |

MS\_­Description

All source tiles used in a copy command

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | token | uniqueidentifier | 16 | NOT NULL |  |
|  | copyfinishedtimestamp | datetime | 8 | NOT NULL |  |
|  | usernamecreator | nvarchar(50) | 100 | NOT NULL |  |
|  | sourceprojectid | int | 4 | NOT NULL |  |
|  | targetprojectid | int | 4 | NULL allowed |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tilecopy\_sourcetile | id | True |
|  | IX\_tilecopy\_sourcetile\_token | token |  |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilecopy\_sourcetile]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [token] [uniqueidentifier] NOT NULL,  [copyfinishedtimestamp] [datetime] NOT NULL,  [usernamecreator] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [sourceprojectid] [int] NOT NULL,  [targetprojectid] [int] NULL,  [tileid] [uniqueidentifier] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilecopy\_sourcetile] ADD CONSTRAINT [PK\_t\_tilecopy\_sourcetile] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_tilecopy\_sourcetile\_token] ON [dbo].[t\_tilecopy\_sourcetile] ([token]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All source tiles used in a copy command', 'SCHEMA', N'dbo', 'TABLE', N't\_tilecopy\_sourcetile', NULL, NULL  GO |

Used By

[[dbo].[p\_tilecopy\_cleanup]](#B8DCD5XXKduW2LGzKEpheITcMD0=)

[[dbo].[p\_tilecopy\_getsourcetiles\_by\_token]](#cmT7FnIv1v7sCmmIynRK098uy4A=)

[[dbo].[p\_tilecopy\_insertsourcetile]](#r9P0d35uYsFf6SpqEej636wKAMM=)

|  |
| --- |
| [dbo].[t\_tilestatus] |

MS\_­Description

The tiles

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |
| (2) | projectid | int | 4 | NOT NULL |  |
|  | parentid | uniqueidentifier | 16 | NULL allowed |  |
|  | tiletype | int | 4 | NOT NULL | ((0)) |
|  | tilecontent | nvarchar(max) | max | NULL allowed |  |
|  | positionx | float | 8 | NOT NULL | ((0)) |
|  | positiony | float | 8 | NOT NULL | ((0)) |
|  | height | float | 8 | NOT NULL | ((0)) |
|  | width | float | 8 | NOT NULL | ((0)) |
|  | zindex | int | 4 | NOT NULL | ((0)) |
|  | scalex | float | 8 | NOT NULL | ((1)) |
|  | scaley | float | 8 | NOT NULL | ((1)) |
|  | rotation | float | 8 | NOT NULL | ((0)) |
|  | totalthumbup | int | 4 | NOT NULL | ((0)) |
|  | totlathumbdown | int | 4 | NOT NULL | ((0)) |
|  | overallrating | decimal(2,1) | 5 | NOT NULL | ((0.0)) |
|  | totalrates | int | 4 | NOT NULL | ((0)) |
|  | ispinned | bit | 1 | NOT NULL | ((0)) |
|  | backgroundcolor | nvarchar(7) | 14 | NULL allowed |  |
|  | lockeduser | nvarchar(50) | 100 | NULL allowed |  |
|  | lockedby lockedby is the uniquemachineid on client | nvarchar(50) | 100 | NULL allowed |  |
|  | lockeddate | datetime | 8 | NULL allowed |  |
|  | lastupdate | datetime | 8 | NOT NULL |  |
| (3) | blobstatusid | int | 4 | NOT NULL | ((1)) |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
| (2) | statemachinestatusid | int | 4 | NOT NULL | ((0)) |
|  | originalfilename | nvarchar(255) | 510 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tilestatus | tileid | True |
|  | IX\_­FK\_t\_tilestatus\_t\_blob\_status | blobstatusid |  |
|  | IX\_­FK\_t\_tilestatus\_t\_project | projectid |  |
|  | IX\_­FK\_t\_tilestatus\_t\_tile\_type | tiletype |  |
|  | IX\_t\_tilestatus\_blobstatusid | blobstatusid |  |
|  | IX\_t\_tilestatus\_bookedby | bookedby |  |
|  | IX\_t\_tilestatus\_lookedby | lockedby |  |
|  | IX\_t\_tilestatus\_parentid | parentid |  |
|  | IX\_t\_tilestatus\_projectid\_blobstatusid | projectid, blobstatusid |  |
|  | IX\_t\_tilestatus\_statemachinestatusid | statemachinestatusid |  |
|  | IX\_t\_tilestatus\_t\_state\_machine\_status | statemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_tilestatus\_t\_blob\_status | blobstatusid->[[dbo].[t\_blob\_status].[blobstatusid]](#Wh7xQn2/X4EcCdSoo4kUljq/Gp8=) |
| FK\_t\_tilestatus\_t\_project | projectid->[[dbo].[t\_project].[projectid]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=) |
| FK\_t\_tilestatus\_t\_tile\_type | tiletype->[[dbo].[t\_tile\_type].[tiletypeid]](#8YaY5k86JZ8/My8nxSV+Md7QE6s=) |
| t\_tilestatus\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilestatus]  (  [tileid] [uniqueidentifier] NOT NULL,  [projectid] [int] NOT NULL,  [parentid] [uniqueidentifier] NULL,  [tiletype] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_tiletype] DEFAULT ((0)),  [tilecontent] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [positionx] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_centerx] DEFAULT ((0)),  [positiony] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_centery] DEFAULT ((0)),  [height] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_height] DEFAULT ((0)),  [width] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_width] DEFAULT ((0)),  [zindex] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_zindex] DEFAULT ((0)),  [scalex] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_scale] DEFAULT ((1)),  [scaley] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_scaley] DEFAULT ((1)),  [rotation] [float] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_rotation] DEFAULT ((0)),  [totalthumbup] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_totalthumbup] DEFAULT ((0)),  [totlathumbdown] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_totlathumbdown] DEFAULT ((0)),  [overallrating] [decimal] (2, 1) NOT NULL CONSTRAINT [DF\_t\_tilestatus\_overalrating] DEFAULT ((0.0)),  [totalrates] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_totalrates] DEFAULT ((0)),  [ispinned] [bit] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_ispinned] DEFAULT ((0)),  [backgroundcolor] [nvarchar] (7) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockeduser] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockedby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lockeddate] [datetime] NULL,  [lastupdate] [datetime] NOT NULL,  [blobstatusid] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_blobstatusid] DEFAULT ((1)),  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_statemachinestatusid] DEFAULT ((0)),  [originalfilename] [nvarchar] (255) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus] ADD CONSTRAINT [PK\_t\_tilestatus] PRIMARY KEY CLUSTERED ([tileid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_tilestatus\_t\_blob\_status] ON [dbo].[t\_tilestatus] ([blobstatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_blobstatusid] ON [dbo].[t\_tilestatus] ([blobstatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_bookedby] ON [dbo].[t\_tilestatus] ([bookedby]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_lookedby] ON [dbo].[t\_tilestatus] ([lockedby]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_parentid] ON [dbo].[t\_tilestatus] ([parentid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_tilestatus\_t\_project] ON [dbo].[t\_tilestatus] ([projectid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_projectid\_blobstatusid] ON [dbo].[t\_tilestatus] ([projectid], [blobstatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_statemachinestatusid] ON [dbo].[t\_tilestatus] ([statemachinestatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_t\_state\_machine\_status] ON [dbo].[t\_tilestatus] ([statemachinestatusid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_tilestatus\_t\_tile\_type] ON [dbo].[t\_tilestatus] ([tiletype]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus] ADD CONSTRAINT [FK\_t\_tilestatus\_t\_blob\_status] FOREIGN KEY ([blobstatusid]) REFERENCES [dbo].[t\_blob\_status] ([blobstatusid])  GO  ALTER TABLE [dbo].[t\_tilestatus] ADD CONSTRAINT [FK\_t\_tilestatus\_t\_project] FOREIGN KEY ([projectid]) REFERENCES [dbo].[t\_project] ([projectid])  GO  ALTER TABLE [dbo].[t\_tilestatus] ADD CONSTRAINT [FK\_t\_tilestatus\_t\_tile\_type] FOREIGN KEY ([tiletype]) REFERENCES [dbo].[t\_tile\_type] ([tiletypeid])  GO  ALTER TABLE [dbo].[t\_tilestatus] ADD CONSTRAINT [t\_tilestatus\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The tiles', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus', NULL, NULL  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'lockedby is the uniquemachineid on client', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus', 'COLUMN', N'lockedby'  GO |

Uses

[[dbo].[t\_blob\_status]](#Wh7xQn2/X4EcCdSoo4kUljq/Gp8=)

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

[[dbo].[t\_tile\_type]](#8YaY5k86JZ8/My8nxSV+Md7QE6s=)

Used By

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

[[dbo].[t\_tilestatus\_stroke]](#eNq8iC4zx70GGM6EXbHSL++uzlU=)

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

[[dbo].[p\_online\_user\_delete]](#TBplcYBoKypjbhjavP5K5dOlL1c=)

[[dbo].[p\_storage\_eraser\_unbook]](#DJ35DDYZvLl9Ltne6CU+Guru1bk=)

[[dbo].[p\_tile\_relation\_by\_participating\_tile]](#4nTl9Mym0JSY27XlS6qwx+oO8Ww=)

[[dbo].[p\_tilestatus\_delete]](#g8XrmtIP1eDM2T1aBQtWCyFzoHA=)

[[dbo].[p\_tilestatus\_get\_by\_id]](#XViQ2TSrI9Jh6ab5s0my1mZmNj4=)

[[dbo].[p\_tilestatus\_get\_by\_parenttile]](#omKDRG2Umem5Xzmwq0PzSReapl0=)

[[dbo].[p\_tilestatus\_get\_by\_projectid]](#vjVE9Go4aDkBtZQTZoJWex+rCTw=)

[[dbo].[p\_tilestatus\_get\_by\_projectid\_paged]](#ZP9t21UM3jQkhPAjnryIGcmsaG0=)

[[dbo].[p\_tilestatus\_get\_by\_projectid\_with\_thumbnails]](#MMa4x8Vn5ChNWEY98YLQDM4wcRk=)

[[dbo].[p\_tilestatus\_get\_max\_zindex]](#8+MBm5Afux+OKPTv/cx5nXn2hyY=)

[[dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile]](#onMohemUZN4w8Ujofa/vIkPMLus=)

[[dbo].[p\_tilestatus\_get\_toplevel\_by\_projectid]](#lBkwsZ7bE97Z9t0qayGlQ38X99w=)

[[dbo].[p\_tilestatus\_get\_upgradable\_for\_project]](#eyljmpfDG1RdeUTtqr/BN0GN86U=)

[[dbo].[p\_tilestatus\_gettree\_by\_id]](#6VlWIweS4UGBOSOX46QlUJpkZ+Y=)

[[dbo].[p\_tilestatus\_insert]](#iMwd2Wua9ye9l0uBpCfK/FdQwyU=)

[[dbo].[p\_tilestatus\_unlock\_by\_ids]](#XXk93kY385SJHTBP+5O3Rfflchc=)

[[dbo].[p\_tilestatus\_unlock\_by\_lockedby]](#FvFyeAP1oR9rBySsO9wdwnY33zk=)

[[dbo].[p\_tilestatus\_unlock\_by\_project\_lockedby]](#4zv/71xy0wAzvROZMa4EIKlifsE=)

[[dbo].[p\_tilestatus\_unlock\_forgot]](#8SPblYSof1I8LItXwQ2VeLi22IY=)

[[dbo].[p\_tilestatus\_update]](#rEGjz6syz+nr2U/s/JA4/83VTuE=)

[[dbo].[p\_tilestatus\_update\_blobstatusid]](#rKXiHuqjFMw5RE7XRuNRz7uwpBs=)

[[dbo].[p\_tilestatus\_update\_containertype]](#EMnhgHE7sLgdHIlNA7PgnQ+Mmac=)

[[dbo].[p\_tilestatus\_update\_content]](#ovOnybcjSuIEeGh3AadnsW3fAak=)

[[dbo].[p\_tilestatus\_update\_grouped]](#jkNZ28XCQaKwniHN89cNfABNs90=)

[[dbo].[p\_tilestatus\_update\_lock]](#aczDQAYDZF2HlDSzivZSrAuFtOc=)

[[dbo].[p\_tilestatus\_update\_pinning]](#He9/XXIp9nHxusqmbIYpJZSVt+Y=)

[[dbo].[p\_tilestatus\_update\_position]](#rVRoekoodnhEE3EEZ9teUKHN6pM=)

[[dbo].[p\_tilestatus\_update\_voting]](#06H4YaAqEgHnlTMG8Tv9dNbrIj8=)

[[dbo].[p\_tilestatus\_update\_zindex]](#Amby7pg+gmRtOz9BmB4nTr5EV6I=)

[[dbo].[p\_tilestatus\_upgrade\_file]](#w3C0inKZgUwc4TuTsp6vtlZZkKo=)

[[dbo].[p\_update\_blobstatus\_for\_thumbnails]](#W9z9KJFncZneNc33VPLiUIevkYs=)

|  |
| --- |
| [dbo].[t\_tilestatus\_copy\_booking] |

MS\_­Description

Copy of a tile state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | Id | int | 4 | NOT NULL | 1 - 1 |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | username | nvarchar(200) | 400 | NOT NULL |  |  |
|  | sourceprojectid | int | 4 | NOT NULL |  |  |
|  | destinationprojectid | int | 4 | NOT NULL |  |  |
|  | tilesandpositions | nvarchar(max) | max | NULL allowed |  |  |
|  | tileoffsetx | float | 8 | NULL allowed |  |  |
|  | tileoffsety | float | 8 | NULL allowed |  |  |
|  | clientconnectionid | nvarchar(50) | 100 | NULL allowed |  |  |
|  | appver | nvarchar(50) | 100 | NOT NULL |  |  |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_tilest\_\_3214EC071AFA3CB9 | Id | True |
|  | IX\_t\_tilestatus\_copy\_booking\_statemachinestatusid | statemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| t\_tilestatus\_copy\_booking\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilestatus\_copy\_booking]  (  [Id] [int] NOT NULL IDENTITY(1, 1),  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_copy\_booking\_creationdate] DEFAULT (getutcdate()),  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [sourceprojectid] [int] NOT NULL,  [destinationprojectid] [int] NOT NULL,  [tilesandpositions] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [tileoffsetx] [float] NULL,  [tileoffsety] [float] NULL,  [clientconnectionid] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [appver] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [lastupdate] [datetime] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_copy\_booking\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_tilestatus\_copy\_booking\_statemachinestatusid] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_copy\_booking] ADD CONSTRAINT [PK\_\_t\_tilest\_\_3214EC071AFA3CB9] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tilestatus\_copy\_booking\_statemachinestatusid] ON [dbo].[t\_tilestatus\_copy\_booking] ([statemachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_copy\_booking] ADD CONSTRAINT [t\_tilestatus\_copy\_booking\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Copy of a tile state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus\_copy\_booking', NULL, NULL  GO |

Uses

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

Used By

[[dbo].[p\_tilestatus\_copy\_book]](#kzIlhS2mJFiJma1P3FyuIngtmj4=)

[[dbo].[p\_tilestatus\_copy\_fallback]](#FAozwoeeZq/SBEnuUGe8CMluFkI=)

[[dbo].[p\_tilestatus\_copy\_history]](#jaRkA76y/rG7Lw8O/YMA4asdxIs=)

[[dbo].[p\_tilestatus\_copy\_insert]](#dhfHL2WtimjmNW26GV2CNafaNn0=)

[[dbo].[p\_tilestatus\_copy\_unbook]](#4S87Z+DjBYEuUD8UEzaRkx7S0CU=)

|  |
| --- |
| [dbo].[t\_tilestatus\_copy\_booking\_history] |

MS\_­Description

Copy of a tile state machine history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:02 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | Id | int | 4 | NOT NULL |
|  | creationdate | datetime | 8 | NOT NULL |
|  | username | nvarchar(200) | 400 | NOT NULL |
|  | sourceprojectid | int | 4 | NOT NULL |
|  | destinationprojectid | int | 4 | NOT NULL |
|  | tilesandpositions | nvarchar(max) | max | NULL allowed |
|  | tileoffsetx | float | 8 | NULL allowed |
|  | tileoffsety | float | 8 | NULL allowed |
|  | clientconnectionid | nvarchar(50) | 100 | NULL allowed |
|  | appver | nvarchar(50) | 100 | NOT NULL |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |
|  | lastupdate | datetime | 8 | NULL allowed |
|  | bookingdate | datetime | 8 | NULL allowed |
|  | retry | int | 4 | NOT NULL |
|  | statemachinestatusid | int | 4 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_tilest\_\_3214EC077B0C4741 | Id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilestatus\_copy\_booking\_history]  (  [Id] [int] NOT NULL,  [creationdate] [datetime] NOT NULL,  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [sourceprojectid] [int] NOT NULL,  [destinationprojectid] [int] NOT NULL,  [tilesandpositions] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [tileoffsetx] [float] NULL,  [tileoffsety] [float] NULL,  [clientconnectionid] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [appver] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [lastupdate] [datetime] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL,  [statemachinestatusid] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_copy\_booking\_history] ADD CONSTRAINT [PK\_\_t\_tilest\_\_3214EC077B0C4741] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Copy of a tile state machine history', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus\_copy\_booking\_history', NULL, NULL  GO |

Used By

[[dbo].[p\_tilestatus\_copy\_history]](#jaRkA76y/rG7Lw8O/YMA4asdxIs=)

[[dbo].[p\_tilestatus\_copy\_history\_search]](#h05wqjsaRO7IKn4gZbxDCQJeE5A=)

|  |
| --- |
| [dbo].[t\_tilestatus\_stroke] |

MS\_­Description

The stroke for inking in a tile

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | tilestatusstrokeid | uniqueidentifier | 16 | NOT NULL |
|  | tileid | uniqueidentifier | 16 | NULL allowed |
|  | stroke | varbinary(max) | max | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_tilestatus\_stroke | tilestatusstrokeid | True |
|  | IX\_­FK\_t\_tilestatus\_stroke\_tileid | tileid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_tilestatus\_stroke\_tileid | tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilestatus\_stroke]  (  [tilestatusstrokeid] [uniqueidentifier] NOT NULL,  [tileid] [uniqueidentifier] NULL,  [stroke] [varbinary] (max) NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_stroke] ADD CONSTRAINT [PK\_t\_tilestatus\_stroke] PRIMARY KEY CLUSTERED ([tilestatusstrokeid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_­FK\_t\_tilestatus\_stroke\_tileid] ON [dbo].[t\_tilestatus\_stroke] ([tileid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_stroke] ADD CONSTRAINT [FK\_t\_tilestatus\_stroke\_tileid] FOREIGN KEY ([tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The stroke for inking in a tile', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus\_stroke', NULL, NULL  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

Used By

[[dbo].[p\_tilestatus\_stroke\_by\_tileid]](#pF4LLqFVJBsIvqjSOGzm05jMQXk=)

[[dbo].[p\_tilestatus\_stroke\_insert]](#plwNMHS8hQljZcchH6tJvNDattQ=)

[[dbo].[p\_tilestatus\_stroke\_update]](#jTMbOmyXVLghL03C2dibHdt3+OY=)

|  |
| --- |
| [dbo].[t\_tilestatus\_upgrade\_booking] |

MS\_­Description

Tile update mechanism state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | Id | int | 4 | NOT NULL | 1 - 1 |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | username | nvarchar(200) | 400 | NOT NULL |  |  |
|  | projectupgradeid | int | 4 | NOT NULL |  |  |
|  | tileid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | uniquedeviceid | uniqueidentifier | 16 | NOT NULL |  |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | statemachinestatusid | int | 4 | NOT NULL |  | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_\_t\_tilest\_\_3214EC07B7EE8479 | Id | True |
|  | IX\_t\_tile\_upgrade\_booking\_statemachinestatusid | statemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| t\_tile\_upgrade\_booking\_t\_project\_upgrade\_booking | projectupgradeid->[[dbo].[t\_project\_upgrade\_booking].[Id]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=) |
| t\_tile\_upgrade\_booking\_t\_state\_machine\_status | statemachinestatusid->[[dbo].[t\_state\_machine\_status].[statemachinestatusid]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=) |
| t\_tile\_upgrade\_booking\_t\_tilestatus | tileid->[[dbo].[t\_tilestatus].[tileid]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_tilestatus\_upgrade\_booking]  (  [Id] [int] NOT NULL IDENTITY(1, 1),  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_tile\_upgrade\_booking\_creationdate] DEFAULT (getutcdate()),  [username] [nvarchar] (200) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [projectupgradeid] [int] NOT NULL,  [tileid] [uniqueidentifier] NOT NULL,  [uniquedeviceid] [uniqueidentifier] NOT NULL,  [bookedby] [uniqueidentifier] NULL,  [lastupdate] [datetime] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_tile\_upgrade\_booking\_retry] DEFAULT ((0)),  [statemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_tile\_upgrade\_booking\_statemachinestatusid] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_upgrade\_booking] ADD CONSTRAINT [PK\_\_t\_tilest\_\_3214EC07B7EE8479] PRIMARY KEY CLUSTERED ([Id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_tile\_upgrade\_booking\_statemachinestatusid] ON [dbo].[t\_tilestatus\_upgrade\_booking] ([statemachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_tilestatus\_upgrade\_booking] ADD CONSTRAINT [t\_tile\_upgrade\_booking\_t\_project\_upgrade\_booking] FOREIGN KEY ([projectupgradeid]) REFERENCES [dbo].[t\_project\_upgrade\_booking] ([Id])  GO  ALTER TABLE [dbo].[t\_tilestatus\_upgrade\_booking] ADD CONSTRAINT [t\_tile\_upgrade\_booking\_t\_state\_machine\_status] FOREIGN KEY ([statemachinestatusid]) REFERENCES [dbo].[t\_state\_machine\_status] ([statemachinestatusid])  GO  ALTER TABLE [dbo].[t\_tilestatus\_upgrade\_booking] ADD CONSTRAINT [t\_tile\_upgrade\_booking\_t\_tilestatus] FOREIGN KEY ([tileid]) REFERENCES [dbo].[t\_tilestatus] ([tileid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Tile update mechanism state machine', 'SCHEMA', N'dbo', 'TABLE', N't\_tilestatus\_upgrade\_booking', NULL, NULL  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[t\_state\_machine\_status]](#QyjxYZf6qxyRTR9G7/p1mIdYvDg=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

Used By

[[dbo].[p\_tilestatus\_upgrade\_book]](#ld1zkb/fqpBa8wnM58UqebM1F4Q=)

[[dbo].[p\_tilestatus\_upgrade\_insert]](#cutcOfLrdt2GiPnsGtG+EKSAy/A=)

[[dbo].[p\_tilestatus\_upgrade\_unbook]](#ULCY9m7aUkthzdBb6jkSOHYQKPQ=)

|  |
| --- |
| [dbo].[t\_traces] |

MS\_­Description

Debug / Trace information

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Heap | True |
| Row Count (~) | 0 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
| ID | int | 4 | NOT NULL | 1 - 1 |  |
| Message | nvarchar(max) | max | NULL allowed |  |  |
| Severity­Level | int | 4 | NULL allowed |  |  |
| Last­Update | datetime | 8 | NULL allowed |  | (getutcdate()) |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_traces]  (  [ID] [int] NOT NULL IDENTITY(1, 1),  [Message] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [Severity­Level] [int] NULL,  [Last­Update] [datetime] NULL CONSTRAINT [DF\_t\_traces\_­Last­Update] DEFAULT (getutcdate())  ) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Debug / Trace information', 'SCHEMA', N'dbo', 'TABLE', N't\_traces', NULL, NULL  GO |

Used By

[[dbo].[p\_insert\_trace]](#Xy+FZ4VbEdhnKPT68XTI1YCsOds=)

|  |
| --- |
| [dbo].[t\_user] |

MS\_­Description

The user as used in tiles and projects

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 6 |
| Created | 06:53:02 vrijdag 14 mei 2021 |
| Last Modified | 06:53:03 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | userid | int | 4 | NOT NULL | 1 - 1 |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | firstname | nvarchar(50) | 100 | NULL allowed |  |
|  | lastname | nvarchar(50) | 100 | NULL allowed |  |
|  | photourl | nvarchar(500) | 1000 | NULL allowed |  |
|  | creationdate | datetime | 8 | NULL allowed |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | terms\_accepted | datetime | 8 | NULL allowed |  |
|  | receive\_news | bit | 1 | NULL allowed |  |
|  | companyname | nvarchar(250) | 500 | NULL allowed |  |
|  | phonenumber | nvarchar(50) | 100 | NULL allowed |  |
|  | language | nvarchar(50) | 100 | NULL allowed |  |
|  | country | nvarchar(5) | 10 | NULL allowed |  |
|  | companysize | tinyint | 1 | NULL allowed |  |
|  | companyindustry | tinyint | 1 | NULL allowed |  |
|  | companyrole | nvarchar(250) | 500 | NULL allowed |  |
|  | usecase | nvarchar(100) | 200 | NULL allowed |  |
|  | is\_guest | bit | 1 | NULL allowed |  |
|  | lastactivitydate | datetime | 8 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_user | userid | True |
|  | IX\_t\_user\_username | username | True |

SQL Script

|  |
| --- |
| CREATE TABLE [dbo].[t\_user]  (  [userid] [int] NOT NULL IDENTITY(1, 1),  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [firstname] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lastname] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [photourl] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [creationdate] [datetime] NULL,  [lastupdate] [datetime] NULL,  [terms\_accepted] [datetime] NULL,  [receive\_news] [bit] NULL,  [companyname] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [phonenumber] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [language] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [country] [nvarchar] (5) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [companysize] [tinyint] NULL,  [companyindustry] [tinyint] NULL,  [companyrole] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [usecase] [nvarchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [is\_guest] [bit] NULL,  [lastactivitydate] [datetime] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[t\_user] ADD CONSTRAINT [PK\_t\_user] PRIMARY KEY CLUSTERED ([userid]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [IX\_t\_user\_username] ON [dbo].[t\_user] ([username]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The user as used in tiles and projects', 'SCHEMA', N'dbo', 'TABLE', N't\_user', NULL, NULL  GO |

Used By

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

[[dbo].[p\_canvasshotter\_booking\_insert\_conditional]](#5iHeTvzxxnH/YtLAt0wKOhWh8Ls=)

[[dbo].[p\_online\_user\_get\_by\_connectionid]](#3hdddy7qEOm97dR0hNzxuENEE04=)

[[dbo].[p\_online\_user\_get\_by\_username]](#LGVJsJlf1/SngrX4ixAa0Bh5h/s=)

[[dbo].[p\_online\_user\_get\_projectid]](#NL9mm0bUGUv+swcXYJoRiYj6zW8=)

[[dbo].[p\_online\_user\_insert]](#qv2u/8StiVLEVqLtKYLw5mU2W1s=)

[[dbo].[p\_online\_user\_search\_backend]](#Djf8Cny5kYj9llD1X8X5SouWFZI=)

[[dbo].[p\_project\_get\_by\_id]](#KGz8phqiKp5J5UrsWlI+RqfLiH8=)

[[dbo].[p\_project\_get\_by\_userid]](#f+k/6T5SoQt87MBYuQ+KyD5PD58=)

[[dbo].[p\_project\_get\_coparticipating\_projects]](#D8T2UsmAddF87lP7WVAHYzS213o=)

[[dbo].[p\_project\_get\_coparticipating\_users]](#Qw+YCRyiWNeg8ihfRXKeBlbOdso=)

[[dbo].[p\_project\_get\_participating\_projects]](#y6U32mvf6Ydonzjyz9Alrl4pM2g=)

[[dbo].[p\_project\_search\_backend]](#xL4aGoOVsBGhh2gahaLWnst60v4=)

[[dbo].[p\_user\_get\_all]](#8ThubK+YA9MeX3EaP6ZtEMifJXM=)

[[dbo].[p\_user\_get\_by\_project]](#pLBDyYqwrvyK2QW29bKsZG0rbnY=)

[[dbo].[p\_user\_get\_by\_project\_count]](#NPhxaTlIkusKkdZ6XY0kKAydQWs=)

[[dbo].[p\_user\_get\_by\_userid]](#DRB0tZre192T3VYLVmPPCshxeo8=)

[[dbo].[p\_user\_get\_by\_username]](#zxQu9m3+uzpNcizGDb+D2z/qJS0=)

[[dbo].[p\_user\_insert]](#erG30Nokr2RkydueWcVzHMfNaT0=)

[[dbo].[p\_user\_update]](#6JsegXsk48YP0Mqt/WgjOedKN6k=)

|  |
| --- |
| [licensing].[t\_billing\_type] |

MS\_­Description

The type of billing

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | id | int | 4 | NOT NULL |
|  | name | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Billing­Type | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_billing\_type]  (  [id] [int] NOT NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_billing\_type] ADD CONSTRAINT [PK\_­Billing­Type] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The type of billing', 'SCHEMA', N'licensing', 'TABLE', N't\_billing\_type', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[p\_billing\_type\_delete]](#vg2gXloGvUNaZu9BAPnJNB00Tg4=)

[[licensing].[p\_billing\_type\_get\_all]](#PMj2YNV5pDGsp56mYX7pgjqA8xE=)

[[licensing].[p\_billing\_type\_get\_by\_id]](#JaB461hdLg32uaON30BvUrc8ZP4=)

[[licensing].[p\_billing\_type\_insert]](#+eFytUJ8juTRrqXTeInbRzxmdJM=)

[[licensing].[p\_billing\_type\_update]](#7tWSQ8KOlGyZnElc0WeQMxzAEyA=)

|  |
| --- |
| [licensing].[t\_customer] |

MS\_­Description

The registered customer

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 1 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | name | nvarchar(500) | 1000 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Customer | id | True |
|  | IX\_t\_customer\_username | username | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_customer]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [name] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_customer] ADD CONSTRAINT [PK\_­Customer] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE UNIQUE NONCLUSTERED INDEX [IX\_t\_customer\_username] ON [licensing].[t\_customer] ([username]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The registered customer', 'SCHEMA', N'licensing', 'TABLE', N't\_customer', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[p\_customer\_delete]](#stZpoeAkeKTIrN6NCVRIdVRFuXo=)

[[licensing].[p\_customer\_get\_all]](#7+yidD4QgvrfFCpIM5ZsfL51iHw=)

[[licensing].[p\_customer\_get\_by\_id]](#Raw4TRsrYeYr6m6E0PbCFwuH9/U=)

[[licensing].[p\_customer\_get\_by\_username]](#74gegkjrVkNFJyF7nHYrL5KVB/w=)

[[licensing].[p\_customer\_insert]](#m7i61yQ4jCHy5DZpBHm79Q7hR4k=)

[[licensing].[p\_customer\_search]](#PsoeVlJWRfP8hEFv80X0T1aRrPE=)

[[licensing].[p\_customer\_update]](#vXR0lcrCEpgSaMl3O6zdfbxDPxw=)

[[licensing].[p\_subscription\_search]](#lwhn7riTIvypHY0gkd39j/IiQ4w=)

|  |
| --- |
| [licensing].[t\_import\_data] |

MS\_­Description

Licensing data that has been imported from an external file

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:27 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity | Default |
|  | id | int | 4 | NOT NULL | 1 - 1 |  |
|  | type | int | 4 | NOT NULL |  |  |
|  | payload | nvarchar(max) | max | NOT NULL |  |  |
|  | creationdate | datetime | 8 | NOT NULL |  | (getutcdate()) |
|  | lastupdate | datetime | 8 | NULL allowed |  |  |
|  | booked\_by | uniqueidentifier | 16 | NULL allowed |  |  |
|  | booking\_date | datetime | 8 | NULL allowed |  |  |
|  | retry | int | 4 | NOT NULL |  | ((0)) |
|  | state\_machine\_status | int | 4 | NOT NULL |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Import­Data | id | True |
|  | IX\_t\_import\_data\_state\_machine\_status | state\_machine\_status |  |
|  | IX\_t\_import\_data\_type\_booked\_by | type, booked\_by |  |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_import\_data]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [type] [int] NOT NULL,  [payload] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [creationdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_import\_data\_creationdate] DEFAULT (getutcdate()),  [lastupdate] [datetime] NULL,  [booked\_by] [uniqueidentifier] NULL,  [booking\_date] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_import\_data\_retry] DEFAULT ((0)),  [state\_machine\_status] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_import\_data] ADD CONSTRAINT [PK\_­Import­Data] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_import\_data\_state\_machine\_status] ON [licensing].[t\_import\_data] ([state\_machine\_status]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_import\_data\_type\_booked\_by] ON [licensing].[t\_import\_data] ([type], [booked\_by]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Licensing data that has been imported from an external file', 'SCHEMA', N'licensing', 'TABLE', N't\_import\_data', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[p\_import\_data\_book]](#Ek+cO1q0ERS+e3XoWT8BuLJr4PE=)

[[licensing].[p\_import\_data\_delete]](#vcv96hVcYtP8SffqokjVQ18e5Lw=)

[[licensing].[p\_import\_data\_insert]](#bGEdfgEGMtl6BJhsSOr7g0K5/PM=)

[[licensing].[p\_import\_data\_unbook]](#uceApiKhjCnE0NqwYFf6UccE1P0=)

|  |
| --- |
| [licensing].[t\_license\_file\_history] |

MS\_­Description

History of all licensing files

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:27 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | customer | nvarchar(500) | 1000 | NULL allowed |  |
|  | license\_data | varbinary(8000) | 8000 | NOT NULL |  |
|  | created\_by | nvarchar(250) | 500 | NULL allowed |  |
|  | date\_created | datetime | 8 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_license\_file\_history | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_license\_file\_history]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [customer] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [license\_data] [varbinary] (8000) NOT NULL,  [created\_by] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [date\_created] [datetime] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_license\_file\_history] ADD CONSTRAINT [PK\_t\_license\_file\_history] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'History of all licensing files', 'SCHEMA', N'licensing', 'TABLE', N't\_license\_file\_history', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[p\_license\_file\_history\_get\_by\_id]](#JVUDyYDrGRP/udh9nVtCVo/4CC8=)

[[licensing].[p\_license\_file\_history\_insert]](#yZjMWthB2MAFeAZ578GrRsdaLs4=)

[[licensing].[p\_license\_file\_history\_search]](#hoe6ni5kflfJih5y4nJFJbDSpMg=)

|  |
| --- |
| [licensing].[t\_object\_type] |

MS\_­Description

Type of object used in look-up tables for licensing

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 10 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | id | int | 4 | NOT NULL |
|  | name | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_object\_type | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_object\_type]  (  [id] [int] NOT NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_object\_type] ADD CONSTRAINT [PK\_t\_object\_type] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Type of object used in look-up tables for licensing', 'SCHEMA', N'licensing', 'TABLE', N't\_object\_type', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_status]](#9aWO1xuBeb9B/Yli1IzB1MagQIA=)

|  |
| --- |
| [licensing].[t\_order] |

MS\_­Description

The orders

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | subscription\_id | int | 4 | NOT NULL |  |
|  | currency | nvarchar(5) | 10 | NULL allowed |  |
|  | promo\_code\_id | int | 4 | NULL allowed |  |
|  | payment\_method\_id | int | 4 | NOT NULL |  |
|  | created\_by\_username | nvarchar(250) | 500 | NULL allowed |  |
|  | date\_created | datetime | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |
|  | external\_reference | nvarchar(500) | 1000 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Order | id | True |
|  | IX\_t\_order\_subscription\_id | subscription\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Order\_­Payment­Method | payment\_method\_id->[[licensing].[t\_payment\_method].[id]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=) |
| FK\_­Order\_­Promo­Code | promo\_code\_id->[[licensing].[t\_promo\_code].[id]](#6oqplzrZY3UacWTAEhXbFt60GSQ=) |
| FK\_­Order\_­Subscription | subscription\_id->[[licensing].[t\_subscription].[id]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_order]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [subscription\_id] [int] NOT NULL,  [currency] [nvarchar] (5) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [promo\_code\_id] [int] NULL,  [payment\_method\_id] [int] NOT NULL,  [created\_by\_username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [date\_created] [datetime] NOT NULL,  [status] [tinyint] NOT NULL,  [external\_reference] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_order] ADD CONSTRAINT [PK\_­Order] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_order\_subscription\_id] ON [licensing].[t\_order] ([subscription\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_order] ADD CONSTRAINT [FK\_­Order\_­Payment­Method] FOREIGN KEY ([payment\_method\_id]) REFERENCES [licensing].[t\_payment\_method] ([id])  GO  ALTER TABLE [licensing].[t\_order] ADD CONSTRAINT [FK\_­Order\_­Promo­Code] FOREIGN KEY ([promo\_code\_id]) REFERENCES [licensing].[t\_promo\_code] ([id])  GO  ALTER TABLE [licensing].[t\_order] ADD CONSTRAINT [FK\_­Order\_­Subscription] FOREIGN KEY ([subscription\_id]) REFERENCES [licensing].[t\_subscription] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The orders', 'SCHEMA', N'licensing', 'TABLE', N't\_order', NULL, NULL  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

Used By

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[[licensing].[p\_order\_delete]](#EigYb0FdLvmqD5ujkVH/6Umwo+4=)

[[licensing].[p\_order\_get\_all]](#A3Tu40WQ/ocSFG2NdWRdaIgZCd0=)

[[licensing].[p\_order\_get\_by\_id]](#UN9WWgJIt/4Y7IL459GNob9qGzo=)

[[licensing].[p\_order\_insert]](#1l0/zq9mmHbzt9IESJrFmx5WuzI=)

[[licensing].[p\_order\_search]](#+Zd80IWktNnJI/CXVkqVbQsPW9g=)

[[licensing].[p\_order\_update]](#xHsh9BAMTe+93MluRECHypDBqi0=)

|  |
| --- |
| [licensing].[t\_orderline] |

MS\_­Description

The line in an order

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | order\_id | int | 4 | NOT NULL |  |
|  | product\_id | int | 4 | NOT NULL |  |
|  | number\_of\_users | int | 4 | NOT NULL |  |
|  | number\_of\_devices | int | 4 | NOT NULL |  |
|  | billing\_type\_id | int | 4 | NOT NULL |  |
|  | amount | money | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Order­Line | id | True |
|  | IX\_t\_orderline\_order\_id | order\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Order­Line\_­Billing­Type | billing\_type\_id->[[licensing].[t\_billing\_type].[id]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=) |
| FK\_­Order­Line\_­Order | order\_id->[[licensing].[t\_order].[id]](#Abm+/exewd8b+kfSgBGP3mREH2k=) |
| FK\_­Order­Line\_­Product | product\_id->[[licensing].[t\_product].[id]](#6EBiFkvfbRTw3F555fk44DFA3ks=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_orderline]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [order\_id] [int] NOT NULL,  [product\_id] [int] NOT NULL,  [number\_of\_users] [int] NOT NULL,  [number\_of\_devices] [int] NOT NULL,  [billing\_type\_id] [int] NOT NULL,  [amount] [money] NOT NULL,  [status] [tinyint] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_orderline] ADD CONSTRAINT [PK\_­Order­Line] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_orderline\_order\_id] ON [licensing].[t\_orderline] ([order\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_orderline] ADD CONSTRAINT [FK\_­Order­Line\_­Billing­Type] FOREIGN KEY ([billing\_type\_id]) REFERENCES [licensing].[t\_billing\_type] ([id])  GO  ALTER TABLE [licensing].[t\_orderline] ADD CONSTRAINT [FK\_­Order­Line\_­Order] FOREIGN KEY ([order\_id]) REFERENCES [licensing].[t\_order] ([id])  GO  ALTER TABLE [licensing].[t\_orderline] ADD CONSTRAINT [FK\_­Order­Line\_­Product] FOREIGN KEY ([product\_id]) REFERENCES [licensing].[t\_product] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The line in an order', 'SCHEMA', N'licensing', 'TABLE', N't\_orderline', NULL, NULL  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

Used By

[[licensing].[p\_order\_search]](#+Zd80IWktNnJI/CXVkqVbQsPW9g=)

[[licensing].[p\_orderline\_delete]](#yo04ar1RAsMNv4QWyHfSjILXSmQ=)

[[licensing].[p\_orderline\_get\_all]](#75r+ShR5FZAs2sh3QzjMImh1IqQ=)

[[licensing].[p\_orderline\_get\_by\_id]](#pk7kG1J7ovRjefewu2BcFkb/l3Y=)

[[licensing].[p\_orderline\_get\_by\_orderid]](#Q+kZj1w/IcBC1C1OqVHudyD9obc=)

[[licensing].[p\_orderline\_insert]](#pgykHfrEgxmWtg6+e/sVd4bvQNk=)

[[licensing].[p\_orderline\_update]](#u2XiJxBFrioPOznFtuO+R3VoAXM=)

|  |
| --- |
| [licensing].[t\_payment] |

MS\_­Description

The payments made

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | order\_id | int | 4 | NOT NULL |  |
|  | amount | money | 8 | NOT NULL |  |
|  | external\_reference | nvarchar(500) | 1000 | NULL allowed |  |
|  | date\_created | datetime | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_payment | id | True |
|  | IX\_t\_payment\_order\_id | order\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_payment\_t\_order | order\_id->[[licensing].[t\_order].[id]](#Abm+/exewd8b+kfSgBGP3mREH2k=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_payment]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [order\_id] [int] NOT NULL,  [amount] [money] NOT NULL,  [external\_reference] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [date\_created] [datetime] NOT NULL,  [status] [tinyint] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_payment] ADD CONSTRAINT [PK\_t\_payment] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_payment\_order\_id] ON [licensing].[t\_payment] ([order\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_payment] ADD CONSTRAINT [FK\_t\_payment\_t\_order] FOREIGN KEY ([order\_id]) REFERENCES [licensing].[t\_order] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The payments made', 'SCHEMA', N'licensing', 'TABLE', N't\_payment', NULL, NULL  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

Used By

[[licensing].[p\_payment\_delete]](#sW3O29K9LPbWKG9+BnhbiiZ2iqA=)

[[licensing].[p\_payment\_get\_all]](#hOJo6fZ7JQlYuXsKioti9JjAVG8=)

[[licensing].[p\_payment\_get\_by\_id]](#6PP2XG1JNzZ8lZJGAKCT3LbMqyo=)

[[licensing].[p\_payment\_insert]](#Vf3dZQR3PXwzJQjcb37g7d7VcGo=)

[[licensing].[p\_payment\_search]](#r80kFLotTafGBkTgvYp8OFtYnBw=)

[[licensing].[p\_payment\_update]](#rnmdjVVAOml4ZiCi1LePwTGB8J8=)

|  |
| --- |
| [licensing].[t\_payment\_method] |

MS\_­Description

The methods of payment

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | id | int | 4 | NOT NULL |
|  | name | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Payment­Method | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_payment\_method]  (  [id] [int] NOT NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_payment\_method] ADD CONSTRAINT [PK\_­Payment­Method] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The methods of payment', 'SCHEMA', N'licensing', 'TABLE', N't\_payment\_method', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[[licensing].[p\_payment\_method\_delete]](#ncnXnBCV/WC5YMn7pQfPGuxy1HQ=)

[[licensing].[p\_payment\_method\_get\_all]](#5lECTxGpiPr6xsX5nmzEnrUolCQ=)

[[licensing].[p\_payment\_method\_get\_by\_id]](#PwYn21M2sGlFfdw8Nhv5Lid7m3A=)

[[licensing].[p\_payment\_method\_insert]](#UqnW41eMtTkBuO1DoFLjs53LrDc=)

[[licensing].[p\_payment\_method\_update]](#YrXfOiz5Ce1PHt72DKgFyXDgbqM=)

|  |
| --- |
| [licensing].[t\_price] |

MS\_­Description

The price

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 24 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | pricelist\_id | int | 4 | NOT NULL |  |
|  | billing\_type\_id | int | 4 | NOT NULL |  |
|  | product\_id | int | 4 | NOT NULL |  |
|  | amount\_initial | money | 8 | NOT NULL |  |
|  | amount\_per\_user | money | 8 | NOT NULL |  |
|  | amount\_per\_device | money | 8 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Price | id | True |
|  | IX\_t\_price\_pricelist\_id | pricelist\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Price\_­Billing­Type | billing\_type\_id->[[licensing].[t\_billing\_type].[id]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=) |
| FK\_­Price\_­Pricelist | pricelist\_id->[[licensing].[t\_pricelist].[id]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=) |
| FK\_­Price\_­Product | product\_id->[[licensing].[t\_product].[id]](#6EBiFkvfbRTw3F555fk44DFA3ks=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_price]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [pricelist\_id] [int] NOT NULL,  [billing\_type\_id] [int] NOT NULL,  [product\_id] [int] NOT NULL,  [amount\_initial] [money] NOT NULL,  [amount\_per\_user] [money] NOT NULL,  [amount\_per\_device] [money] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_price] ADD CONSTRAINT [PK\_­Price] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_price\_pricelist\_id] ON [licensing].[t\_price] ([pricelist\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_price] ADD CONSTRAINT [FK\_­Price\_­Billing­Type] FOREIGN KEY ([billing\_type\_id]) REFERENCES [licensing].[t\_billing\_type] ([id])  GO  ALTER TABLE [licensing].[t\_price] ADD CONSTRAINT [FK\_­Price\_­Pricelist] FOREIGN KEY ([pricelist\_id]) REFERENCES [licensing].[t\_pricelist] ([id])  GO  ALTER TABLE [licensing].[t\_price] ADD CONSTRAINT [FK\_­Price\_­Product] FOREIGN KEY ([product\_id]) REFERENCES [licensing].[t\_product] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The price', 'SCHEMA', N'licensing', 'TABLE', N't\_price', NULL, NULL  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

Used By

[[licensing].[p\_price\_delete]](#7yq5yyx/eC926x3R57o+1G26lIw=)

[[licensing].[p\_price\_get\_all]](#jBbpnPWNgDMdhxWZpWcvZKE4uBI=)

[[licensing].[p\_price\_get\_by\_id]](#mIiMl/2xxNxxuY8Mz9ODP3OYOcI=)

[[licensing].[p\_price\_get\_by\_pricelistid]](#87jnmKQllPTudmDm4HhRauy9QL4=)

[[licensing].[p\_price\_insert]](#GcvNDY3H3byKAlXsXPajIGFWTPw=)

[[licensing].[p\_price\_update]](#hMy20Kl0XfPgaESO5O80Kfwd0ZA=)

|  |
| --- |
| [licensing].[t\_pricelist] |

MS\_­Description

The pricelist

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 3 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | name | nvarchar(500) | 1000 | NOT NULL |  |
|  | currency | nvarchar(5) | 10 | NOT NULL |  |
|  | active\_from | datetime | 8 | NOT NULL |  |
|  | active\_to | datetime | 8 | NOT NULL |  |
|  | is\_active | bit | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Pricelist | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_pricelist]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [name] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [currency] [nvarchar] (5) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [active\_from] [datetime] NOT NULL,  [active\_to] [datetime] NOT NULL,  [is\_active] [bit] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_pricelist] ADD CONSTRAINT [PK\_­Pricelist] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The pricelist', 'SCHEMA', N'licensing', 'TABLE', N't\_pricelist', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[[licensing].[p\_pricelist\_delete]](#jADKgqBjxt24G6rJVLeEmPYtDfQ=)

[[licensing].[p\_pricelist\_get\_all]](#4K9Ybtdg6rkRn2sgKFqw/qpFKLo=)

[[licensing].[p\_pricelist\_get\_by\_id]](#jGqwrvZPVIRpKSdkypF5RXzCxaA=)

[[licensing].[p\_pricelist\_insert]](#6TvK++zjKQgqsucsSsf6hsMPnfw=)

[[licensing].[p\_pricelist\_update]](#7D3GLHnCwfUBN0mooD6tWb3t+XA=)

|  |
| --- |
| [licensing].[t\_product] |

MS\_­Description

The products available for licensing

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 4 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | name | nvarchar(500) | 1000 | NOT NULL |  |
|  | description | nvarchar(max) | max | NULL allowed |  |
|  | product\_family\_code | nvarchar(100) | 200 | NOT NULL |  |
|  | min\_number\_of\_users | int | 4 | NOT NULL |  |
|  | max\_number\_of\_users | int | 4 | NULL allowed |  |
|  | min\_number\_of\_devices | int | 4 | NOT NULL |  |
|  | max\_number\_of\_devices | int | 4 | NULL allowed |  |
|  | external\_reference | nvarchar(500) | 1000 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Product | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_product]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [name] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [description] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [product\_family\_code] [nvarchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [min\_number\_of\_users] [int] NOT NULL,  [max\_number\_of\_users] [int] NULL,  [min\_number\_of\_devices] [int] NOT NULL,  [max\_number\_of\_devices] [int] NULL,  [external\_reference] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_product] ADD CONSTRAINT [PK\_­Product] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The products available for licensing', 'SCHEMA', N'licensing', 'TABLE', N't\_product', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[p\_product\_delete]](#0yQlK4RXZoBMCgohU3Tr05y5dXo=)

[[licensing].[p\_product\_get\_all]](#mG7He7twbIdITtsYe0mPN+FKtqw=)

[[licensing].[p\_product\_get\_by\_id]](#AL9A0IZVpoVamMNplmKopBu/HC4=)

[[licensing].[p\_product\_get\_by\_productfamily]](#d9koIjkTMqNAi6h4wOkRfAlwwFU=)

[[licensing].[p\_product\_insert]](#aJLPQNgYOmXCXpsBy2Y4WHdv7qo=)

[[licensing].[p\_product\_update]](#p2bEFY4v/8sVkniVLFF1OsQqGxk=)

[[licensing].[p\_subscription\_get\_by\_customerid\_productfamilycode]](#Eqcq8a+bqpwjB2XTA0kgOpQVImE=)

[[licensing].[p\_subscription\_user\_search]](#FtQXLxWS7Ltpw64vCnmLv2Hj89w=)

|  |
| --- |
| [licensing].[t\_promo\_code] |

MS\_­Description

Promo codes

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | promo\_type\_id | int | 4 | NOT NULL |  |
|  | code | nvarchar(100) | 200 | NOT NULL |  |
|  | description | nvarchar(1000) | 2000 | NULL allowed |  |
|  | product\_id | int | 4 | NULL allowed |  |
|  | max\_number\_of\_uses | int | 4 | NULL allowed |  |
|  | number\_of\_users | int | 4 | NULL allowed |  |
|  | number\_of\_devices | int | 4 | NULL allowed |  |
|  | number\_of\_free\_months | int | 4 | NULL allowed |  |
|  | active\_from | datetime | 8 | NULL allowed |  |
|  | active\_to | datetime | 8 | NULL allowed |  |
|  | discount\_amount | money | 8 | NULL allowed |  |
|  | discount\_percent | decimal(5,2) | 5 | NULL allowed |  |
|  | usage\_count | int | 4 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Promo­Code | id | True |
|  | IX\_t\_promo\_code\_code | code |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Promo­Code\_­Product | product\_id->[[licensing].[t\_product].[id]](#6EBiFkvfbRTw3F555fk44DFA3ks=) |
| FK\_­Promo­Code\_­Promo­Type | promo\_type\_id->[[licensing].[t\_promo\_type].[id]](#XWhkhxJf9mvneWNL4gIOGe4ENts=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_promo\_code]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [promo\_type\_id] [int] NOT NULL,  [code] [nvarchar] (100) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [description] [nvarchar] (1000) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [product\_id] [int] NULL,  [max\_number\_of\_uses] [int] NULL,  [number\_of\_users] [int] NULL,  [number\_of\_devices] [int] NULL,  [number\_of\_free\_months] [int] NULL,  [active\_from] [datetime] NULL,  [active\_to] [datetime] NULL,  [discount\_amount] [money] NULL,  [discount\_percent] [decimal] (5, 2) NULL,  [usage\_count] [int] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_promo\_code] ADD CONSTRAINT [PK\_­Promo­Code] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_promo\_code\_code] ON [licensing].[t\_promo\_code] ([code]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_promo\_code] ADD CONSTRAINT [FK\_­Promo­Code\_­Product] FOREIGN KEY ([product\_id]) REFERENCES [licensing].[t\_product] ([id])  GO  ALTER TABLE [licensing].[t\_promo\_code] ADD CONSTRAINT [FK\_­Promo­Code\_­Promo­Type] FOREIGN KEY ([promo\_type\_id]) REFERENCES [licensing].[t\_promo\_type] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Promo codes', 'SCHEMA', N'licensing', 'TABLE', N't\_promo\_code', NULL, NULL  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

Used By

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[[licensing].[p\_promo\_code\_delete]](#OpGkGZVyA4BSb9PHHq3pbA8qdsE=)

[[licensing].[p\_promo\_code\_get\_all]](#jn/4m7TsEzpKslzo5MpTnaMKe3Y=)

[[licensing].[p\_promo\_code\_get\_by\_code]](#wWw1FjA+m2gThQa53R9vugFSZyg=)

[[licensing].[p\_promo\_code\_get\_by\_id]](#0iyFjkYSBBn/eH+0fdx/eGO9pP4=)

[[licensing].[p\_promo\_code\_get\_by\_promotypeid]](#f+PJlqsB+iee2nH7TxRACZ7UK/U=)

[[licensing].[p\_promo\_code\_insert]](#ag0TjjocRDKqpsRij/iGO0TN7Hk=)

[[licensing].[p\_promo\_code\_update]](#6ZntmHL35uudvC/8cBW13cfg9m0=)

|  |
| --- |
| [licensing].[t\_promo\_type] |

MS\_­Description

The types of promo code

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | name | nvarchar(250) | 500 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_promo\_type | id | True |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_promo\_type]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_promo\_type] ADD CONSTRAINT [PK\_t\_promo\_type] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The types of promo code', 'SCHEMA', N'licensing', 'TABLE', N't\_promo\_type', NULL, NULL  GO |

Uses

[licensing]

Used By

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[[licensing].[p\_promo\_type\_delete]](#/i+5YjG2+gcdsXmggZdDijyux30=)

[[licensing].[p\_promo\_type\_get\_all]](#6nnCF+6G7J0uavIag4xaAqahcZI=)

[[licensing].[p\_promo\_type\_get\_by\_id]](#fZiUoRLUXPGtFq2C1ERX1XM63GY=)

[[licensing].[p\_promo\_type\_insert]](#zds++evrQWyZtzdVMN47ZykpFhU=)

[[licensing].[p\_promo\_type\_update]](#JoQNEezfB7Ctq3QWDSEeDgqpeoE=)

|  |
| --- |
| [licensing].[t\_status] |

MS\_­Description

The status of the licensing related objects

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 21 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | object\_type\_id | int | 4 | NOT NULL |
|  | status\_id | int | 4 | NOT NULL |
|  | name | nvarchar(250) | 500 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_status | object\_type\_id, status\_id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_status\_t\_object\_type | object\_type\_id->[[licensing].[t\_object\_type].[id]](#iVCrlYdN90WWswSpfUGP/SVaY7w=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_status]  (  [object\_type\_id] [int] NOT NULL,  [status\_id] [int] NOT NULL,  [name] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_status] ADD CONSTRAINT [PK\_t\_status] PRIMARY KEY CLUSTERED ([object\_type\_id], [status\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_status] ADD CONSTRAINT [FK\_t\_status\_t\_object\_type] FOREIGN KEY ([object\_type\_id]) REFERENCES [licensing].[t\_object\_type] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The status of the licensing related objects', 'SCHEMA', N'licensing', 'TABLE', N't\_status', NULL, NULL  GO |

Uses

[[licensing].[t\_object\_type]](#iVCrlYdN90WWswSpfUGP/SVaY7w=)

[licensing]

|  |
| --- |
| [licensing].[t\_subscription] |

MS\_­Description

Subscriptions

Properties

|  |  |
| --- | --- |
| Property | Value |
| Row Count (~) | 1 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | customer\_id | int | 4 | NOT NULL |  |
|  | product\_id | int | 4 | NOT NULL |  |
|  | billing\_type\_id | int | 4 | NOT NULL |  |
|  | number\_of\_users | int | 4 | NOT NULL |  |
|  | number\_of\_devices | int | 4 | NOT NULL |  |
|  | active\_from | datetime | 8 | NOT NULL |  |
|  | active\_to | datetime | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |
|  | date\_created | datetime | 8 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Subscription | id | True |
|  | IX\_t\_subscription\_customer\_id | customer\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Subscription\_­Billing­Type | billing\_type\_id->[[licensing].[t\_billing\_type].[id]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=) |
| FK\_­Subscription\_­Customer | customer\_id->[[licensing].[t\_customer].[id]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=) |
| FK\_­Subscription\_­Product | product\_id->[[licensing].[t\_product].[id]](#6EBiFkvfbRTw3F555fk44DFA3ks=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_subscription]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [customer\_id] [int] NOT NULL,  [product\_id] [int] NOT NULL,  [billing\_type\_id] [int] NOT NULL,  [number\_of\_users] [int] NOT NULL,  [number\_of\_devices] [int] NOT NULL,  [active\_from] [datetime] NOT NULL,  [active\_to] [datetime] NOT NULL,  [status] [tinyint] NOT NULL,  [date\_created] [datetime] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription] ADD CONSTRAINT [PK\_­Subscription] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_subscription\_customer\_id] ON [licensing].[t\_subscription] ([customer\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription] ADD CONSTRAINT [FK\_­Subscription\_­Billing­Type] FOREIGN KEY ([billing\_type\_id]) REFERENCES [licensing].[t\_billing\_type] ([id])  GO  ALTER TABLE [licensing].[t\_subscription] ADD CONSTRAINT [FK\_­Subscription\_­Customer] FOREIGN KEY ([customer\_id]) REFERENCES [licensing].[t\_customer] ([id])  GO  ALTER TABLE [licensing].[t\_subscription] ADD CONSTRAINT [FK\_­Subscription\_­Product] FOREIGN KEY ([product\_id]) REFERENCES [licensing].[t\_product] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Subscriptions', 'SCHEMA', N'licensing', 'TABLE', N't\_subscription', NULL, NULL  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

Used By

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[[licensing].[p\_order\_search]](#+Zd80IWktNnJI/CXVkqVbQsPW9g=)

[[licensing].[p\_subscription\_delete]](#QFCZ6Ii7ZB93S4FdFze/UElWHzw=)

[[licensing].[p\_subscription\_get\_all]](#SIrgPGxXvlV4FNipPBDsFdkzV5Y=)

[[licensing].[p\_subscription\_get\_by\_customerid]](#VrF8ENqCAdEFAAWK45zZM2vNzbo=)

[[licensing].[p\_subscription\_get\_by\_customerid\_productfamilycode]](#Eqcq8a+bqpwjB2XTA0kgOpQVImE=)

[[licensing].[p\_subscription\_get\_by\_id]](#TPWwcHuwcnnusHTrosCTD6LMS/Q=)

[[licensing].[p\_subscription\_get\_by\_subscriptiondevice]](#225pATZ7KewLBKmJFHmdFIAENo8=)

[[licensing].[p\_subscription\_get\_by\_subscriptionuser]](#tTJad1cWPnrLdIvzBtd86vYpybA=)

[[licensing].[p\_subscription\_insert]](#MMX7Y2afT72zjSWWeotCzrAh6Ik=)

[[licensing].[p\_subscription\_search]](#lwhn7riTIvypHY0gkd39j/IiQ4w=)

[[licensing].[p\_subscription\_update]](#Y1ehYjOfAxdvre7tC6sfEqYC/B8=)

[[licensing].[p\_subscription\_user\_search]](#FtQXLxWS7Ltpw64vCnmLv2Hj89w=)

|  |
| --- |
| [licensing].[t\_subscription\_device] |

MS\_­Description

The devices allowed to use a subscription

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:29 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | subscription\_id | int | 4 | NOT NULL |  |
|  | device\_code | nvarchar(250) | 500 | NOT NULL |  |
|  | device\_name | nvarchar(500) | 1000 | NOT NULL |  |
|  | date\_created | datetime | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Subscription\_­Device | id | True |
|  | IX\_t\_subscription\_device\_code | device\_code |  |
|  | IX\_t\_subscription\_device\_subscription\_id | subscription\_id |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Subscription\_­Device\_­Subscription | subscription\_id->[[licensing].[t\_subscription].[id]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_subscription\_device]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [subscription\_id] [int] NOT NULL,  [device\_code] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [device\_name] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [date\_created] [datetime] NOT NULL,  [status] [tinyint] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription\_device] ADD CONSTRAINT [PK\_­Subscription\_­Device] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_subscription\_device\_code] ON [licensing].[t\_subscription\_device] ([device\_code]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_subscription\_device\_subscription\_id] ON [licensing].[t\_subscription\_device] ([subscription\_id]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription\_device] ADD CONSTRAINT [FK\_­Subscription\_­Device\_­Subscription] FOREIGN KEY ([subscription\_id]) REFERENCES [licensing].[t\_subscription] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The devices allowed to use a subscription', 'SCHEMA', N'licensing', 'TABLE', N't\_subscription\_device', NULL, NULL  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

Used By

[[licensing].[p\_subscription\_device\_delete]](#OvW9HlRBjCiNzot5nv3QfsXOMRk=)

[[licensing].[p\_subscription\_device\_get\_all]](#ZNwcrfKOZjwN6gLDU2gsQShLg64=)

[[licensing].[p\_subscription\_device\_get\_by\_id]](#J2Ixk+CPTdOyzdV2pdjCwYxFxSY=)

[[licensing].[p\_subscription\_device\_get\_by\_subscriptionid]](#XJwL3gZ7pJ7ohOKLoOIDmmcQne8=)

[[licensing].[p\_subscription\_device\_get\_by\_subscriptionid\_devicecode]](#TqbljfXL9mvriG0RqnwK7WwwR6g=)

[[licensing].[p\_subscription\_device\_insert]](#kTCEvnUJK1U9T28z5paLXAjnCq0=)

[[licensing].[p\_subscription\_device\_update]](#hfaMAggSWYiwBchRNl7IpyHeHCs=)

[[licensing].[p\_subscription\_get\_by\_subscriptiondevice]](#225pATZ7KewLBKmJFHmdFIAENo8=)

|  |
| --- |
| [licensing].[t\_subscription\_user] |

MS\_­Description

The users allowed to use a subscription

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 1 |
| Created | 06:53:27 vrijdag 14 mei 2021 |
| Last Modified | 06:53:28 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Identity |
|  | id | int | 4 | NOT NULL | 1 - 1 |
|  | subscription\_id | int | 4 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | date\_created | datetime | 8 | NOT NULL |  |
|  | status | tinyint | 1 | NOT NULL |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Subscription\_­User | id | True |
|  | IX\_t\_subscription\_user\_subscription\_id | subscription\_id |  |
|  | IX\_t\_subscription\_user\_username | username |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Subscription\_­User\_­Subscription | subscription\_id->[[licensing].[t\_subscription].[id]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [licensing].[t\_subscription\_user]  (  [id] [int] NOT NULL IDENTITY(1, 1),  [subscription\_id] [int] NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [date\_created] [datetime] NOT NULL,  [status] [tinyint] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription\_user] ADD CONSTRAINT [PK\_­Subscription\_­User] PRIMARY KEY CLUSTERED ([id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_subscription\_user\_subscription\_id] ON [licensing].[t\_subscription\_user] ([subscription\_id]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_subscription\_user\_username] ON [licensing].[t\_subscription\_user] ([username]) ON [PRIMARY]  GO  ALTER TABLE [licensing].[t\_subscription\_user] ADD CONSTRAINT [FK\_­Subscription\_­User\_­Subscription] FOREIGN KEY ([subscription\_id]) REFERENCES [licensing].[t\_subscription] ([id])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'The users allowed to use a subscription', 'SCHEMA', N'licensing', 'TABLE', N't\_subscription\_user', NULL, NULL  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

Used By

[[licensing].[p\_subscription\_get\_by\_subscriptionuser]](#tTJad1cWPnrLdIvzBtd86vYpybA=)

[[licensing].[p\_subscription\_user\_delete]](#PgAR8zhkf9scprZhWYeXceAht+4=)

[[licensing].[p\_subscription\_user\_get\_all]](#ZwvfnamhlF8i4TI05qX7rsH8lFw=)

[[licensing].[p\_subscription\_user\_get\_by\_id]](#uZ6svt0rbDgzpf5wg+dLE5i741Y=)

[[licensing].[p\_subscription\_user\_get\_by\_subscriptionid\_username]](#+9OG++c2Z91qAZmkSNfUF+BLiaU=)

[[licensing].[p\_subscription\_user\_insert]](#DDN48PXPjMZo8/enrDvN1VXcExs=)

[[licensing].[p\_subscription\_user\_search]](#FtQXLxWS7Ltpw64vCnmLv2Hj89w=)

[[licensing].[p\_subscription\_user\_update]](#+5YEV95dtYx+ZRRIxHbuRMji5/4=)

|  |
| --- |
| [mft].[t\_chunkingmachine] |

MS\_­Description

State machine for the chunking service (download Blobs)

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | chunkingmachineid | uniqueidentifier | 16 | NOT NULL |  |
|  | chunkingmachinestatusid | int | 4 | NOT NULL | ((0)) |
|  | domain | nvarchar(250) | 500 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | password | nvarchar(500) | 1000 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NOT NULL |  |
|  | filepath | nvarchar(500) | 1000 | NOT NULL |  |
|  | filename | nvarchar(500) | 1000 | NOT NULL |  |
|  | filecheksum | nvarchar(128) | 256 | NOT NULL |  |
|  | temppath | nvarchar(500) | 1000 | NOT NULL |  |
|  | buffersize | int | 4 | NOT NULL | ((0)) |
|  | chunks | int | 4 | NOT NULL | ((0)) |
|  | updateby | nvarchar(50) | 100 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NOT NULL | (getutcdate()) |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_chunkingmachine | chunkingmachineid | True |
|  | IX\_t\_chunkingmachine\_chunkingmachinestatusid | chunkingmachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_chunkingmachine\_t\_chunkingmachine\_status | chunkingmachinestatusid->[[mft].[t\_chunkingmachine\_status].[chunkingmachinestatusid]](#orTfJWER9t2qKsaqUNd9g7WUDiQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_chunkingmachine]  (  [chunkingmachineid] [uniqueidentifier] NOT NULL,  [chunkingmachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_chunkingmachinestatusid] DEFAULT ((0)),  [domain] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [password] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [machineid] [uniqueidentifier] NOT NULL,  [filepath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filecheksum] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [temppath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [buffersize] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_buffersize] DEFAULT ((0)),  [chunks] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_chunks] DEFAULT ((0)),  [updateby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [lastupdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_lastupdate] DEFAULT (getutcdate()),  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_retry] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_chunkingmachine] ADD CONSTRAINT [PK\_t\_chunkingmachine] PRIMARY KEY CLUSTERED ([chunkingmachineid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_chunkingmachine\_chunkingmachinestatusid] ON [mft].[t\_chunkingmachine] ([chunkingmachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_chunkingmachine] ADD CONSTRAINT [FK\_t\_chunkingmachine\_t\_chunkingmachine\_status] FOREIGN KEY ([chunkingmachinestatusid]) REFERENCES [mft].[t\_chunkingmachine\_status] ([chunkingmachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the chunking service (download Blobs)', 'SCHEMA', N'mft', 'TABLE', N't\_chunkingmachine', NULL, NULL  GO |

Uses

[[mft].[t\_chunkingmachine\_status]](#orTfJWER9t2qKsaqUNd9g7WUDiQ=)

[mft]

Used By

[[mft].[p\_check\_download\_available]](#9OQhqSWAVHD+UBy5XRVnUeTSWWU=)

[[mft].[p\_chunkingmachine\_book]](#PnS53PCRBYBj/nxp1iARCISBVY8=)

[[mft].[p\_chunkingmachine\_completed]](#GbqFV8ZfXcU29GDG9v8bxnOw5Us=)

[[mft].[p\_chunkingmachine\_insert]](#f6LK3IajM3RnECpMSzPHWBs1WoU=)

[[mft].[p\_chunkingmachine\_unbook]](#6o9FQNq5KnrXXUlc5B0tjGAOujg=)

[[mft].[p\_cleaningmachine\_bookchunking]](#MwISX6HBX5uwfbo4eD1sPM5Ba+w=)

[[mft].[p\_cleaningmachine\_move\_chunkinghistory]](#SFqDu+Wsq3Xfrp/t8dMSsrI3VAM=)

[[mft].[p\_cleaningmachine\_unbookchunking]](#ZSgOCGEe9avmVuDxDZqiS/jwcis=)

[[mft].[p\_cleaningmachinenotcompleted\_bookchunking]](#DQUu0qPhdSDXCubN8KtH3sGAOos=)

[[mft].[p\_cleaningmachinenotcompleted\_bookchunkingmoving]](#BlF5NLkxVeX1k7IbVyiELdR7Yzc=)

[[mft].[p\_get\_activity\_from\_chunking]](#qXRdjSXddfxXIHJ7Eym6hAXQJDM=)

|  |
| --- |
| [mft].[t\_chunkingmachine\_history] |

MS\_­Description

State machine for the chunking service history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | chunkingmachineid | uniqueidentifier | 16 | NOT NULL |  |
|  | chunkingmachinestatusid | int | 4 | NOT NULL | ((0)) |
|  | domain | nvarchar(250) | 500 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | password | nvarchar(500) | 1000 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NOT NULL |  |
|  | filepath | nvarchar(500) | 1000 | NOT NULL |  |
|  | filename | nvarchar(500) | 1000 | NOT NULL |  |
|  | filecheksum | nvarchar(128) | 256 | NOT NULL |  |
|  | temppath | nvarchar(500) | 1000 | NOT NULL |  |
|  | buffersize | int | 4 | NOT NULL | ((0)) |
|  | chunks | int | 4 | NOT NULL | ((0)) |
|  | updateby | nvarchar(50) | 100 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NOT NULL | (getutcdate()) |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
|  | moved\_on | datetime | 8 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_chunkingmachine\_history | chunkingmachineid | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_chunkingmachine\_history\_t\_chunkingmachine\_status | chunkingmachinestatusid->[[mft].[t\_chunkingmachine\_status].[chunkingmachinestatusid]](#orTfJWER9t2qKsaqUNd9g7WUDiQ=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_chunkingmachine\_history]  (  [chunkingmachineid] [uniqueidentifier] NOT NULL,  [chunkingmachinestatusid] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_history\_chunkingmachinestatusid] DEFAULT ((0)),  [domain] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [password] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [machineid] [uniqueidentifier] NOT NULL,  [filepath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filecheksum] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [temppath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [buffersize] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_history\_buffersize] DEFAULT ((0)),  [chunks] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_history\_chunks] DEFAULT ((0)),  [updateby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [lastupdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_history\_lastupdate] DEFAULT (getutcdate()),  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_chunkingmachine\_history\_retry] DEFAULT ((0)),  [moved\_on] [datetime] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_chunkingmachine\_history] ADD CONSTRAINT [PK\_t\_chunkingmachine\_history] PRIMARY KEY CLUSTERED ([chunkingmachineid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_chunkingmachine\_history] ADD CONSTRAINT [FK\_t\_chunkingmachine\_history\_t\_chunkingmachine\_status] FOREIGN KEY ([chunkingmachinestatusid]) REFERENCES [mft].[t\_chunkingmachine\_status] ([chunkingmachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the chunking service history', 'SCHEMA', N'mft', 'TABLE', N't\_chunkingmachine\_history', NULL, NULL  GO |

Uses

[[mft].[t\_chunkingmachine\_status]](#orTfJWER9t2qKsaqUNd9g7WUDiQ=)

[mft]

Used By

[[mft].[p\_cleaningmachine\_move\_chunkinghistory]](#SFqDu+Wsq3Xfrp/t8dMSsrI3VAM=)

|  |
| --- |
| [mft].[t\_chunkingmachine\_status] |

MS\_­Description

Status of the chunking service state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 9 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | chunkingmachinestatusid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_chunkingmachine\_status | chunkingmachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_chunkingmachine\_status]  (  [chunkingmachinestatusid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_chunkingmachine\_status] ADD CONSTRAINT [PK\_t\_chunkingmachine\_status] PRIMARY KEY CLUSTERED ([chunkingmachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Status of the chunking service state machine', 'SCHEMA', N'mft', 'TABLE', N't\_chunkingmachine\_status', NULL, NULL  GO |

Uses

[mft]

Used By

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[t\_chunkingmachine\_history]](#aU/tVVaTKU0VGvpN8lOEMo7OrWQ=)

|  |
| --- |
| [mft].[t\_mergemachine] |

MS\_­Description

State machine for the merging service (upload Blobs)

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | mergemachineid | uniqueidentifier | 16 | NOT NULL |  |
|  | mergemachinestatusid | int | 4 | NOT NULL | ((-1)) |
|  | domain | nvarchar(250) | 500 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | password | nvarchar(500) | 1000 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NOT NULL |  |
|  | filepath | nvarchar(500) | 1000 | NOT NULL |  |
|  | filename | nvarchar(500) | 1000 | NOT NULL |  |
|  | filecheksum | nvarchar(128) | 256 | NOT NULL |  |
|  | temppath | nvarchar(500) | 1000 | NOT NULL |  |
|  | chunks | int | 4 | NOT NULL | ((0)) |
|  | updateby | nvarchar(50) | 100 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NOT NULL | (getutcdate()) |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_mergemachine | mergemachineid | True |
|  | IX\_t\_mergemachine\_mergemachinestatusid | mergemachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_mergemachine\_t\_mergemachine\_status | mergemachinestatusid->[[mft].[t\_mergemachine\_status].[mergemachinestatusid]](#fAT3wGUeZiRwTusB4E+KrU+Trcc=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_mergemachine]  (  [mergemachineid] [uniqueidentifier] NOT NULL,  [mergemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_­Table\_1\_chunkingmachinestatusid] DEFAULT ((-1)),  [domain] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [password] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [machineid] [uniqueidentifier] NOT NULL,  [filepath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filecheksum] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [temppath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [chunks] [int] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_chunks] DEFAULT ((0)),  [updateby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [lastupdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_lastupdate] DEFAULT (getutcdate()),  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_retry] DEFAULT ((0))  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_mergemachine] ADD CONSTRAINT [PK\_t\_mergemachine] PRIMARY KEY CLUSTERED ([mergemachineid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_mergemachine\_mergemachinestatusid] ON [mft].[t\_mergemachine] ([mergemachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_mergemachine] ADD CONSTRAINT [FK\_t\_mergemachine\_t\_mergemachine\_status] FOREIGN KEY ([mergemachinestatusid]) REFERENCES [mft].[t\_mergemachine\_status] ([mergemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the merging service (upload Blobs)', 'SCHEMA', N'mft', 'TABLE', N't\_mergemachine', NULL, NULL  GO |

Uses

[[mft].[t\_mergemachine\_status]](#fAT3wGUeZiRwTusB4E+KrU+Trcc=)

[mft]

Used By

[[mft].[p\_check\_upload\_finished]](#88BhiFm5uHwIoRoVhAIAiPjw0sw=)

[[mft].[p\_cleaningmachine\_bookmergemoving]](#lr7RkXHqmwkD5T1jKtSL+cyVZn8=)

[[mft].[p\_cleaningmachine\_bookmerging]](#PcBMt0tIPqIsieGQG+6oFWBJhXQ=)

[[mft].[p\_cleaningmachine\_move\_mergehistory]](#BZi0PPVjQ18TUJWBNm6Fb3uwbaI=)

[[mft].[p\_cleaningmachine\_unbookmerging]](#u6M/KQdFp+HD6diKs1auoaoNYME=)

[[mft].[p\_cleaningmachinenotcompleted\_bookmerging]](#BGEQIhIkhDirIWgbHwqOmRwJtIk=)

[[mft].[p\_get\_activity\_from\_merge]](#3yUdhmqUMo+5TUAe4NtlDssCtbs=)

[[mft].[p\_mergemachine\_book]](#pNy6chimtMu1ga2KNNPvnYdyfzA=)

[[mft].[p\_mergemachine\_insert]](#iUuoicTqGrjxPJNYyujMR6ek43U=)

[[mft].[p\_mergemachine\_ready\_to\_process]](#Dh290Ssk5A7Vl7lE4ssUSvb0/To=)

[[mft].[p\_mergemachine\_unbook]](#h9rJTw3uKsvsNflnarbRyuFacLM=)

|  |
| --- |
| [mft].[t\_mergemachine\_history] |

MS\_­Description

State machine for the merging service history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | mergemachineid | uniqueidentifier | 16 | NOT NULL |  |
|  | mergemachinestatusid | int | 4 | NOT NULL | ((-1)) |
|  | domain | nvarchar(250) | 500 | NOT NULL |  |
|  | username | nvarchar(250) | 500 | NOT NULL |  |
|  | password | nvarchar(500) | 1000 | NOT NULL |  |
|  | machineid | uniqueidentifier | 16 | NOT NULL |  |
|  | filepath | nvarchar(500) | 1000 | NOT NULL |  |
|  | filename | nvarchar(500) | 1000 | NOT NULL |  |
|  | filecheksum | nvarchar(128) | 256 | NOT NULL |  |
|  | temppath | nvarchar(500) | 1000 | NOT NULL |  |
|  | chunks | int | 4 | NOT NULL | ((0)) |
|  | updateby | nvarchar(50) | 100 | NOT NULL |  |
|  | lastupdate | datetime | 8 | NOT NULL | (getutcdate()) |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NOT NULL | ((0)) |
|  | moved\_on | datetime | 8 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_mergemachine\_history | mergemachineid | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_t\_mergemachine\_history\_t\_mergemachine\_status | mergemachinestatusid->[[mft].[t\_mergemachine\_status].[mergemachinestatusid]](#fAT3wGUeZiRwTusB4E+KrU+Trcc=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_mergemachine\_history]  (  [mergemachineid] [uniqueidentifier] NOT NULL,  [mergemachinestatusid] [int] NOT NULL CONSTRAINT [DF\_chunkingmachinestatusid\_history] DEFAULT ((-1)),  [domain] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [username] [nvarchar] (250) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [password] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [machineid] [uniqueidentifier] NOT NULL,  [filepath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filename] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [filecheksum] [nvarchar] (128) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [temppath] [nvarchar] (500) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [chunks] [int] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_chunks\_history] DEFAULT ((0)),  [updateby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL,  [lastupdate] [datetime] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_lastupdate\_history] DEFAULT (getutcdate()),  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NOT NULL CONSTRAINT [DF\_t\_mergemachine\_retry\_history] DEFAULT ((0)),  [moved\_on] [datetime] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_mergemachine\_history] ADD CONSTRAINT [PK\_t\_mergemachine\_history] PRIMARY KEY CLUSTERED ([mergemachineid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_mergemachine\_history] ADD CONSTRAINT [FK\_t\_mergemachine\_history\_t\_mergemachine\_status] FOREIGN KEY ([mergemachinestatusid]) REFERENCES [mft].[t\_mergemachine\_status] ([mergemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the merging service history', 'SCHEMA', N'mft', 'TABLE', N't\_mergemachine\_history', NULL, NULL  GO |

Uses

[[mft].[t\_mergemachine\_status]](#fAT3wGUeZiRwTusB4E+KrU+Trcc=)

[mft]

Used By

[[mft].[p\_cleaningmachine\_move\_mergehistory]](#BZi0PPVjQ18TUJWBNm6Fb3uwbaI=)

|  |
| --- |
| [mft].[t\_mergemachine\_status] |

MS\_­Description

Status of the merging service state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 12 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | mergemachinestatusid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_mergemachine\_status | mergemachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_mergemachine\_status]  (  [mergemachinestatusid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_mergemachine\_status] ADD CONSTRAINT [PK\_t\_mergemachine\_status] PRIMARY KEY CLUSTERED ([mergemachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Status of the merging service state machine', 'SCHEMA', N'mft', 'TABLE', N't\_mergemachine\_status', NULL, NULL  GO |

Uses

[mft]

Used By

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[t\_mergemachine\_history]](#A4S2F2HPlFcxaD6FNedLE5yGu/A=)

|  |
| --- |
| [mft].[t\_state\_machine\_status] |

MS\_­Description

All states for the MFT state machines

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 5 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:39 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | statemachinestatusid | int | 4 | NOT NULL |
|  | statemachinestatusdescription | nvarchar(50) | 100 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_state\_machine\_status | statemachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_state\_machine\_status]  (  [statemachinestatusid] [int] NOT NULL,  [statemachinestatusdescription] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_state\_machine\_status] ADD CONSTRAINT [PK\_t\_state\_machine\_status] PRIMARY KEY CLUSTERED ([statemachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'All states for the MFT state machines', 'SCHEMA', N'mft', 'TABLE', N't\_state\_machine\_status', NULL, NULL  GO |

Uses

[mft]

|  |
| --- |
| [mft].[t\_storagemachine\_status] |

MS\_­Description

State of the Storage state machine

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 5 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | storagemachinestatusid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NOT NULL |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_storagemachine\_status | storagemachinestatusid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_storagemachine\_status]  (  [storagemachinestatusid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storagemachine\_status] ADD CONSTRAINT [PK\_t\_storagemachine\_status] PRIMARY KEY CLUSTERED ([storagemachinestatusid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State of the Storage state machine ', 'SCHEMA', N'mft', 'TABLE', N't\_storagemachine\_status', NULL, NULL  GO |

Uses

[mft]

Used By

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[t\_storageoperations\_history]](#Wikc4aMSZrWQB6JETj0eWwnYvU0=)

|  |
| --- |
| [mft].[t\_storageoperations] |

MS\_­Description

State machine for the Storage service (file operations)

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | storageoperationsid | uniqueidentifier | 16 | NOT NULL | (newid()) |
|  | storageoperationsmachinestatusid | int | 4 | NOT NULL |  |
|  | storageoperationtype | int | 4 | NOT NULL |  |
|  | updatedby | nvarchar(50) | 100 | NULL allowed |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NULL allowed |  |
|  | action1 | nvarchar(max) | max | NULL allowed |  |
|  | action2 | nvarchar(max) | max | NULL allowed |  |
|  | username | nvarchar(50) | 100 | NULL allowed |  |
|  | password | nvarchar(max) | max | NULL allowed |  |
|  | domain | nvarchar(50) | 100 | NULL allowed |  |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_storageoperations | storageoperationsid | True |
|  | IX\_t\_storageoperations\_storageoperationsmachinestatusid | storageoperationsmachinestatusid |  |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_\_t\_storage\_\_stora\_\_5D95E53A | storageoperationtype->[[mft].[t\_storageoperationtype].[storageoperationtypeid]](#qO3E1mf5h5nTxmZrP6eixHVA/6k=) |
| FK\_\_t\_storage\_\_stora\_\_5E8A0973 | storageoperationsmachinestatusid->[[mft].[t\_storagemachine\_status].[storagemachinestatusid]](#fzTCNIvpwKpEL9bwvmEgtn5jvxk=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_storageoperations]  (  [storageoperationsid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_t\_storageoperations\_storageoperationsid] DEFAULT (newid()),  [storageoperationsmachinestatusid] [int] NOT NULL,  [storageoperationtype] [int] NOT NULL,  [updatedby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NULL,  [action1] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [action2] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [username] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [password] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [domain] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storageoperations] ADD CONSTRAINT [PK\_t\_storageoperations] PRIMARY KEY CLUSTERED ([storageoperationsid]) ON [PRIMARY]  GO  CREATE NONCLUSTERED INDEX [IX\_t\_storageoperations\_storageoperationsmachinestatusid] ON [mft].[t\_storageoperations] ([storageoperationsmachinestatusid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storageoperations] ADD CONSTRAINT [FK\_\_t\_storage\_\_stora\_\_5D95E53A] FOREIGN KEY ([storageoperationtype]) REFERENCES [mft].[t\_storageoperationtype] ([storageoperationtypeid])  GO  ALTER TABLE [mft].[t\_storageoperations] ADD CONSTRAINT [FK\_\_t\_storage\_\_stora\_\_5E8A0973] FOREIGN KEY ([storageoperationsmachinestatusid]) REFERENCES [mft].[t\_storagemachine\_status] ([storagemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the Storage service (file operations)', 'SCHEMA', N'mft', 'TABLE', N't\_storageoperations', NULL, NULL  GO |

Uses

[[mft].[t\_storagemachine\_status]](#fzTCNIvpwKpEL9bwvmEgtn5jvxk=)

[[mft].[t\_storageoperationtype]](#qO3E1mf5h5nTxmZrP6eixHVA/6k=)

[mft]

Used By

[[mft].[p\_check\_storageoperation\_finished]](#YWckhcm8yx4WBJg+di4+9EJyP0Y=)

[[mft].[p\_get\_activity\_from\_storage]](#9e7ywF3uhRnjGcJZ4e9+VeUE340=)

[[mft].[p\_storagemachine\_book]](#HjNI/SX3pG0VAK1u/2UQ3RjjiQc=)

[[mft].[p\_storagemachine\_insert]](#WiPK2k7Zanqsa370TGyEzfu2RLc=)

[[mft].[p\_storagemachine\_unbook]](#q4B3C6jmuIZfKqfaSkV2LMG4jSI=)

[[mft].[p\_storageoperations\_move\_history]](#HzyPp0rH2W7p+J3+QqOKqUUnxuM=)

|  |
| --- |
| [mft].[t\_storageoperations\_history] |

MS\_­Description

State machine for the Storage service history

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 0 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability | Default |
|  | storageoperationsid | uniqueidentifier | 16 | NOT NULL | (newid()) |
|  | storageoperationsmachinestatusid | int | 4 | NOT NULL |  |
|  | storageoperationtype | int | 4 | NOT NULL |  |
|  | updatedby | nvarchar(50) | 100 | NULL allowed |  |
|  | lastupdate | datetime | 8 | NULL allowed |  |
|  | bookedby | uniqueidentifier | 16 | NULL allowed |  |
|  | bookingdate | datetime | 8 | NULL allowed |  |
|  | retry | int | 4 | NULL allowed |  |
|  | action1 | nvarchar(max) | max | NULL allowed |  |
|  | action2 | nvarchar(max) | max | NULL allowed |  |
|  | username | nvarchar(50) | 100 | NULL allowed |  |
|  | password | nvarchar(max) | max | NULL allowed |  |
|  | domain | nvarchar(50) | 100 | NULL allowed |  |
|  | machineid | uniqueidentifier | 16 | NULL allowed |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_storageoperations\_history | storageoperationsid | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_\_t\_storage\_history\_\_stora\_\_5D95E53A | storageoperationtype->[[mft].[t\_storageoperationtype].[storageoperationtypeid]](#qO3E1mf5h5nTxmZrP6eixHVA/6k=) |
| FK\_\_t\_storage\_history\_\_stora\_\_5E8A0973 | storageoperationsmachinestatusid->[[mft].[t\_storagemachine\_status].[storagemachinestatusid]](#fzTCNIvpwKpEL9bwvmEgtn5jvxk=) |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_storageoperations\_history]  (  [storageoperationsid] [uniqueidentifier] NOT NULL ROWGUIDCOL CONSTRAINT [DF\_t\_storageoperations\_history\_storageoperationsid] DEFAULT (newid()),  [storageoperationsmachinestatusid] [int] NOT NULL,  [storageoperationtype] [int] NOT NULL,  [updatedby] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [lastupdate] [datetime] NULL,  [bookedby] [uniqueidentifier] NULL,  [bookingdate] [datetime] NULL,  [retry] [int] NULL,  [action1] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [action2] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [username] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [password] [nvarchar] (max) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [domain] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL,  [machineid] [uniqueidentifier] NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storageoperations\_history] ADD CONSTRAINT [PK\_t\_storageoperations\_history] PRIMARY KEY CLUSTERED ([storageoperationsid]) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storageoperations\_history] ADD CONSTRAINT [FK\_\_t\_storage\_history\_\_stora\_\_5D95E53A] FOREIGN KEY ([storageoperationtype]) REFERENCES [mft].[t\_storageoperationtype] ([storageoperationtypeid])  GO  ALTER TABLE [mft].[t\_storageoperations\_history] ADD CONSTRAINT [FK\_\_t\_storage\_history\_\_stora\_\_5E8A0973] FOREIGN KEY ([storageoperationsmachinestatusid]) REFERENCES [mft].[t\_storagemachine\_status] ([storagemachinestatusid])  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'State machine for the Storage service history', 'SCHEMA', N'mft', 'TABLE', N't\_storageoperations\_history', NULL, NULL  GO |

Uses

[[mft].[t\_storagemachine\_status]](#fzTCNIvpwKpEL9bwvmEgtn5jvxk=)

[[mft].[t\_storageoperationtype]](#qO3E1mf5h5nTxmZrP6eixHVA/6k=)

[mft]

Used By

[[mft].[p\_storageoperations\_move\_history]](#HzyPp0rH2W7p+J3+QqOKqUUnxuM=)

|  |
| --- |
| [mft].[t\_storageoperationtype] |

MS\_­Description

Storage state machine operation types

Properties

|  |  |
| --- | --- |
| Property | Value |
| Collation | SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS |
| Row Count (~) | 6 |
| Created | 06:53:39 vrijdag 14 mei 2021 |
| Last Modified | 06:53:40 vrijdag 14 mei 2021 |

Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Nullability |
|  | storageoperationtypeid | int | 4 | NOT NULL |
|  | description | nvarchar(50) | 100 | NULL allowed |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_t\_storageoperationtype | storageoperationtypeid | True |

SQL Script

|  |
| --- |
| CREATE TABLE [mft].[t\_storageoperationtype]  (  [storageoperationtypeid] [int] NOT NULL,  [description] [nvarchar] (50) COLLATE SQL\_­Latin1\_­General\_­CP1\_­CI\_­AS NULL  ) ON [PRIMARY]  GO  ALTER TABLE [mft].[t\_storageoperationtype] ADD CONSTRAINT [PK\_t\_storageoperationtype] PRIMARY KEY CLUSTERED ([storageoperationtypeid]) ON [PRIMARY]  GO  EXEC sp\_addextendedproperty N'MS\_­Description', N'Storage state machine operation types', 'SCHEMA', N'mft', 'TABLE', N't\_storageoperationtype', NULL, NULL  GO |

Uses

[mft]

Used By

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[t\_storageoperations\_history]](#Wikc4aMSZrWQB6JETj0eWwnYvU0=)

|  |
| --- |
| Stored Procedures |

Objects

|  |
| --- |
| Name |
| [auth.p\_user\_add\_role](#ck5ZcQ+YNnKJyGtrky4Tlwi1VHY=) |
| [auth.p\_user\_delete](#KPWoxg6bGn3VOhpJdtEAD584I3I=) |
| [auth.p\_user\_get\_by\_username](#2PVxAKdNDzeBHG1fDzq92gVGxKw=) |
| [auth.p\_user\_get\_password](#yIEMH4Nl3SYhz4iZlZJIOvqJinM=) |
| [auth.p\_user\_get\_roles](#mYS79/VnbJ6BW6+5ChFc46J5CPY=) |
| [auth.p\_user\_insert](#lHytnjCFoXFrDfVlt1SmfGVhBH0=) |
| [auth.p\_user\_remove\_role](#prGivRZpVkEHaarCD80Rh2XIA6E=) |
| [auth.p\_user\_update](#y/hmn19lbyzRoHunXWnYBxvPeaU=) |
| [auth.p\_user\_update\_password](#yMG4qktZRlbNCOQri9AyYQaYh4Y=) |
| [conversion.p\_conversion\_book](#2ky7nZ3brsF0UI6fkxfmgT8GZqI=) |
| [conversion.p\_conversion\_book\_fallback](#N/g2rojWTtdssTZ449KkIIx/4B8=) |
| [conversion.p\_conversion\_book\_history](#Fndejxz7Ohz6SURqC1Fm5gQZiYQ=) |
| [conversion.p\_conversion\_insert](#ZAAI0TdWV+PMPe0e2gKthXBu6VY=) |
| [conversion.p\_conversion\_unbook](#VOMFu7g1zPWBppeP3UenCTegvVk=) |
| [conversion.p\_image\_resizing\_book](#8H9bGc3/Fy7+H4uFbIgZhvy1agY=) |
| [conversion.p\_image\_resizing\_book\_fallback](#s8FUIxTi+6LBh8mYurqnSND9xLQ=) |
| [conversion.p\_image\_resizing\_book\_history](#1xpeiE5gtXIQzKAuzLeH5MY+gHc=) |
| [conversion.p\_image\_resizing\_booking\_insert](#lJxGVVKhAjqYDYdtS1TeWbGi0Ko=) |
| [conversion.p\_image\_resizing\_unbook](#RlPaP7QJJSbXEeDJQW4D5ItTc0Q=) |
| [dbo.p\_cache\_load](#x5NpJBEk3hMDw/mO6nOObbXG3r4=) |
| [dbo.p\_cache\_publishid\_get](#1PcgqfvGzy3UkcWN7LQKIzudc+Y=) |
| [dbo.p\_canvasshotter\_book](#GBecoS5NNIUwnYY10IG98Y4ntws=) |
| [dbo.p\_canvasshotter\_booking\_history\_search](#94xtpURKbcm+rjwpqeLkew5J9lc=) |
| [dbo.p\_canvasshotter\_booking\_insert\_conditional](#5iHeTvzxxnH/YtLAt0wKOhWh8Ls=) |
| [dbo.p\_canvasshotter\_fallback](#w7g5EjCSqGzhgN064FP5YdNO6GM=) |
| [dbo.p\_canvasshotter\_history](#N8r5uy8daNeo8TkSqrrpdiwTXaU=) |
| [dbo.p\_canvasshotter\_unbook](#L/BBCL4V4hSpLE86zujRBjY+MW8=) |
| [dbo.p\_configuration\_get\_all](#EsJYab4euWPOJA6iZ0wTBFk/Gm8=) |
| [dbo.p\_configuration\_load](#4A8WPG9EXmbdkLYr/w11wlL7oSw=) |
| [dbo.p\_configuration\_set](#DeIdr/PSElcRt6bzhuI+IlR1rlU=) |
| [dbo.p\_get\_all\_template\_types](#QKiRgmfSorkRF8B3TsazZeJPlDI=) |
| [dbo.p\_insert\_event](#xN4F4KQcQp+aOyu0EvbP2rFSyLE=) |
| [dbo.p\_insert\_trace](#Xy+FZ4VbEdhnKPT68XTI1YCsOds=) |
| [dbo.p\_language\_get\_all](#X9O6KKdbCG9O6uoQedQsHiv8C/E=) |
| [dbo.p\_language\_insert](#IT7Hy/qMDQo++95jr0UXosszJys=) |
| [dbo.p\_language\_is\_id\_valid](#HBZCyP1MDpjLjZZ3HMhxiyNqAKA=) |
| [dbo.p\_newsletter\_subscriber\_insert](#34hs/68I0Kq5v/3iyTP3SkEzeXQ=) |
| [dbo.p\_online\_user\_book](#c5iw4Jj06eWGv8RAQS1V5t4MAv8=) |
| [dbo.p\_online\_user\_check\_connected](#TKm9+V+71Fc5TxTKUdeoAk8SXOw=) |
| [dbo.p\_online\_user\_delete](#TBplcYBoKypjbhjavP5K5dOlL1c=) |
| [dbo.p\_online\_user\_get\_by\_connectionid](#3hdddy7qEOm97dR0hNzxuENEE04=) |
| [dbo.p\_online\_user\_get\_by\_username](#LGVJsJlf1/SngrX4ixAa0Bh5h/s=) |
| [dbo.p\_online\_user\_get\_first\_by\_project\_id](#g/xQegeHFdihIcAOJ+yFPZuRE+o=) |
| [dbo.p\_online\_user\_get\_projectid](#NL9mm0bUGUv+swcXYJoRiYj6zW8=) |
| [dbo.p\_online\_user\_insert](#qv2u/8StiVLEVqLtKYLw5mU2W1s=) |
| [dbo.p\_online\_user\_search\_backend](#Djf8Cny5kYj9llD1X8X5SouWFZI=) |
| [dbo.p\_online\_user\_set\_disconnected](#6aV4Lug75izZ/3woIWF+ktDiCm4=) |
| [dbo.p\_online\_user\_unbook](#m8zhIuLkW2eQZExBU+OrYX9Q4P8=) |
| [dbo.p\_online\_user\_unbook\_forgot](#CR8dfDUDNElJfvYHsUNNh1QwKbk=) |
| [dbo.p\_online\_user\_update\_keepalive](#RGffZq+MHbZHzhLGgmYB7+RxXLQ=) |
| [dbo.p\_project\_copy\_book](#Xc2uZUUHsCqQzKVu6toSxfLcKL8=) |
| [dbo.p\_project\_copy\_fallback](#SZSN716jO+FTCfcstEifkqpd04k=) |
| [dbo.p\_project\_copy\_history](#iP9nGEt2cxagal93V/gZLHBLNeA=) |
| [dbo.p\_project\_copy\_history\_search](#SaV7osn7TKY5pGBFnNs0lyjeMxw=) |
| [dbo.p\_project\_copy\_insert](#IElOCneHQ59gypn3QPnkEKG6pSo=) |
| [dbo.p\_project\_copy\_unbook](#OIWAcYC3Mg2Um97d273ICxc0pwc=) |
| [dbo.p\_project\_delete\_template](#h6ZHMHXodjZQ7/r28dA/xoJe5Ic=) |
| [dbo.p\_project\_get\_by\_id](#KGz8phqiKp5J5UrsWlI+RqfLiH8=) |
| [dbo.p\_project\_get\_by\_userid](#f+k/6T5SoQt87MBYuQ+KyD5PD58=) |
| [dbo.p\_project\_get\_connectionid\_for\_presenter](#9ossGhPV6Hp0q1Wns3apYI72crQ=) |
| [dbo.p\_project\_get\_containeruri](#wHz01VXthrjWCG2k8L5EJx+O8+U=) |
| [dbo.p\_project\_get\_coparticipating\_projects](#D8T2UsmAddF87lP7WVAHYzS213o=) |
| [dbo.p\_project\_get\_coparticipating\_users](#Qw+YCRyiWNeg8ihfRXKeBlbOdso=) |
| [dbo.p\_project\_get\_licensed\_projects](#ceaCreTDTNvU0ICOTDx6kd9f8/Q=) |
| [dbo.p\_project\_get\_participating\_projects](#y6U32mvf6Ydonzjyz9Alrl4pM2g=) |
| [dbo.p\_project\_get\_participating\_projects\_count](#zIPBmWH0udwZ+u/O1fQlR71fyDk=) |
| [dbo.p\_project\_get\_templates](#Kx+UI3EG+DPsEGR/IXReULy0pqM=) |
| [dbo.p\_project\_insert](#DF2GFjYrkWYMFByvqV+nQcUlbKI=) |
| [dbo.p\_project\_participant\_delete](#qLGpKwd5+XtoBnbBN/kM9Pz0ZpI=) |
| [dbo.p\_project\_participant\_get\_by\_coparticipant](#JCJRheGtspnhunwKoYA/h8irV3I=) |
| [dbo.p\_project\_participant\_get\_by\_projectid](#E+DU3Nso4w1mgOq/VwtiREaUIrI=) |
| [dbo.p\_project\_participant\_get\_by\_projectid\_and\_userid](#kxWIu4dGjs334k0/sp7ExWmQERg=) |
| [dbo.p\_project\_participant\_get\_by\_userid](#Cvz0fhSf+WGHZxztYeigtKrc5qg=) |
| [dbo.p\_project\_participant\_insert](#P+CWh+/pskzvWIPXLquLWExDBaY=) |
| [dbo.p\_project\_participant\_update](#/Qukp870xZ16Zg9oI3zYOVxn2fw=) |
| [dbo.p\_project\_remove\_user\_as\_presenter\_in\_all](#eeHk7j2e7OVwoMFeLX3kgTe63Vk=) |
| [dbo.p\_project\_search\_backend](#xL4aGoOVsBGhh2gahaLWnst60v4=) |
| [dbo.p\_project\_set\_presenter](#gacANbG/B4xV95rDVlQFT5dl+wQ=) |
| [dbo.p\_project\_setlock](#XoZhwvaT1d/nS9CAEnKSYKp2IH4=) |
| [dbo.p\_project\_setversion](#gylQ5y5AlOemxenGqzVrVAeMMUc=) |
| [dbo.p\_project\_tag\_delete](#3z8R9RcW5Wjwzu5hfm2oJLVRbqc=) |
| [dbo.p\_project\_tag\_get\_by\_projectid](#3W7c4wEtBdaF/TMWjiRDP1vahwg=) |
| [dbo.p\_project\_tag\_get\_by\_projectids](#x3lNVjzNt+wEIiEXQvAI2D8+6QQ=) |
| [dbo.p\_project\_tag\_insert](#3vvsElx+lZXKCjcNsf3X5c+aQpY=) |
| [dbo.p\_project\_update](#HnARsWtIToI2HkIeehqGO8xLPYU=) |
| [dbo.p\_project\_update\_background](#tVCiNBHhV9nbMY5h705o5/e2aTY=) |
| [dbo.p\_project\_update\_canvas\_size\_ratio](#X9EZEEciQlSU00elRen/dNjE7SA=) |
| [dbo.p\_project\_update\_deleted](#nb5QQNpPcSK5w5nOBulGHodgG1w=) |
| [dbo.p\_project\_update\_owner](#OfB+9BtUWlve8RbDyeNuNPzX3yc=) |
| [dbo.p\_project\_update\_sas\_token](#RKyUApgnw8W6pXmnH6bJzqDSX5Y=) |
| [dbo.p\_project\_update\_template](#puatJhq48ml0PZR3M/9OUxn1CLY=) |
| [dbo.p\_project\_update\_thumbnail](#QUvLEBPm045EqTjq3BhBwvvubbY=) |
| [dbo.p\_project\_update\_zoom\_offset](#nYA8gp7ufhzJqRDzS9UMHz+um64=) |
| [dbo.p\_project\_upgrade\_book](#ZtIHMxzkIaQqTAum/wUN5RfUzwk=) |
| [dbo.p\_project\_upgrade\_fallback](#eyFMA0mMdoB7kA1BgRbWM2aeqpE=) |
| [dbo.p\_project\_upgrade\_history](#TRfRFOGkYYT9dyRx+1BlOHqMJnk=) |
| [dbo.p\_project\_upgrade\_insert](#2XDAj09j2hETAOl2Ohb7ZFRhJUU=) |
| [dbo.p\_project\_upgrade\_unbook](#9/vewVcF04vLmiCKTfJh/bhSZrU=) |
| [dbo.p\_projecttypes\_convert\_to\_template](#JwgwHPiObJe/EYScukAUlCcGImA=) |
| [dbo.p\_projecttypes\_delete](#70ejMUHyGW7VE9h4zbtOWktjRas=) |
| [dbo.p\_projecttypes\_get\_all\_template\_types](#Hqsb1aBo2hQhIDxbDm0pmRoxiog=) |
| [dbo.p\_projecttypes\_insert\_new\_projecttype](#6m8B+ayGYiWnCRtFav0XhiYLSOQ=) |
| [dbo.p\_projecttypes\_update](#r8wawLDNZrHLDtPZFcORekZDs4M=) |
| [dbo.p\_projecttypes\_verify\_if\_template](#10qW7rShvQlXAgL1Fh1vVBvRPVQ=) |
| [dbo.p\_quicklink\_delete](#hSJmCOcnSP/0EQyXqFjk0TGGCrU=) |
| [dbo.p\_quicklink\_get\_by\_id](#20MjjZnKch28Y9YLDxEDS3FGmTo=) |
| [dbo.p\_quicklink\_get\_by\_projectid](#CjFA3Ut9sPwMnJs7sFx+eBX6Pf8=) |
| [dbo.p\_quicklink\_get\_tiles\_for\_quicklink](#Cucj13pAuEzWnCUMIObApOpoAn0=) |
| [dbo.p\_quicklink\_insert](#7L+mI5DHWSS1FBBxmtHfmfPa4Mc=) |
| [dbo.p\_quicklink\_remove\_tile\_from\_quicklinks](#kIJK7kTX7Jb4WDa8+PQQ1AGu1T8=) |
| [dbo.p\_quicklink\_update](#dCF41uDCmFp5QJyNe7l1hO5erx8=) |
| [dbo.p\_storage\_eraser\_book](#y8l5XarsJbQbKA/O4NBf8iqG0nI=) |
| [dbo.p\_storage\_eraser\_unbook](#DJ35DDYZvLl9Ltne6CU+Guru1bk=) |
| [dbo.p\_telemetry\_insert](#K+D3VDXeBNU758yas8f/5w0zA+0=) |
| [dbo.p\_tile\_relation\_by\_participating\_tile](#4nTl9Mym0JSY27XlS6qwx+oO8Ww=) |
| [dbo.p\_tile\_relation\_delete](#qaxYLU/+tQD5a8bg+OzEjVFLvOA=) |
| [dbo.p\_tile\_relation\_delete\_all\_by\_tileid](#fos8gM3ClGfCxOrzgmfdOF2zYyw=) |
| [dbo.p\_tile\_relation\_get\_by\_projectid](#PjHeRYIUeUzVAfG8OgmofXHPZwY=) |
| [dbo.p\_tile\_relation\_get\_by\_projectid\_paged](#4tHkV/GK4O7uKFVRAnsUzM5KpJA=) |
| [dbo.p\_tile\_relation\_insert](#ATmtljARCqxJIPGy8/izryEv/y4=) |
| [dbo.p\_tile\_relation\_update](#OPU9oKDp8YMJ9WjYBAoaPsfOSRY=) |
| [dbo.p\_tile\_relation\_with\_tile](#TFrhkYoEzh9+cYTxXnhVxpekoBg=) |
| [dbo.p\_tile\_voting\_get\_by\_projectid\_and\_user\_id](#pM1zrS3UXx8O9xap0QOnGAXEIP0=) |
| [dbo.p\_tile\_voting\_get\_by\_projectid\_and\_user\_id\_paged](#uORfngnLeXLGViltwAI1sKjEVXs=) |
| [dbo.p\_tile\_voting\_update](#Oy2PprjsLacUoOpGiBn/tfpZFfQ=) |
| [dbo.p\_tilecopy\_cleanup](#B8DCD5XXKduW2LGzKEpheITcMD0=) |
| [dbo.p\_tilecopy\_getcopiedtiles\_by\_token](#GApIlX6Y9pIlbETFFMYFvJHAKTU=) |
| [dbo.p\_tilecopy\_getrelations\_by\_token](#tAWK7qE6vmmZvtMeEfSUlTEvQgU=) |
| [dbo.p\_tilecopy\_getsourcetiles\_by\_token](#cmT7FnIv1v7sCmmIynRK098uy4A=) |
| [dbo.p\_tilecopy\_insertcopiedtile](#HDv12qJtmkCLjWHumxpbZOTBz8g=) |
| [dbo.p\_tilecopy\_insertrelation](#rZoqGf5JuiqmeFlO6/jHRu4hBwg=) |
| [dbo.p\_tilecopy\_insertsourcetile](#r9P0d35uYsFf6SpqEej636wKAMM=) |
| [dbo.p\_tilestatus\_copy\_book](#kzIlhS2mJFiJma1P3FyuIngtmj4=) |
| [dbo.p\_tilestatus\_copy\_fallback](#FAozwoeeZq/SBEnuUGe8CMluFkI=) |
| [dbo.p\_tilestatus\_copy\_history](#jaRkA76y/rG7Lw8O/YMA4asdxIs=) |
| [dbo.p\_tilestatus\_copy\_history\_search](#h05wqjsaRO7IKn4gZbxDCQJeE5A=) |
| [dbo.p\_tilestatus\_copy\_insert](#dhfHL2WtimjmNW26GV2CNafaNn0=) |
| [dbo.p\_tilestatus\_copy\_unbook](#4S87Z+DjBYEuUD8UEzaRkx7S0CU=) |
| [dbo.p\_tilestatus\_delete](#g8XrmtIP1eDM2T1aBQtWCyFzoHA=) |
| [dbo.p\_tilestatus\_get\_by\_id](#XViQ2TSrI9Jh6ab5s0my1mZmNj4=) |
| [dbo.p\_tilestatus\_get\_by\_parenttile](#omKDRG2Umem5Xzmwq0PzSReapl0=) |
| [dbo.p\_tilestatus\_get\_by\_projectid](#vjVE9Go4aDkBtZQTZoJWex+rCTw=) |
| [dbo.p\_tilestatus\_get\_by\_projectid\_paged](#ZP9t21UM3jQkhPAjnryIGcmsaG0=) |
| [dbo.p\_tilestatus\_get\_by\_projectid\_with\_thumbnails](#MMa4x8Vn5ChNWEY98YLQDM4wcRk=) |
| [dbo.p\_tilestatus\_get\_max\_zindex](#8+MBm5Afux+OKPTv/cx5nXn2hyY=) |
| [dbo.p\_tilestatus\_get\_thumbnails\_for\_tile](#onMohemUZN4w8Ujofa/vIkPMLus=) |
| [dbo.p\_tilestatus\_get\_toplevel\_by\_projectid](#lBkwsZ7bE97Z9t0qayGlQ38X99w=) |
| [dbo.p\_tilestatus\_get\_upgradable\_for\_project](#eyljmpfDG1RdeUTtqr/BN0GN86U=) |
| [dbo.p\_tilestatus\_gettree\_by\_id](#6VlWIweS4UGBOSOX46QlUJpkZ+Y=) |
| [dbo.p\_tilestatus\_insert](#iMwd2Wua9ye9l0uBpCfK/FdQwyU=) |
| [dbo.p\_tilestatus\_stroke\_by\_tileid](#pF4LLqFVJBsIvqjSOGzm05jMQXk=) |
| [dbo.p\_tilestatus\_stroke\_insert](#plwNMHS8hQljZcchH6tJvNDattQ=) |
| [dbo.p\_tilestatus\_stroke\_update](#jTMbOmyXVLghL03C2dibHdt3+OY=) |
| [dbo.p\_tilestatus\_unlock\_by\_ids](#XXk93kY385SJHTBP+5O3Rfflchc=) |
| [dbo.p\_tilestatus\_unlock\_by\_lockedby](#FvFyeAP1oR9rBySsO9wdwnY33zk=) |
| [dbo.p\_tilestatus\_unlock\_by\_project\_lockedby](#4zv/71xy0wAzvROZMa4EIKlifsE=) |
| [dbo.p\_tilestatus\_unlock\_forgot](#8SPblYSof1I8LItXwQ2VeLi22IY=) |
| [dbo.p\_tilestatus\_update](#rEGjz6syz+nr2U/s/JA4/83VTuE=) |
| [dbo.p\_tilestatus\_update\_blobstatusid](#rKXiHuqjFMw5RE7XRuNRz7uwpBs=) |
| [dbo.p\_tilestatus\_update\_containertype](#EMnhgHE7sLgdHIlNA7PgnQ+Mmac=) |
| [dbo.p\_tilestatus\_update\_content](#ovOnybcjSuIEeGh3AadnsW3fAak=) |
| [dbo.p\_tilestatus\_update\_grouped](#jkNZ28XCQaKwniHN89cNfABNs90=) |
| [dbo.p\_tilestatus\_update\_lock](#aczDQAYDZF2HlDSzivZSrAuFtOc=) |
| [dbo.p\_tilestatus\_update\_pinning](#He9/XXIp9nHxusqmbIYpJZSVt+Y=) |
| [dbo.p\_tilestatus\_update\_position](#rVRoekoodnhEE3EEZ9teUKHN6pM=) |
| [dbo.p\_tilestatus\_update\_voting](#06H4YaAqEgHnlTMG8Tv9dNbrIj8=) |
| [dbo.p\_tilestatus\_update\_zindex](#Amby7pg+gmRtOz9BmB4nTr5EV6I=) |
| [dbo.p\_tilestatus\_upgrade\_book](#ld1zkb/fqpBa8wnM58UqebM1F4Q=) |
| [dbo.p\_tilestatus\_upgrade\_file](#w3C0inKZgUwc4TuTsp6vtlZZkKo=) |
| [dbo.p\_tilestatus\_upgrade\_insert](#cutcOfLrdt2GiPnsGtG+EKSAy/A=) |
| [dbo.p\_tilestatus\_upgrade\_unbook](#ULCY9m7aUkthzdBb6jkSOHYQKPQ=) |
| [dbo.p\_update\_blobstatus\_for\_thumbnails](#W9z9KJFncZneNc33VPLiUIevkYs=) |
| [dbo.p\_user\_get\_all](#8ThubK+YA9MeX3EaP6ZtEMifJXM=) |
| [dbo.p\_user\_get\_by\_project](#pLBDyYqwrvyK2QW29bKsZG0rbnY=) |
| [dbo.p\_user\_get\_by\_project\_count](#NPhxaTlIkusKkdZ6XY0kKAydQWs=) |
| [dbo.p\_user\_get\_by\_userid](#DRB0tZre192T3VYLVmPPCshxeo8=) |
| [dbo.p\_user\_get\_by\_username](#zxQu9m3+uzpNcizGDb+D2z/qJS0=) |
| [dbo.p\_user\_insert](#erG30Nokr2RkydueWcVzHMfNaT0=) |
| [dbo.p\_user\_update](#6JsegXsk48YP0Mqt/WgjOedKN6k=) |
| [licensing.p\_billing\_type\_delete](#vg2gXloGvUNaZu9BAPnJNB00Tg4=) |
| [licensing.p\_billing\_type\_get\_all](#PMj2YNV5pDGsp56mYX7pgjqA8xE=) |
| [licensing.p\_billing\_type\_get\_by\_id](#JaB461hdLg32uaON30BvUrc8ZP4=) |
| [licensing.p\_billing\_type\_insert](#+eFytUJ8juTRrqXTeInbRzxmdJM=) |
| [licensing.p\_billing\_type\_update](#7tWSQ8KOlGyZnElc0WeQMxzAEyA=) |
| [licensing.p\_customer\_delete](#stZpoeAkeKTIrN6NCVRIdVRFuXo=) |
| [licensing.p\_customer\_get\_all](#7+yidD4QgvrfFCpIM5ZsfL51iHw=) |
| [licensing.p\_customer\_get\_by\_id](#Raw4TRsrYeYr6m6E0PbCFwuH9/U=) |
| [licensing.p\_customer\_get\_by\_username](#74gegkjrVkNFJyF7nHYrL5KVB/w=) |
| [licensing.p\_customer\_insert](#m7i61yQ4jCHy5DZpBHm79Q7hR4k=) |
| [licensing.p\_customer\_search](#PsoeVlJWRfP8hEFv80X0T1aRrPE=) |
| [licensing.p\_customer\_update](#vXR0lcrCEpgSaMl3O6zdfbxDPxw=) |
| [licensing.p\_import\_data\_book](#Ek+cO1q0ERS+e3XoWT8BuLJr4PE=) |
| [licensing.p\_import\_data\_delete](#vcv96hVcYtP8SffqokjVQ18e5Lw=) |
| [licensing.p\_import\_data\_insert](#bGEdfgEGMtl6BJhsSOr7g0K5/PM=) |
| [licensing.p\_import\_data\_unbook](#uceApiKhjCnE0NqwYFf6UccE1P0=) |
| [licensing.p\_license\_file\_history\_get\_by\_id](#JVUDyYDrGRP/udh9nVtCVo/4CC8=) |
| [licensing.p\_license\_file\_history\_insert](#yZjMWthB2MAFeAZ578GrRsdaLs4=) |
| [licensing.p\_license\_file\_history\_search](#hoe6ni5kflfJih5y4nJFJbDSpMg=) |
| [licensing.p\_order\_delete](#EigYb0FdLvmqD5ujkVH/6Umwo+4=) |
| [licensing.p\_order\_get\_all](#A3Tu40WQ/ocSFG2NdWRdaIgZCd0=) |
| [licensing.p\_order\_get\_by\_id](#UN9WWgJIt/4Y7IL459GNob9qGzo=) |
| [licensing.p\_order\_insert](#1l0/zq9mmHbzt9IESJrFmx5WuzI=) |
| [licensing.p\_order\_search](#+Zd80IWktNnJI/CXVkqVbQsPW9g=) |
| [licensing.p\_order\_update](#xHsh9BAMTe+93MluRECHypDBqi0=) |
| [licensing.p\_orderline\_delete](#yo04ar1RAsMNv4QWyHfSjILXSmQ=) |
| [licensing.p\_orderline\_get\_all](#75r+ShR5FZAs2sh3QzjMImh1IqQ=) |
| [licensing.p\_orderline\_get\_by\_id](#pk7kG1J7ovRjefewu2BcFkb/l3Y=) |
| [licensing.p\_orderline\_get\_by\_orderid](#Q+kZj1w/IcBC1C1OqVHudyD9obc=) |
| [licensing.p\_orderline\_insert](#pgykHfrEgxmWtg6+e/sVd4bvQNk=) |
| [licensing.p\_orderline\_update](#u2XiJxBFrioPOznFtuO+R3VoAXM=) |
| [licensing.p\_payment\_delete](#sW3O29K9LPbWKG9+BnhbiiZ2iqA=) |
| [licensing.p\_payment\_get\_all](#hOJo6fZ7JQlYuXsKioti9JjAVG8=) |
| [licensing.p\_payment\_get\_by\_id](#6PP2XG1JNzZ8lZJGAKCT3LbMqyo=) |
| [licensing.p\_payment\_insert](#Vf3dZQR3PXwzJQjcb37g7d7VcGo=) |
| [licensing.p\_payment\_method\_delete](#ncnXnBCV/WC5YMn7pQfPGuxy1HQ=) |
| [licensing.p\_payment\_method\_get\_all](#5lECTxGpiPr6xsX5nmzEnrUolCQ=) |
| [licensing.p\_payment\_method\_get\_by\_id](#PwYn21M2sGlFfdw8Nhv5Lid7m3A=) |
| [licensing.p\_payment\_method\_insert](#UqnW41eMtTkBuO1DoFLjs53LrDc=) |
| [licensing.p\_payment\_method\_update](#YrXfOiz5Ce1PHt72DKgFyXDgbqM=) |
| [licensing.p\_payment\_search](#r80kFLotTafGBkTgvYp8OFtYnBw=) |
| [licensing.p\_payment\_update](#rnmdjVVAOml4ZiCi1LePwTGB8J8=) |
| [licensing.p\_price\_delete](#7yq5yyx/eC926x3R57o+1G26lIw=) |
| [licensing.p\_price\_get\_all](#jBbpnPWNgDMdhxWZpWcvZKE4uBI=) |
| [licensing.p\_price\_get\_by\_id](#mIiMl/2xxNxxuY8Mz9ODP3OYOcI=) |
| [licensing.p\_price\_get\_by\_pricelistid](#87jnmKQllPTudmDm4HhRauy9QL4=) |
| [licensing.p\_price\_insert](#GcvNDY3H3byKAlXsXPajIGFWTPw=) |
| [licensing.p\_price\_update](#hMy20Kl0XfPgaESO5O80Kfwd0ZA=) |
| [licensing.p\_pricelist\_delete](#jADKgqBjxt24G6rJVLeEmPYtDfQ=) |
| [licensing.p\_pricelist\_get\_all](#4K9Ybtdg6rkRn2sgKFqw/qpFKLo=) |
| [licensing.p\_pricelist\_get\_by\_id](#jGqwrvZPVIRpKSdkypF5RXzCxaA=) |
| [licensing.p\_pricelist\_insert](#6TvK++zjKQgqsucsSsf6hsMPnfw=) |
| [licensing.p\_pricelist\_update](#7D3GLHnCwfUBN0mooD6tWb3t+XA=) |
| [licensing.p\_product\_delete](#0yQlK4RXZoBMCgohU3Tr05y5dXo=) |
| [licensing.p\_product\_get\_all](#mG7He7twbIdITtsYe0mPN+FKtqw=) |
| [licensing.p\_product\_get\_by\_id](#AL9A0IZVpoVamMNplmKopBu/HC4=) |
| [licensing.p\_product\_get\_by\_productfamily](#d9koIjkTMqNAi6h4wOkRfAlwwFU=) |
| [licensing.p\_product\_insert](#aJLPQNgYOmXCXpsBy2Y4WHdv7qo=) |
| [licensing.p\_product\_update](#p2bEFY4v/8sVkniVLFF1OsQqGxk=) |
| [licensing.p\_promo\_code\_delete](#OpGkGZVyA4BSb9PHHq3pbA8qdsE=) |
| [licensing.p\_promo\_code\_get\_all](#jn/4m7TsEzpKslzo5MpTnaMKe3Y=) |
| [licensing.p\_promo\_code\_get\_by\_code](#wWw1FjA+m2gThQa53R9vugFSZyg=) |
| [licensing.p\_promo\_code\_get\_by\_id](#0iyFjkYSBBn/eH+0fdx/eGO9pP4=) |
| [licensing.p\_promo\_code\_get\_by\_promotypeid](#f+PJlqsB+iee2nH7TxRACZ7UK/U=) |
| [licensing.p\_promo\_code\_insert](#ag0TjjocRDKqpsRij/iGO0TN7Hk=) |
| [licensing.p\_promo\_code\_update](#6ZntmHL35uudvC/8cBW13cfg9m0=) |
| [licensing.p\_promo\_type\_delete](#/i+5YjG2+gcdsXmggZdDijyux30=) |
| [licensing.p\_promo\_type\_get\_all](#6nnCF+6G7J0uavIag4xaAqahcZI=) |
| [licensing.p\_promo\_type\_get\_by\_id](#fZiUoRLUXPGtFq2C1ERX1XM63GY=) |
| [licensing.p\_promo\_type\_insert](#zds++evrQWyZtzdVMN47ZykpFhU=) |
| [licensing.p\_promo\_type\_update](#JoQNEezfB7Ctq3QWDSEeDgqpeoE=) |
| [licensing.p\_subscription\_delete](#QFCZ6Ii7ZB93S4FdFze/UElWHzw=) |
| [licensing.p\_subscription\_device\_delete](#OvW9HlRBjCiNzot5nv3QfsXOMRk=) |
| [licensing.p\_subscription\_device\_get\_all](#ZNwcrfKOZjwN6gLDU2gsQShLg64=) |
| [licensing.p\_subscription\_device\_get\_by\_id](#J2Ixk+CPTdOyzdV2pdjCwYxFxSY=) |
| [licensing.p\_subscription\_device\_get\_by\_subscriptionid](#XJwL3gZ7pJ7ohOKLoOIDmmcQne8=) |
| [licensing.p\_subscription\_device\_get\_by\_subscriptionid\_devicecode](#TqbljfXL9mvriG0RqnwK7WwwR6g=) |
| [licensing.p\_subscription\_device\_insert](#kTCEvnUJK1U9T28z5paLXAjnCq0=) |
| [licensing.p\_subscription\_device\_update](#hfaMAggSWYiwBchRNl7IpyHeHCs=) |
| [licensing.p\_subscription\_get\_all](#SIrgPGxXvlV4FNipPBDsFdkzV5Y=) |
| [licensing.p\_subscription\_get\_by\_customerid](#VrF8ENqCAdEFAAWK45zZM2vNzbo=) |
| [licensing.p\_subscription\_get\_by\_customerid\_productfamilycode](#Eqcq8a+bqpwjB2XTA0kgOpQVImE=) |
| [licensing.p\_subscription\_get\_by\_id](#TPWwcHuwcnnusHTrosCTD6LMS/Q=) |
| [licensing.p\_subscription\_get\_by\_subscriptiondevice](#225pATZ7KewLBKmJFHmdFIAENo8=) |
| [licensing.p\_subscription\_get\_by\_subscriptionuser](#tTJad1cWPnrLdIvzBtd86vYpybA=) |
| [licensing.p\_subscription\_insert](#MMX7Y2afT72zjSWWeotCzrAh6Ik=) |
| [licensing.p\_subscription\_search](#lwhn7riTIvypHY0gkd39j/IiQ4w=) |
| [licensing.p\_subscription\_update](#Y1ehYjOfAxdvre7tC6sfEqYC/B8=) |
| [licensing.p\_subscription\_user\_delete](#PgAR8zhkf9scprZhWYeXceAht+4=) |
| [licensing.p\_subscription\_user\_get\_all](#ZwvfnamhlF8i4TI05qX7rsH8lFw=) |
| [licensing.p\_subscription\_user\_get\_by\_id](#uZ6svt0rbDgzpf5wg+dLE5i741Y=) |
| [licensing.p\_subscription\_user\_get\_by\_subscriptionid\_username](#+9OG++c2Z91qAZmkSNfUF+BLiaU=) |
| [licensing.p\_subscription\_user\_insert](#DDN48PXPjMZo8/enrDvN1VXcExs=) |
| [licensing.p\_subscription\_user\_search](#FtQXLxWS7Ltpw64vCnmLv2Hj89w=) |
| [licensing.p\_subscription\_user\_update](#+5YEV95dtYx+ZRRIxHbuRMji5/4=) |
| [mft.p\_check\_download\_available](#9OQhqSWAVHD+UBy5XRVnUeTSWWU=) |
| [mft.p\_check\_storageoperation\_finished](#YWckhcm8yx4WBJg+di4+9EJyP0Y=) |
| [mft.p\_check\_upload\_finished](#88BhiFm5uHwIoRoVhAIAiPjw0sw=) |
| [mft.p\_chunkingmachine\_book](#PnS53PCRBYBj/nxp1iARCISBVY8=) |
| [mft.p\_chunkingmachine\_completed](#GbqFV8ZfXcU29GDG9v8bxnOw5Us=) |
| [mft.p\_chunkingmachine\_insert](#f6LK3IajM3RnECpMSzPHWBs1WoU=) |
| [mft.p\_chunkingmachine\_unbook](#6o9FQNq5KnrXXUlc5B0tjGAOujg=) |
| [mft.p\_cleaningmachine\_bookchunking](#MwISX6HBX5uwfbo4eD1sPM5Ba+w=) |
| [mft.p\_cleaningmachine\_bookmergemoving](#lr7RkXHqmwkD5T1jKtSL+cyVZn8=) |
| [mft.p\_cleaningmachine\_bookmerging](#PcBMt0tIPqIsieGQG+6oFWBJhXQ=) |
| [mft.p\_cleaningmachine\_move\_chunkinghistory](#SFqDu+Wsq3Xfrp/t8dMSsrI3VAM=) |
| [mft.p\_cleaningmachine\_move\_mergehistory](#BZi0PPVjQ18TUJWBNm6Fb3uwbaI=) |
| [mft.p\_cleaningmachine\_unbookchunking](#ZSgOCGEe9avmVuDxDZqiS/jwcis=) |
| [mft.p\_cleaningmachine\_unbookmerging](#u6M/KQdFp+HD6diKs1auoaoNYME=) |
| [mft.p\_cleaningmachinenotcompleted\_bookchunking](#DQUu0qPhdSDXCubN8KtH3sGAOos=) |
| [mft.p\_cleaningmachinenotcompleted\_bookchunkingmoving](#BlF5NLkxVeX1k7IbVyiELdR7Yzc=) |
| [mft.p\_cleaningmachinenotcompleted\_bookmerging](#BGEQIhIkhDirIWgbHwqOmRwJtIk=) |
| [mft.p\_configuration\_load\_documentexplorer](#5lgRRNwkSMsNNynrmoi8cocasNM=) |
| [mft.p\_get\_activity\_from\_chunking](#qXRdjSXddfxXIHJ7Eym6hAXQJDM=) |
| [mft.p\_get\_activity\_from\_merge](#3yUdhmqUMo+5TUAe4NtlDssCtbs=) |
| [mft.p\_get\_activity\_from\_storage](#9e7ywF3uhRnjGcJZ4e9+VeUE340=) |
| [mft.p\_mergemachine\_book](#pNy6chimtMu1ga2KNNPvnYdyfzA=) |
| [mft.p\_mergemachine\_insert](#iUuoicTqGrjxPJNYyujMR6ek43U=) |
| [mft.p\_mergemachine\_ready\_to\_process](#Dh290Ssk5A7Vl7lE4ssUSvb0/To=) |
| [mft.p\_mergemachine\_unbook](#h9rJTw3uKsvsNflnarbRyuFacLM=) |
| [mft.p\_storagemachine\_book](#HjNI/SX3pG0VAK1u/2UQ3RjjiQc=) |
| [mft.p\_storagemachine\_insert](#WiPK2k7Zanqsa370TGyEzfu2RLc=) |
| [mft.p\_storagemachine\_unbook](#q4B3C6jmuIZfKqfaSkV2LMG4jSI=) |
| [mft.p\_storageoperations\_move\_history](#HzyPp0rH2W7p+J3+QqOKqUUnxuM=) |

|  |
| --- |
| [auth].[p\_user\_add\_role] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @rolename | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_add\_role]  @username nvarchar(250),  @rolename nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    INSERT INTO auth.t\_user\_role ([userid], [roleid])  SELECT TOP 1 t\_user.id as userid, t\_role.id as roleid  FROM auth.t\_user, auth.t\_role  WHERE t\_user.username = @username  AND t\_role.[name] = @rolename  set @rows = @@ROWCOUNT    SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_role]](#AweQXMLX6KvUKdT9zh9dn14odbs=)

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[auth]

|  |
| --- |
| [auth].[p\_user\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_delete]  @username nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    DELETE t\_user\_role  FROM auth.t\_user\_role  INNER JOIN auth.t\_user ON t\_user\_role.userid = t\_user.id  WHERE t\_user.username = @username  DELETE  FROM auth.t\_user  WHERE t\_user.username = @username  set @rows = @@ROWCOUNT    SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[auth]

|  |
| --- |
| [auth].[p\_user\_get\_by\_username] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @input\_username | nvarchar(250) | 500 |  |
| @id | uniqueidentifier | 16 | Out |
| @username | nvarchar(250) | 500 | Out |
| @firstname | nvarchar(250) | 500 | Out |
| @lastname | nvarchar(250) | 500 | Out |
| @email | nvarchar(250) | 500 | Out |
| @phonenumber | nvarchar(50) | 100 | Out |
| @language | nvarchar(50) | 100 | Out |
| @photourl | nvarchar(500) | 1000 | Out |
| @creationdate | datetime | 8 | Out |
| @lastlogindate | datetime | 8 | Out |
| @terms\_accepted | datetime | 8 | Out |
| @isverified | bit | 1 | Out |
| @lockout\_enabled | bit | 1 | Out |
| @lockout\_end\_date | datetime | 8 | Out |
| @authentication\_mode | int | 4 | Out |
| @otp\_security\_key | nvarchar(250) | 500 | Out |
| @external\_provider | nvarchar(50) | 100 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_get\_by\_username]  @input\_username NVARCHAR (250),  @id UNIQUEIDENTIFIER OUTPUT,  @username NVARCHAR (250) OUTPUT,  @firstname NVARCHAR (250) OUTPUT,  @lastname NVARCHAR (250) OUTPUT,  @email NVARCHAR (250) OUTPUT,  @phonenumber NVARCHAR (50) OUTPUT,  @language NVARCHAR (50) OUTPUT,  @photourl NVARCHAR (500) OUTPUT,  @creationdate DATETIME OUTPUT,  @lastlogindate DATETIME OUTPUT,  @terms\_accepted DATETIME OUTPUT,  @isverified BIT OUTPUT,  @lockout\_enabled BIT OUTPUT,  @lockout\_end\_date DATETIME OUTPUT,  @authentication\_mode INT OUTPUT,  @otp\_security\_key NVARCHAR (250) OUTPUT,  @external\_provider NVARCHAR (50) OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @id = id,  @username = username,  @firstname = firstname,  @lastname = lastname,  @email = email,  @phonenumber = phonenumber,  @language = [language],  @photourl = photourl,  @creationdate = creationdate,  @lastlogindate = lastlogindate,  @terms\_accepted = [terms\_accepted],  @isverified = isverified,  @lockout\_enabled = lockout\_enabled,  @lockout\_end\_date = lockout\_end\_date,  @authentication\_mode = authentication\_mode,  @otp\_security\_key = otp\_security\_key,  @external\_provider = external\_provider  FROM [auth].t\_user WITH (NOLOCK)  WHERE username = @input\_username  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

|  |
| --- |
| [auth].[p\_user\_get\_password] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @password­Hash | nvarchar(250) | 500 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_get\_password]  @username NVARCHAR (250),  @password­Hash NVARCHAR (250) OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @password­Hash = passwordhash  FROM [auth].t\_user WITH (NOLOCK)  WHERE username = @username  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

|  |
| --- |
| [auth].[p\_user\_get\_roles] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_get\_roles]  @username NVARCHAR (250)  AS BEGIN  SET NOCOUNT ON;  SELECT t\_role.[name] as rolename  FROM [auth].t\_role WITH (NOLOCK)  INNER JOIN [auth].t\_user\_role ON t\_role.id = t\_user\_role.roleid  INNER JOIN [auth].t\_user ON t\_user\_role.userid = t\_user.id  WHERE [auth].t\_user.username = @username  RETURN  END  GO |

Uses

[[auth].[t\_role]](#AweQXMLX6KvUKdT9zh9dn14odbs=)

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[auth]

|  |
| --- |
| [auth].[p\_user\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @firstname | nvarchar(250) | 500 |  |
| @lastname | nvarchar(250) | 500 |  |
| @email | nvarchar(250) | 500 |  |
| @phonenumber | nvarchar(50) | 100 |  |
| @language | nvarchar(50) | 100 |  |
| @photourl | nvarchar(500) | 1000 |  |
| @authentication\_mode | int | 4 |  |
| @external\_provider | nvarchar(50) | 100 |  |
| @terms\_accepted | datetime | 8 |  |
| @isverified | bit | 1 |  |
| @id | uniqueidentifier | 16 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_insert]  @username nvarchar(250),  @firstname nvarchar(250),  @lastname nvarchar(250),  @email nvarchar(250),  @phonenumber nvarchar(50),  @language nvarchar(50),  @photourl nvarchar(500),  @authentication\_mode int,  @external\_provider nvarchar(50),  @terms\_accepted datetime,  @isverified bit,  @id UNIQUEIDENTIFIER output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  SET @id = NEWID()  INSERT INTO [auth].t\_user  (  id,  username,  firstname,  lastname,  email,  phonenumber,  [language],  photourl,  authentication\_mode,  external\_provider,  terms\_accepted,  isverified,  creationdate  )  VALUES  (  @id,  @username,  @firstname,  @lastname,  @email,  @phonenumber,  @language,  @photourl,  @authentication\_mode,  @external\_provider,  @terms\_accepted,  @isverified,  GETUTCDATE()  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

|  |
| --- |
| [auth].[p\_user\_remove\_role] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @rolename | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_remove\_role]  @username nvarchar(250),  @rolename nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    DELETE t\_user\_role  FROM auth.t\_user\_role  INNER JOIN auth.t\_user ON t\_user\_role.userid = t\_user.id  INNER JOIN auth.t\_role ON t\_user\_role.roleid = t\_role.id  WHERE t\_user.username = @username  AND t\_role.[name] = @rolename  set @rows = @@ROWCOUNT    SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_role]](#AweQXMLX6KvUKdT9zh9dn14odbs=)

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[[auth].[t\_user\_role]](#z0oD8EiDELcVSPbOHQev9pusO10=)

[auth]

|  |
| --- |
| [auth].[p\_user\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | uniqueidentifier | 16 |  |
| @username | nvarchar(250) | 500 |  |
| @firstname | nvarchar(250) | 500 |  |
| @lastname | nvarchar(250) | 500 |  |
| @email | nvarchar(250) | 500 |  |
| @phonenumber | nvarchar(50) | 100 |  |
| @language | nvarchar(50) | 100 |  |
| @photourl | nvarchar(500) | 1000 |  |
| @lastlogindate | datetime | 8 |  |
| @terms\_accepted | datetime | 8 |  |
| @isverified | bit | 1 |  |
| @lockout\_enabled | bit | 1 |  |
| @lockout\_end\_date | datetime | 8 |  |
| @authentication\_mode | int | 4 |  |
| @otp\_security\_key | nvarchar(250) | 500 |  |
| @external\_provider | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [auth].[p\_user\_update]  @id UNIQUEIDENTIFIER,  @username nvarchar(250),  @firstname nvarchar(250),  @lastname nvarchar(250),  @email nvarchar(250),  @phonenumber nvarchar(50),  @language nvarchar(50),  @photourl nvarchar(500),  @lastlogindate datetime null,  @terms\_accepted datetime null,  @isverified bit,  @lockout\_enabled bit,  @lockout\_end\_date datetime,  @authentication\_mode int,  @otp\_security\_key nvarchar(250),  @external\_provider nvarchar(50),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    UPDATE [auth].t\_user  SET username = @username,  firstname = @firstname,  lastname = @lastname,  email = @email,  phonenumber = @phonenumber,  [language] = @language,  photourl = @photourl,  lastlogindate = @lastlogindate,  terms\_accepted = @terms\_accepted,  isverified = @isverified,  lockout\_enabled = @lockout\_enabled,  lockout\_end\_date = @lockout\_end\_date,  authentication\_mode = @authentication\_mode,  otp\_security\_key = @otp\_security\_key,  external\_provider = @external\_provider  WHERE id = @id  set @rows = @@ROWCOUNT    SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

|  |
| --- |
| [auth].[p\_user\_update\_password] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(250) | 500 |
| @passwordhash | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| /\*\*\*\*\*\* Object: Stored­Procedure [dbo].[p\_user\_update\_password] Script Date: 28/11/2019 4:52:06 μμ \*\*\*\*\*\*/  CREATE PROCEDURE [auth].[p\_user\_update\_password]  @username NVARCHAR (250),  @passwordhash NVARCHAR (250)  AS BEGIN  DECLARE @rows INT  SET NOCOUNT ON  UPDATE [auth].t\_user  SET passwordhash = @passwordhash  WHERE username = @username  SET @rows = @@ROWCOUNT    SET NOCOUNT OFF  RETURN (@rows - 1)  END  GO |

Uses

[[auth].[t\_user]](#DO0Gsi+pRv3WrPujaZIDUG5kUes=)

[auth]

|  |
| --- |
| [conversion].[p\_conversion\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_conversion\_book]  AS  BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) conversion.t\_conversion\_booking  SET bookedby = @booked\_by,  bookingdate = @booking\_date,  lastupdate = @booking\_date,  state­Machinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('working')  WHERE  (  state­Machinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('ready')  )  IF @@ROWCOUNT > 0  BEGIN  SELECT  [id],  [entityid],  [updatedby],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [domain],  [username],  [filename],  [destinationfolder],  [convertablefiletypeid],  [projectid],  [state­Machinestatusid],  [machineid]  FROM conversion.t\_conversion\_booking  WHERE bookedby=@booked\_by  END  SET NOCOUNT OFF  END  GO |

Uses

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_conversion\_book\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_conversion\_book\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE conversion.t\_conversion\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN conversion.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  conversion.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec    END  GO |

Uses

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_conversion\_book\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_conversion\_book\_history]  AS  BEGIN  INSERT [conversion].[t\_conversion\_booking\_history]  (  [id],  [entityid],  [updatedby],  [creationdate],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [domain],  [username],  [filename],  [destinationfolder],  [convertablefiletypeid],  [projectid],  [state­Machinestatusid],  [machineid]  )  SELECT  [id],  [entityid],  [updatedby],  [creationdate],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [domain],  [username],  [filename],  [destinationfolder],  [convertablefiletypeid],  [projectid],  [state­Machinestatusid],  [machineid]  FROM  (  DELETE [conversion].[t\_conversion\_booking]  OUTPUT  DELETED.[id],  DELETED.[entityid],  DELETED.[updatedby],  DELETED.[creationdate],  DELETED.[lastupdate],  DELETED.[bookedby],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[domain],  DELETED.[username],  DELETED.[filename],  DELETED.[destinationfolder],  DELETED.[convertablefiletypeid],  DELETED.[projectid],  DELETED.[state­Machinestatusid],  DELETED.[machineid]  WHERE [state­Machinestatusid] >= [conversion].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[[conversion].[t\_conversion\_booking\_history]](#Y0UXxUluDiZBREfu3Qg07kR0UW8=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_conversion\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @entityid | uniqueidentifier | 16 |
| @updatedby | nvarchar(200) | 400 |
| @lastupdate | datetime | 8 |
| @domain | nvarchar(200) | 400 |
| @username | nvarchar(200) | 400 |
| @filename | nvarchar(250) | 500 |
| @destinationfolder | nvarchar(250) | 500 |
| @convertablefiletypeid | int | 4 |
| @projectid | int | 4 |
| @machineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_conversion\_insert]  @entityid UNIQUEIDENTIFIER,  @updatedby NVARCHAR(200),  @lastupdate DATETIME,  @domain NVARCHAR(200),  @username NVARCHAR(200),  @filename NVARCHAR(250),  @destinationfolder NVARCHAR(250),  @convertablefiletypeid INT,  @projectid INT,  @machineid UNIQUEIDENTIFIER  AS  BEGIN  DECLARE @statemachinestatus INT  SET @statemachinestatus = [conversion].fn\_­Get­State­Machine­Status­Id('ready')  INSERT INTO [conversion].[t\_conversion\_booking]  (  [entityid],  [updatedby],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [domain],  [username],  [filename],  [destinationfolder],  [convertablefiletypeid],  [projectid],  [state­Machinestatusid],  [machineid]  )  VALUES(  @entityid,  @updatedby,  @lastupdate,  NULL, -- bookedby  NULL, -- boiking date  0, -- retry  @domain,  @username,  @filename,  @destinationfolder,  @convertablefiletypeid,  @projectid,  @statemachinestatus,  @machineid  )  RETURN @@IDENTITY  END  GO |

Uses

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_conversion\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_conversion\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()  UPDATE conversion.t\_conversion\_booking  SET  bookedby = NULL,  bookingdate = NULL,  retry = @retries,  lastupdate = @lastupdate,  state­Machinestatusid = @statemachinestatusid  WHERE  id = @id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[conversion].[t\_conversion\_booking]](#uTiDswOkhkFY12t13BlyECo1WqA=)

[conversion]

|  |
| --- |
| [conversion].[p\_image\_resizing\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_image\_resizing\_book]  AS  BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) conversion.t\_image\_resizing\_booking  SET bookedby=@booked\_by,  lastupdate = @booking\_date,  bookingdate = @booking\_date,  state­Machinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('working')  WHERE state­Machinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('ready')  IF @@ROWCOUNT > 0  BEGIN  SELECT  imageresizingbookingid,  tileid,  updatedby,  lastupdate,  file­Name,  bookedby,  bookingdate,  retry,  projectid,  state­Machinestatusid,  machineid  FROM conversion.t\_image\_resizing\_booking  WHERE bookedby=@booked\_by  END  SET NOCOUNT OFF  END  GO |

Uses

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_image\_resizing\_book\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_image\_resizing\_book\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE conversion.t\_image\_resizing\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN conversion.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  conversion.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = conversion.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec  END  GO |

Uses

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_image\_resizing\_book\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_image\_resizing\_book\_history]  AS  BEGIN  INSERT [conversion].[t\_image\_resizing\_booking\_history]  (  [imageresizingbookingid],  [tileid],  [updatedby],  [creationdate],  [lastupdate],  [file­Name],  [bookedby],  [bookingdate],  [retry],  [projectid],  [state­Machinestatusid],  [machineid]  )  SELECT  [imageresizingbookingid],  [tileid],  [updatedby],  [creationdate],  [lastupdate],  [file­Name],  [bookedby],  [bookingdate],  [retry],  [projectid],  [state­Machinestatusid],  [machineid]  FROM  (  DELETE [conversion].[t\_image\_resizing\_booking]  OUTPUT  DELETED.[imageresizingbookingid],  DELETED.[tileid],  DELETED.[updatedby],  DELETED.[creationdate],  DELETED.[lastupdate],  DELETED.[file­Name],  DELETED.[bookedby],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[projectid],  DELETED.[state­Machinestatusid],  DELETED.[machineid]  WHERE [state­Machinestatusid] >= [conversion].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

[[conversion].[t\_image\_resizing\_booking\_history]](#ZTGqNCVGVuaAWpCIqJJZokIlpuM=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_image\_resizing\_booking\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tile­Id | uniqueidentifier | 16 |
| @updated­By | nvarchar(200) | 400 |
| @project­Id | int | 4 |
| @file­Name | nvarchar(250) | 500 |
| @machine­Id | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_image\_resizing\_booking\_insert]  @tile­Id UNIQUEIDENTIFIER,  @updated­By NVARCHAR(200),  @project­Id INT,  @file­Name NVARCHAR(250),  @machine­Id UNIQUEIDENTIFIER  AS  BEGIN  SET NOCOUNT ON  DECLARE @last­Update DATETIME  SET @last­Update = GETUTCDATE()  INSERT INTO [conversion].[t\_image\_resizing\_booking]  (  [tileid],  [updatedby],  [lastupdate],  [projectid],  [file­Name],  [state­Machinestatusid],  [machineid]  )  VALUES  ( @tile­Id,  @updated­By,  @last­Update,  @project­Id,  @file­Name,  conversion.fn\_­Get­State­Machine­Status­Id('ready'),  @machine­Id  )  SET NOCOUNT OFF  END  GO |

Uses

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

[[conversion].[fn\_­Get­State­Machine­Status­Id]](#RsSpSE3LsfeWt/t7ng/gELcm9i0=)

[conversion]

|  |
| --- |
| [conversion].[p\_image\_resizing\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [conversion].[p\_image\_resizing\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()  UPDATE conversion.t\_image\_resizing\_booking  SET  bookedby = NULL,  bookingdate = NULL,  retry = @retries,  lastupdate = @lastupdate,  state­Machinestatusid = @statemachinestatusid  WHERE  imageresizingbookingid = @id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[conversion].[t\_image\_resizing\_booking]](#PFy0f4aYM/KVhD40Db4VdEF3FDk=)

[conversion]

|  |
| --- |
| [dbo].[p\_cache\_load] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @publishid | uniqueidentifier | 16 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_cache\_load]  @publishid uniqueidentifier output  AS  SET NOCOUNT ON;  SELECT [cacheid], [pollingtimeinseconds]  FROM t\_caches with (nolock)  SELECT @publishid = publishid  FROM t\_caches with (nolock)  WHERE cacheid = dbo.fn\_­Get­Cache­Id('cache')  RETURN  GO |

Uses

[[dbo].[t\_caches]](#n+xoB6o4S9rPTyl8cBF9APNvuGs=)

[[dbo].[fn\_­Get­Cache­Id]](#O/aCmRIgLeKi5tcqjHQHnuXfp00=)

|  |
| --- |
| [dbo].[p\_cache\_publishid\_get] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @cacheid | int | 4 |  |
| @publishid | uniqueidentifier | 16 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_cache\_publishid\_get]  @cacheid int,  @publishid uniqueidentifier output  AS  SELECT @publishid = publishid  FROM t\_caches with (nolock)  WHERE cacheid = @cacheid    RETURN 0  GO |

Uses

[[dbo].[t\_caches]](#n+xoB6o4S9rPTyl8cBF9APNvuGs=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_book]  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate smalldatetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(5) t\_canvasshotter\_booking  SET bookedby = @bookeby,  bookingdate = @bookingdate,  lastupdate = @bookingdate,  statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  WHERE statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY')  IF @@ROWCOUNT > 0  BEGIN  SELECT canvasshotterid, userid, projectid, lastupdate, bookedby, bookingdate, retry, statemachinestatusid  FROM t\_canvasshotter\_booking with(nolock)  WHERE bookedby = @bookeby  END  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_booking\_history\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_booking\_history\_search]  @projectid int,  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT h.[canvasshotterid],  h.[creationdate],  h.[userid],  h.[projectid],  h.[lastupdate],  h.[bookedby],  h.[bookingdate],  h.[retry],  h.[statemachinestatusid]  FROM [dbo].[t\_canvasshotter\_booking\_history] as h WITH (NOLOCK)  WHERE ((@projectid IS NULL) OR (h.projectid = @projectid))  ORDER BY [canvasshotterid]  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM [dbo].[t\_canvasshotter\_booking\_history] as h WITH (NOLOCK)  WHERE ((@projectid IS NULL) OR (h.projectid = @projectid))  END ELSE BEGIN  SET @totalrows = 0  END  END  GO |

Uses

[[dbo].[t\_canvasshotter\_booking\_history]](#AdXuTn3PyjKlMrK2nZ5poeksshY=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_booking\_insert\_conditional] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @projectid | int | 4 |  |
| @delta | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_booking\_insert\_conditional]  @username nvarchar(250),  @projectid int,  @delta int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  declare @userid int  set @userid = 0  SELECT @userid = userid  FROM t\_user WITH (NOLOCK)  WHERE username = @username  SET NOCOUNT ON  IF(@userid>0 AND NOT EXISTS (  SELECT 1  FROM t\_canvasshotter\_booking  WHERE projectid = @projectid  AND statemachinestatusid <= dbo.fn\_­Get­State­Machine­Status­Id('FINISHED')  AND DATEDIFF(millisecond, lastupdate, GETUTCDATE()) >= @delta ))  BEGIN  INSERT INTO t\_canvasshotter\_booking  (  userid,  projectid,  lastupdate  )  VALUES  (  @userid,  @projectid,  @lastupdate  )  SET @rows = @@ROWCOUNT  END      SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE dbo.t\_canvasshotter\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN dbo.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  dbo.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec    END  GO |

Uses

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_history]  AS  BEGIN  INSERT [dbo].[t\_canvasshotter\_booking\_history]  (  [canvasshotterid],  [creationdate],  [userid],  [projectid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid]  )  SELECT  [canvasshotterid],  [creationdate],  [userid],  [projectid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid]  FROM  (  DELETE [dbo].[t\_canvasshotter\_booking]  OUTPUT  DELETED.[canvasshotterid],  DELETED.[creationdate],  DELETED.[userid],  DELETED.[projectid],  DELETED.[lastupdate],  DELETED.[bookedby],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[statemachinestatusid]  WHERE [statemachinestatusid] >= [dbo].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

[[dbo].[t\_canvasshotter\_booking\_history]](#AdXuTn3PyjKlMrK2nZ5poeksshY=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_canvasshotter\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @canvasshotterid | int | 4 |  |
| @retry | int | 4 |  |
| @statemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_canvasshotter\_unbook]  @canvasshotterid int,  @retry int,  @statemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_canvasshotter\_booking  SET lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = @retry,  statemachinestatusid = @statemachinestatusid  WHERE canvasshotterid = @canvasshotterid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_canvasshotter\_booking]](#jU82GEddQUIqor8nUoFVYesFEEo=)

|  |
| --- |
| [dbo].[p\_configuration\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_configuration\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS  SET NOCOUNT ON;  SELECT [key], [value], [description], [App­Domain]  FROM t\_configuration with (nolock)  ORDER BY [key]  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_configuration with (nolock)  RETURN 0  RETURN  GO |

Uses

[[dbo].[t\_configuration]](#SlYHTPBSmXSvh/WEaW+L97YJRd8=)

|  |
| --- |
| [dbo].[p\_configuration\_load] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @appdomain | nvarchar(50) | 100 |  |
| @publishid | uniqueidentifier | 16 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_configuration\_load]  @appdomain nvarchar(50),  @publishid uniqueidentifier output  AS  SET NOCOUNT ON;  SELECT [key], [value]  FROM t\_configuration with (nolock)  WHERE [App­Domain] = @appdomain or [App­Domain] = '\*'  SELECT @publishid = publishid  FROM t\_caches with (nolock)  WHERE cacheid = dbo.fn\_­Get­Cache­Id('configuration')  RETURN  GO |

Uses

[[dbo].[t\_caches]](#n+xoB6o4S9rPTyl8cBF9APNvuGs=)

[[dbo].[t\_configuration]](#SlYHTPBSmXSvh/WEaW+L97YJRd8=)

[[dbo].[fn\_­Get­Cache­Id]](#O/aCmRIgLeKi5tcqjHQHnuXfp00=)

|  |
| --- |
| [dbo].[p\_configuration\_set] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @key | nvarchar(100) | 200 |  |
| @value | nvarchar(max) | max |  |
| @appdomain | nvarchar(50) | 100 |  |
| @description | nvarchar(250) | 500 |  |
| @publishid | uniqueidentifier | 16 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_configuration\_set]  @key nvarchar(100),  @value nvarchar(max),  @appdomain nvarchar(50),  @description nvarchar(250),  @publishid uniqueidentifier output  AS  SET NOCOUNT ON;    IF NOT EXISTS (SELECT NULL FROM t\_configuration where [key] = @key) BEGIN  INSERT INTO t\_configuration ([key], [value], [App­Domain], [description])  VALUES (@key, @value, @appdomain, @description)  END ELSE BEGIN  UPDATE t\_configuration  SET [value] = @value,  [App­Domain] = @appdomain,  [description] = @description  WHERE [key] = @key  END  SET @publishid = newid();  UPDATE t\_caches  SET publishid = @publishid  WHERE cacheid = dbo.fn\_­Get­Cache­Id('configuration')  SET NOCOUNT OFF  RETURN  GO |

Uses

[[dbo].[t\_caches]](#n+xoB6o4S9rPTyl8cBF9APNvuGs=)

[[dbo].[t\_configuration]](#SlYHTPBSmXSvh/WEaW+L97YJRd8=)

[[dbo].[fn\_­Get­Cache­Id]](#O/aCmRIgLeKi5tcqjHQHnuXfp00=)

|  |
| --- |
| [dbo].[p\_get\_all\_template\_types] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @language­Id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_get\_all\_template\_types]  @language­Id int = -1  AS  BEGIN  SET NOCOUNT ON;  SELECT projecttypeid,  description  FROM dbo.t\_project\_type  WHERE projecttypeid = 1  OR projecttypeid & 256 <> 0    END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

|  |
| --- |
| [dbo].[p\_insert\_event] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @message | nvarchar(max) | max |
| @params | nvarchar(max) | max |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_insert\_event]  @message nvarchar(MAX),  @params nvarchar(MAX)  AS  BEGIN  SET NOCOUNT ON;  insert into t\_events(Message,Params) values(@message, @params)  END  GO |

Uses

[[dbo].[t\_events]](#SAhKe3M7QyyIog5mlg0zn7hNQIg=)

|  |
| --- |
| [dbo].[p\_insert\_trace] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @message | nvarchar(max) | max |
| @severityelevel | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_insert\_trace]  @message nvarchar(MAX),  @severityelevel int  AS  BEGIN  SET NOCOUNT ON;  insert into t\_traces(Message,Severity­Level) values(@message, @severityelevel)  END  GO |

Uses

[[dbo].[t\_traces]](#KsOYXNDwJr+c7gsoWx5SwI5231k=)

|  |
| --- |
| [dbo].[p\_language\_get\_all] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_language\_get\_all]  AS BEGIN  SET NOCOUNT ON;  SELECT language­Id,  language­Code,  lcid,  name\_english,  name\_native  FROM dbo.t\_language WITH (nolock)  END  GO |

Uses

[[dbo].[t\_language]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=)

|  |
| --- |
| [dbo].[p\_language\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @language­Code | nvarchar(25) | 50 |  |
| @lcid | int | 4 |  |
| @name\_english | nvarchar(250) | 500 |  |
| @name\_native | nvarchar(250) | 500 |  |
| @language­Id | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_language\_insert]  @language­Code NVARCHAR(25),  @lcid INT,  @name\_english NVARCHAR(250),  @name\_native NVARCHAR(250),  @language­Id INT OUTPUT  AS BEGIN  INSERT INTO dbo.t\_language  (  language­Code,  lcid,  name\_english,  name\_native  )  VALUES  ( @language­Code, -- language­Code - nvarchar(25)  @lcid, -- lcid - int  @name\_english, -- name\_english - nvarchar(250)  @name\_native -- name\_native - nvarchar(250)  )  SELECT @language­Id = @@IDENTITY  END  GO |

Uses

[[dbo].[t\_language]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=)

|  |
| --- |
| [dbo].[p\_language\_is\_id\_valid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @languageid | int | 4 |  |
| @result | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_language\_is\_id\_valid]  @languageid INT,  @result BIT OUTPUT    AS BEGIN  SELECT @result =  CASE  WHEN EXISTS(SELECT \* FROM dbo.t\_language WHERE language­Id=@languageid)  THEN 1  ELSE 0  END  END  GO |

Uses

[[dbo].[t\_language]](#xEu9R9RGtGfNrJWbXHuK3R3cNpY=)

|  |
| --- |
| [dbo].[p\_newsletter\_subscriber\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @email | nvarchar(250) | 500 |  |
| @firstname | nvarchar(50) | 100 |  |
| @lastname | nvarchar(50) | 100 |  |
| @language | nvarchar(50) | 100 |  |
| @is\_active | bit | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_newsletter\_subscriber\_insert]  @email NVARCHAR(250),  @firstname NVARCHAR(50),  @lastname NVARCHAR(50),  @language NVARCHAR (50),  @is\_active BIT,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    INSERT INTO t\_newsletter\_subscriber  (  email,  firstname,  lastname,  [language],  is\_active,  creationdate  )  VALUES  (  @email,  @firstname,  @lastname,  @language,  @is\_active,  GETUTCDATE()  )  set @id = @@IDENTITY  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_newsletter\_subscriber]](#AWHNaP7PiJsuDf+PCHEERpzAUn4=)

|  |
| --- |
| [dbo].[p\_online\_user\_book] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @maxelapsedtimeinminutes | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_book]  @maxelapsedtimeinminutes INT  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate smalldatetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(5) t\_online\_user  SET bookedby = @bookeby,  bookingdate = @bookingdate,  lastupdate = @bookingdate,  statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  WHERE (statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY'))  AND @bookingdate > DATEADD(MINUTE, @maxelapsedtimeinminutes, lastupdate)  IF @@ROWCOUNT > 0  BEGIN  SELECT connectionid, machineid, userid, projectid, logindate, ispresenting, isdisconnected, lastupdate, bookedby, bookingdate, retry, statemachinestatusid  FROM t\_online\_user with(nolock)  WHERE bookedby = @bookeby  END  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_online\_user\_check\_connected] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @projectid | int | 4 |  |
| @isalsoconnected | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_check\_connected]  @userid INT,  @projectid INT,  @isalsoconnected BIT OUTPUT  AS  BEGIN  SET NOCOUNT ON  SELECT TOP(1)userid FROM dbo.t\_online\_user WITH (NOLOCK)  WHERE projectid=@projectid  AND userid = @userid  AND isdisconnected=0  ORDER BY userid  IF(@@ROWCOUNT = 0)  SET @isalsoconnected = 0  ELSE  SET @isalsoconnected = 1  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

|  |
| --- |
| [dbo].[p\_online\_user\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_delete]  @connectionid nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  UPDATE t\_tilestatus  SET lockeduser = null,  lockedby = null,  lockeddate = null,  lastupdate = @lastupdate  OUTPUT INSERTED.tileid  WHERE lockedby = @connectionid  SET NOCOUNT ON  --IF EXISTS (SELECT 1 FROM t\_online\_user WHERE connectionid = @connectionid)  -- BEGIN  DELETE  FROM t\_online\_user  WHERE connectionid = @connectionid  --END  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_online\_user\_get\_by\_connectionid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @username | nvarchar(250) | 500 | Out |
| @userid | int | 4 | Out |
| @firstname | nvarchar(50) | 100 | Out |
| @lastname | nvarchar(50) | 100 | Out |
| @photourl | nvarchar(250) | 500 | Out |
| @is\_guest | bit | 1 | Out |
| @projectid | int | 4 | Out |
| @ispresenting | bit | 1 | Out |
| @isdisconnected | bit | 1 | Out |
| @machineid | uniqueidentifier | 16 | Out |
| @lastupdate | datetime | 8 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_get\_by\_connectionid]  @connectionid NVARCHAR (50),  @username NVARCHAR (250) OUTPUT,  @userid INT = 0 OUTPUT,  @firstname NVARCHAR (50) OUTPUT,  @lastname NVARCHAR (50) OUTPUT,  @photourl NVARCHAR (250) OUTPUT,  @is\_guest BIT OUTPUT,  @projectid INT = 0 OUTPUT,  @ispresenting BIT OUTPUT,  @isdisconnected BIT OUTPUT,  @machineid UNIQUEIDENTIFIER OUTPUT,  @lastupdate DATETIME OUTPUT,  @rows INT = 0 OUTPUT    AS  SET NOCOUNT ON  SELECT @username = u.username,  @userid = u.userid,  @firstname = u.firstname,  @lastname = u.lastname,  @photourl = u.photourl,  @is\_guest = u.is\_guest,  @projectid = o.projectid,  @ispresenting = o.ispresenting,  @isdisconnected = o.isdisconnected,  @machineid = o.machineid,  @lastupdate = o.lastupdate  FROM t\_user as u WITH (NOLOCK)  LEFT JOIN t\_online\_user as o on o.userid = u.userid  WHERE connectionid = @connectionid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_online\_user\_get\_by\_username] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @user­Name | nvarchar(250) | 500 |  |
| @connection­Id | nvarchar(50) | 100 | Out |
| @machine­Id | uniqueidentifier | 16 | Out |
| @user­Id | int | 4 | Out |
| @project­Id | int | 4 | Out |
| @login­Date | datetime | 8 | Out |
| @ispresenting | bit | 1 | Out |
| @isdisconnected | bit | 1 | Out |
| @last­Update | datetime | 8 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_get\_by\_username]  @user­Name NVARCHAR(250),  @connection­Id NVARCHAR(50) OUTPUT,  @machine­Id UNIQUEIDENTIFIER OUTPUT,  @user­Id INT OUTPUT,  @project­Id INT OUTPUT,  @login­Date DATETIME OUTPUT,  @ispresenting BIT OUTPUT,  @isdisconnected BIT OUTPUT,  @last­Update DATETIME OUTPUT,  @rows INT OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT TOP(1)  @connection­Id = o.connectionid,  @machine­Id = o.machineid,  @user­Id = o.userid,  @project­Id = o.projectid,  @login­Date = o.logindate,  @ispresenting = o.ispresenting,  @isdisconnected = o.isdisconnected,  @last­Update = o.lastupdate  FROM t\_user AS u WITH (NOLOCK)  RIGHT JOIN t\_online\_user AS o ON o.userid = u.userid  WHERE u.username = @user­Name  ORDER BY o.lastupdate DESC  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_online\_user\_get\_first\_by\_project\_id] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @connectionid | nvarchar(50) | 100 | Out |
| @machineid | uniqueidentifier | 16 | Out |
| @userid | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_get\_first\_by\_project\_id]  @projectid INT,  @connectionid NVARCHAR (50) OUTPUT,  @machineid UNIQUEIDENTIFIER OUTPUT,  @userid INT OUTPUT,  @rows INT = 0 OUTPUT    AS  SET NOCOUNT ON  SELECT TOP(1)  @connectionid = connectionid,  @machineid = machineid,  @userid = userid  FROM t\_online\_user WITH (NOLOCK)  WHERE projectid = @projectid  ORDER BY lastupdate desc  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

|  |
| --- |
| [dbo].[p\_online\_user\_get\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_get\_projectid]  @projectid int  AS BEGIN  SET NOCOUNT ON  SELECT o.connectionid,  u.userid,  u.username,  u.firstname,  u.lastname,  u.photourl,  u.is\_guest,  u.creationdate,  u.lastactivitydate,  o.lastupdate,  o.ispresenting,  o.isdisconnected,  o.machineid  FROM t\_user as u WITH (NOLOCK)  RIGHT JOIN t\_online\_user as o on o.userid = u.userid  WHERE o.projectid = @projectid  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_online\_user\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @machineid | uniqueidentifier | 16 |  |
| @username | nvarchar(250) | 500 |  |
| @projectid | int | 4 |  |
| @ispresenting | bit | 1 |  |
| @isdisconnected | bit | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_insert]  @connectionid NVARCHAR (50),  @machineid uniqueidentifier,  @username nvarchar(250),  @projectid int,  @ispresenting BIT,  @isdisconnected BIT,  @rows int = 0 output  AS  declare @creationdate datetime  set @creationdate = GETUTCDATE()  declare @userid int  set @userid = 0  SELECT @userid = userid  FROM t\_user WITH (NOLOCK)  WHERE username = @username  SET NOCOUNT ON  IF(@userid>0)  BEGIN  BEGIN TRY  INSERT INTO t\_online\_user  (  connectionid,  machineid,  userid,  projectid,  ispresenting,  isdisconnected,  logindate,  lastupdate  )  VALUES  (  @connectionid,  @machineid,  @userid,  @projectid,  @ispresenting,  @isdisconnected,  @creationdate,  @creationdate  )  SET @rows = @@ROWCOUNT  END TRY  BEGIN CATCH  UPDATE t\_online\_user  SET machineid = @machineid,  userid = @userid,  projectid = @projectid,  logindate = @creationdate,  lastupdate = @creationdate  WHERE connectionid = @connectionid  SET @rows = @@ROWCOUNT  END CATCH  END      SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_online\_user\_search\_backend] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @username | nvarchar(250) | 500 |  |
| @is­Guest | bit | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_search\_backend]  @project­Id int,  @username nvarchar(250),  @is­Guest bit,  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT ou.connectionid,  ou.machineid,  ou.creationdate,  ou.userid,  ou.projectid,  ou.logindate,  ou.lastupdate,  ou.bookedby,  ou.bookingdate,  ou.retry,  ou.statemachinestatusid,  ou.ispresenting,  ou.isdisconnected,  p.[description] as project­Description,  u.username as username,  u.is\_guest as is\_guest  FROM t\_online\_user as ou WITH (NOLOCK)  LEFT JOIN t\_project as p ON ou.projectid = p.projectid  LEFT JOIN t\_user as u ON ou.userid = u.userid  WHERE ((@projectid IS NULL) OR (ou.projectid = @projectid))  AND ((@username IS NULL) OR (u.username = @username))  AND ((@is­Guest IS NULL) OR (u.is\_guest = @is­Guest))  ORDER BY [connectionid]  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_online\_user as ou WITH (NOLOCK)  LEFT JOIN t\_user as u ON ou.userid = u.userid  WHERE ((@projectid IS NULL) OR (ou.projectid = @projectid))  AND ((@username IS NULL) OR (u.username = @username))  AND ((@is­Guest IS NULL) OR (u.is\_guest = @is­Guest))  END ELSE BEGIN  SET @totalrows = 0  END  RETURN  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_online\_user\_set\_disconnected] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @hassucceeded | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_set\_disconnected]  @connectionid NVARCHAR(50),  @hassucceeded BIT OUTPUT  AS  BEGIN  SET NOCOUNT ON  UPDATE dbo.t\_online\_user  SET isdisconnected = 1  WHERE  (connectionid = @connectionid  AND isdisconnected = 0  )  IF(@@ROWCOUNT = 1)  SET @hassucceeded = 1  ELSE  SET @hassucceeded = 0  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

|  |
| --- |
| [dbo].[p\_online\_user\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @statemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_unbook]  @connectionid NVARCHAR (50),  @retry int,  @statemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_online\_user  SET lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = @retry,  statemachinestatusid = @statemachinestatusid  WHERE connectionid = @connectionid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

|  |
| --- |
| [dbo].[p\_online\_user\_unbook\_forgot] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_online\_user\_unbook\_forgot]  @rows int = 0 output  AS  declare @lastupdate smalldatetime  set @lastupdate = GETUTCDATE()  declare @bookingdate smalldatetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_online\_user  SET lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = retry+1,  statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY')  WHERE bookedby is not null  AND @bookingdate > DATEADD(MINUTE, 20, bookingdate)  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_online\_user\_update\_keepalive] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @connectionid | nvarchar(50) | 100 |  |
| @projectid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_online\_user\_update\_keepalive]  @connectionid NVARCHAR (50),  @projectid int,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_online\_user  SET projectid = @projectid,  lastupdate = GETUTCDATE(),  bookedby = null,  bookingdate = null,  retry = 0,  statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY')  WHERE connectionid = @connectionid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_copy\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_book]  AS BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) dbo.t\_project\_copy\_booking  SET bookedby=@booked\_by,  bookingdate = @booking\_date,  lastupdate = @booking\_date,  statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('working')  WHERE statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('ready')    IF @@ROWCOUNT > 0  BEGIN  SELECT Id,  projectid,  description,  isfromtemplate,  clientconnectionid,  username,  appver,  uniquedeviceid,  bookedby,  bookingdate,  retry,  statemachinestatusid,  xoffset,  yoffset,  templateid  FROM dbo.t\_project\_copy\_booking  WHERE bookedby=@booked\_by  END  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_copy\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE dbo.t\_project\_copy\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN dbo.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  dbo.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec    END  GO |

Uses

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_copy\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_history]  AS  BEGIN  INSERT [dbo].[t\_project\_copy\_booking\_history]  (  [Id],  [creationdate],  [projectid],  [description],  [isfromtemplate],  [clientconnectionid],  [username],  [appver],  [uniquedeviceid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid],  [xoffset],  [yoffset],  [templateid]  )  SELECT  [Id],  [creationdate],  [projectid],  [description],  [isfromtemplate],  [clientconnectionid],  [username],  [appver],  [uniquedeviceid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid],  [xoffset],  [yoffset],  [templateid]  FROM  (  DELETE [dbo].[t\_project\_copy\_booking]  OUTPUT  DELETED.[Id],  DELETED.[creationdate],  DELETED.[projectid],  DELETED.[description],  DELETED.[isfromtemplate],  DELETED.[clientconnectionid],  DELETED.[username],  DELETED.[appver],  DELETED.[uniquedeviceid],  DELETED.[lastupdate],  DELETED.[bookedby],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[statemachinestatusid],  DELETED.[xoffset],  DELETED.[yoffset],  DELETED.[templateid]  WHERE [statemachinestatusid] >= [dbo].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

[[dbo].[t\_project\_copy\_booking\_history]](#8yKLMTMt2d+nVkiLPu6HrcDuBEE=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_copy\_history\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_history\_search]  @projectid int,  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT [Id],  [creationdate],  [projectid],  [description],  [isfromtemplate],  [clientconnectionid],  [username],  [appver],  [uniquedeviceid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid],  [xoffset],  [yoffset],  [templateid]  FROM [dbo].[t\_project\_copy\_booking\_history] WITH (NOLOCK)  WHERE ((@projectid IS NULL) OR (projectid = @projectid))  ORDER BY [id]  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM [dbo].[t\_project\_copy\_booking\_history] WITH (NOLOCK)  WHERE ((@projectid IS NULL) OR (projectid = @projectid))  END ELSE BEGIN  SET @totalrows = 0  END  END  GO |

Uses

[[dbo].[t\_project\_copy\_booking\_history]](#8yKLMTMt2d+nVkiLPu6HrcDuBEE=)

|  |
| --- |
| [dbo].[p\_project\_copy\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @description | nvarchar(250) | 500 |
| @isfromtemplate | bit | 1 |
| @clientconnectionid | nvarchar(50) | 100 |
| @xoffset | float | 8 |
| @yoffset | float | 8 |
| @templateid | int | 4 |
| @username | nvarchar(200) | 400 |
| @appver | nvarchar(50) | 100 |
| @uniquedeviceid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_insert]  @projectid INT,  @description NVARCHAR(250),  @isfromtemplate BIT,  @clientconnectionid NVARCHAR(50) ,  @xoffset FLOAT,  @yoffset FLOAT,  @templateid INT,  @username NVARCHAR(200) ,  @appver NVARCHAR(50) ,  @uniquedeviceid UNIQUEIDENTIFIER  AS  BEGIN  DECLARE @statemachinestatusid INT  SET @statemachinestatusid = [dbo].[fn\_­Get­State­Machine­Status­Id]('ready')  INSERT INTO [dbo].[t\_project\_copy\_booking]  (  [projectid],  [description],  [isfromtemplate],  [clientconnectionid],  [username],  [appver],  [uniquedeviceid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid],  [xoffset],  [yoffset],  [templateid]  )  VALUES  (  @projectid,  @description,  @isfromtemplate,  @clientconnectionid,  @username,  @appver,  @uniquedeviceid,  GETUTCDATE(),  NULL, -- bookedby - uniqueidentifier  NULL, -- bookingdate - datetime  0, -- retry - int  @statemachinestatusid,  @xoffset,  @yoffset,  @templateid  )  RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_copy\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_copy\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  UPDATE dbo.t\_project\_copy\_booking  SET  bookedby=NULL,  bookingdate=NULL,  retry=@retries,  statemachinestatusid=@statemachinestatusid,  lastupdate = GETUTCDATE()  WHERE  Id=@id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project\_copy\_booking]](#xK10ySnNmS3vqciPJwVKT7p8JzQ=)

|  |
| --- |
| [dbo].[p\_project\_delete\_template] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_delete\_template]  @project­Id INT,  @rows INT OUTPUT  AS BEGIN  SET NOCOUNT ON  DELETE FROM dbo.t\_template\_translation  WHERE [projectid] = @project­Id  UPDATE dbo.t\_project  SET projecttypeid = 0  WHERE projectid = @project­Id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

|  |
| --- |
| [dbo].[p\_project\_get\_by\_id] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @description | nvarchar(250) | 500 | Out |
| @containeruri | nvarchar(500) | 1000 | Out |
| @projecttypeid | int | 4 | Out |
| @zoomfactor | float | 8 | Out |
| @offsetx | float | 8 | Out |
| @offsety | float | 8 | Out |
| @background | nvarchar(7) | 14 | Out |
| @createdby | int | 4 | Out |
| @createdbyuser | nvarchar(250) | 500 | Out |
| @createdbyuniquemachineid | uniqueidentifier | 16 | Out |
| @creationdate | datetime | 8 | Out |
| @updatedby | int | 4 | Out |
| @updatedbyuser | nvarchar(250) | 500 | Out |
| @lastupdate | datetime | 8 | Out |
| @presenter | nvarchar(250) | 500 | Out |
| @thumbnail | varbinary(max) | max | Out |
| @canvassizeratio | int | 4 | Out |
| @upgradeversion | int | 4 | Out |
| @islocked | bit | 1 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_by\_id]  @projectid INT,  @description NVARCHAR (250) OUTPUT,  @containeruri NVARCHAR (500) OUTPUT,  @projecttypeid INT OUTPUT,  @zoomfactor FLOAT OUTPUT,  @offsetx FLOAT OUTPUT,  @offsety FLOAT OUTPUT,  @background NVARCHAR (7) OUTPUT,  @createdby INT OUTPUT,  @createdbyuser NVARCHAR (250) OUTPUT,  @createdbyuniquemachineid UNIQUEIDENTIFIER OUTPUT,  @creationdate DATETIME OUTPUT,  @updatedby INT OUTPUT,  @updatedbyuser NVARCHAR (250) OUTPUT,  @lastupdate DATETIME OUTPUT,  @presenter NVARCHAR(250) OUTPUT,  @thumbnail VARBINARY (MAX) OUTPUT,  @canvassizeratio INT OUTPUT,  @upgradeversion INT OUTPUT,  @islocked BIT OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @description = p.[description],  @containeruri = p.containeruri,  @projecttypeid = p.projecttypeid,  @createdby = p.createdby,  @zoomfactor = p.zoomfactor,  @offsetx = p.offsetx,  @offsety = p.offsety,  @background = p.backgroundcolor,  @createdbyuniquemachineid = p.createdbyuniquemachineid,  @creationdate = p.creationdate,  @updatedby = p.updatedby,  @lastupdate = p.lastupdate,  @presenter = p.presenter,  @createdbyuser = created.username,  @updatedbyuser = updated.username,  @thumbnail = p.thumbnail,  @canvassizeratio = p.canvas\_size\_ratio,  @upgradeversion = p.upgradeversion,  @islocked = p.islocked  FROM t\_project as p WITH (NOLOCK)  left join t\_user as created on createdby = created.userid  left join t\_user as updated on updatedby = updated.userid  WHERE projectid = @projectid  AND deleted = 0  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_get\_by\_userid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_by\_userid]  @userid int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT p.projectid, p.[description], p.backgroundcolor,  p.containeruri, p.projecttypeid, p.thumbnail, p.backgroundcolor, p.createdby,  p.createdbyuniquemachineid, p.creationdate, p.updatedby,  p.lastupdate, p.presenter, p.upgradeversion, p.islocked, created.username as createdbyuser, updated.username as updatedbyuser  FROM t\_project as p WITH (NOLOCK)  left join t\_user as created on createdby = created.userid  left join t\_user as updated on updatedby = updated.userid  WHERE createdby = @userid  AND deleted = 0  ORDER BY projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_project with (nolock)  WHERE createdby = @userid  AND deleted = 0  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_get\_connectionid\_for\_presenter] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @presenter­Connection­Id | nvarchar(50) | 100 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_connectionid\_for\_presenter]  @project­Id INT,  @presenter­Connection­Id NVARCHAR(50) OUTPUT  AS  BEGIN  SELECT  @presenter­Connection­Id = ou.connectionid  FROM dbo.t\_online\_user ou  LEFT JOIN dbo.t\_project p  ON  p.projectid = ou.projectid  WHERE  p.presenter IS NOT NULL  AND  p.projectid = @project­Id  AND  ou.isdisconnected = 0  IF @@ROWCOUNT = 0  SET @presenter­Connection­Id = NULL  END  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_get\_containeruri] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @container­Uri | nvarchar(500) | 1000 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_containeruri]  @project­Id INT,  @container­Uri NVARCHAR(500) OUTPUT  AS  BEGIN  SELECT TOP(1) @container­Uri = containeruri  FROM dbo.t\_project  WHERE projectid = @project­Id  ORDER BY projectid  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_get\_coparticipating\_projects] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @coparticipantid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_coparticipating\_projects]  @userid int,  @coparticipantid int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT p.projectid, p.[description], p.backgroundcolor,  p.containeruri, p.projecttypeid, p.thumbnail, p.createdby,  p.createdbyuniquemachineid, p.creationdate, p.updatedby,  p.lastupdate, p.presenter, p.upgradeversion, p.islocked, created.username as createdbyuser, updated.username as updatedbyuser  FROM t\_project as p WITH (NOLOCK)  LEFT JOIN t\_user as created WITH (NOLOCK) on createdby = created.userid  LEFT JOIN t\_user as updated WITH (NOLOCK) on updatedby = updated.userid  WHERE EXISTS (  SELECT null  FROM t\_project\_participant as pp­A WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp­B WITH (NOLOCK) on pp­A.projectid = pp­B.projectid  WHERE pp­A.projectid = p.projectid  AND pp­A.userid = @userid  AND pp­B.userid = @coparticipantid  )  AND p.deleted = 0  ORDER BY projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_project\_participant as pp­A WITH (NOLOCK)  LEFT JOIN t\_project\_participant as pp­B WITH (NOLOCK) on pp­A.projectid = pp­B.projectid  LEFT JOIN t\_project as p WITH (NOLOCK) on pp­A.projectid = p.projectid  WHERE pp­A.userid = @userid  AND pp­B.userid = @coparticipantid  AND p.deleted = 0  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_get\_coparticipating\_users] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @is\_guest | bit | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_coparticipating\_users]  @userid int,  @is\_guest bit,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT u.userid, u.username, u.firstname, u.lastname, u.photourl, u.is\_guest, u.creationdate, u.lastupdate, u.lastactivitydate  FROM t\_user as u WITH (NOLOCK)  WHERE EXISTS (  SELECT null  FROM t\_project\_participant as pp­A WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp­B WITH (NOLOCK) on pp­A.projectid = pp­B.projectid  INNER JOIN t\_project as p WITH (NOLOCK) on pp­A.projectid = p.projectid  WHERE pp­A.userid = u.userid  AND pp­B.userid = @userid  AND pp­A.userid != @userid  AND p.deleted = 0  )  AND ((@is\_guest is null) or (u.is\_guest = @is\_guest))  ORDER BY u.userid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_user as u WITH (NOLOCK)  WHERE EXISTS (  SELECT null  FROM t\_project\_participant as pp­A WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp­B WITH (NOLOCK) on pp­A.projectid = pp­B.projectid  INNER JOIN t\_project as p WITH (NOLOCK) on pp­A.projectid = p.projectid  WHERE pp­A.userid = u.userid  AND pp­B.userid = @userid  AND pp­A.userid != @userid  AND p.deleted = 0  )  AND ((@is\_guest is null) or (u.is\_guest = @is\_guest))  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_get\_licensed\_projects] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @allowed­Number­Of­Projects | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_licensed\_projects]  @userid int,  @allowed­Number­Of­Projects int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT \*  FROM (  SELECT p.projectid, p.[description], p.backgroundcolor,  p.containeruri, p.projecttypeid, p.thumbnail, p.createdby,  p.createdbyuniquemachineid, p.creationdate, p.updatedby, p.lastupdate, p.upgradeversion, p.islocked,pp.lastaccessdate  FROM t\_project as p WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp WITH (NOLOCK) on p.projectid = pp.projectid  WHERE pp.userid = @userid  AND p.deleted = 0  AND pp.permission = 255  ORDER BY pp.lastaccessdate desc, p.projectid desc  OFFSET 0 ROWS  FETCH NEXT @allowed­Number­Of­Projects ROWS ONLY  ) T0  ORDER BY T0.lastaccessdate desc, T0.projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(\*)  FROM (  SELECT p.projectid  FROM t\_project as p WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp WITH (NOLOCK) on p.projectid = pp.projectid  WHERE pp.userid = @userid  AND p.deleted = 0  AND pp.permission = 255  ORDER BY pp.lastaccessdate desc, p.projectid desc  OFFSET 0 ROWS  FETCH NEXT @allowed­Number­Of­Projects ROWS ONLY  ) T0  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_get\_participating\_projects] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @permission | tinyint | 1 |  |
| @description­Filter | nvarchar(500) | 1000 |  |
| @tag­Filter | nvarchar(500) | 1000 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @sort­Column | nvarchar(100) | 200 |  |
| @sort­Direction | nvarchar(4) | 8 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_participating\_projects]  @userid int,  @permission tinyint,  @description­Filter nvarchar(500),  @tag­Filter nvarchar(500),  @pagesize int,  @pagenumber int,  @sort­Column nvarchar(100),  @sort­Direction nvarchar(4),  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT p.projectid, p.[description], p.backgroundcolor,  p.containeruri, p.projecttypeid, p.thumbnail, p.createdby,  p.createdbyuniquemachineid, p.creationdate, p.updatedby,  p.lastupdate, p.presenter, p.upgradeversion, p.islocked, created.username as createdbyuser, updated.username as updatedbyuser,  own.userid as owner\_userid, own.username as owner\_username, own.firstname as owner\_firstname, own.lastname as owner\_lastname, own.photourl as owner\_photourl  FROM t\_project as p WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp WITH (NOLOCK) on p.projectid = pp.projectid  LEFT JOIN t\_user as created on createdby = created.userid  LEFT JOIN t\_user as updated on updatedby = updated.userid  OUTER APPLY (  SELECT TOP 1 t\_user.userid, t\_user.username, t\_user.firstname, t\_user.lastname, t\_user.photourl  FROM t\_user  INNER JOIN t\_project\_participant ON t\_project\_participant.userid = t\_user.userid  WHERE t\_project\_participant.projectid = p.projectid and t\_project\_participant.permission = 255  ORDER BY t\_user.userid asc  ) own  WHERE pp.userid = @userid  AND p.deleted = 0  AND ((@permission is null) or (pp.permission = @permission))  AND ((@description­Filter is null) or (p.[description] like @description­Filter))  AND ((@tag­Filter is null) or EXISTS (  SELECT NULL FROM t\_project\_tag WITH (NOLOCK)  WHERE p.projectid = t\_project\_tag.projectid  AND t\_project\_tag.tag like @tag­Filter  ))  ORDER BY  CASE WHEN @sort­Direction = 'ASC' THEN  CASE @sort­Column  WHEN 'Project­Name' THEN p.[description]  WHEN 'Project­Owner' THEN CONCAT(own.[firstname], own.[lastname])  WHEN 'Project­Last­Access­Date' THEN CONVERT(CHAR(23), pp.lastaccessdate, 121)  END  END,  CASE WHEN @sort­Direction = 'DESC' THEN  CASE @sort­Column  WHEN 'Project­Name' THEN p.[description]  WHEN 'Project­Owner' THEN CONCAT(own.[firstname], own.[lastname])  WHEN 'Project­Last­Access­Date' THEN CONVERT(CHAR(23), pp.lastaccessdate, 121)  END  END DESC,  p.projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_project as p WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp with (NOLOCK) on p.projectid = pp.projectid  WHERE pp.userid = @userid  AND p.deleted = 0  AND ((@permission is null) or (pp.permission = @permission))  AND ((@description­Filter is null) or (p.[description] like @description­Filter))  AND ((@tag­Filter is null) or EXISTS (  SELECT NULL FROM t\_project\_tag WITH (NOLOCK)  WHERE p.projectid = t\_project\_tag.projectid  AND t\_project\_tag.tag like @tag­Filter  ))  END ELSE BEGIN  SET @totalrows = 0  END  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_get\_participating\_projects\_count] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @permission | tinyint | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_participating\_projects\_count]  @userid int,  @permission tinyint,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT @totalrows = count(\*)  FROM t\_project as p WITH (NOLOCK)  INNER JOIN t\_project\_participant as pp WITH (NOLOCK) on p.projectid = pp.projectid  WHERE pp.userid = @userid  AND ((@permission is null) or (pp.permission = @permission))  AND p.deleted = 0  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_get\_templates] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @language­Id | int | 4 |  |
| @page­Size | int | 4 |  |
| @page­Number | int | 4 |  |
| @category­Id | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_get\_templates]  @language­Id INT,  @page­Size INT,  @page­Number INT,  @category­Id INT,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT p.projectid,  p.lastupdate,  p.projecttypeid,  p.thumbnail,  t.languageid,  t.[description],  t.extendeddescription,  t.moreinfourl  FROM dbo.t\_project p WITH (NOLOCK)  LEFT JOIN dbo.t\_template\_translation t WITH (NOLOCK) ON t.projectid = p.projectid  WHERE p.deleted = 0  AND (  ((@category­Id > 0) AND (p.projecttypeid = @category­Id))  OR ((@category­Id = -1) AND ((p.projecttypeid = 1) OR (p.projecttypeid & 256 > 0)))  )  AND ((t.languageid = @language­Id) OR (t.languageid=1))  ORDER BY t.projectid, t.languageid  OFFSET @page­Size \* (@page­Number - 1) ROWS  FETCH NEXT @page­Size ROWS ONLY;  SELECT @totalrows = count(1)  FROM dbo.t\_project p WITH (NOLOCK)  LEFT JOIN dbo.t\_template\_translation t WITH (NOLOCK) ON t.projectid = p.projectid  WHERE p.deleted = 0  AND (  ((@category­Id > 0) AND (p.projecttypeid = @category­Id))  OR ((@category­Id = -1) AND ((p.projecttypeid = 1) OR (p.projecttypeid & 256 > 0)))  )  AND ((t.languageid = @language­Id) OR (t.languageid=1))  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

|  |
| --- |
| [dbo].[p\_project\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @description | nvarchar(250) | 500 |  |
| @backgroundcolor | nvarchar(7) | 14 |  |
| @containeruri | nvarchar(500) | 1000 |  |
| @createdby | int | 4 |  |
| @createdbyuniquemachineid | uniqueidentifier | 16 |  |
| @upgradeversion | int | 4 |  |
| @projectid | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_insert]  @description nvarchar(250),  @backgroundcolor nvarchar(7),  @containeruri nvarchar (500),  @createdby int,  @createdbyuniquemachineid uniqueidentifier,  @upgradeversion INT,  @projectid int output,  @rows int = 0 output  AS  declare @creationdate datetime  set @creationdate = GETUTCDATE()  SET NOCOUNT ON  INSERT INTO t\_project  (  [description],  [backgroundcolor],  [containeruri],  [createdby],  [createdbyuniquemachineid],  [creationdate],  [lastupdate],  [upgradeversion]  )  --OUTPUT inserted.projectid  VALUES  (  @description,  @backgroundcolor,  @containeruri,  @createdby,  @createdbyuniquemachineid,  @creationdate,  @creationdate,  @upgradeversion  )  set @projectid = @@IDENTITY  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_participant\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_delete]  @projectid int,  @userid int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    DELETE  FROM t\_project\_participant  WHERE projectid = @projectid  AND userid = @userid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_get\_by\_coparticipant] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_get\_by\_coparticipant]  @userid int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT A.projectid, A.userid, A.permission, A.lastaccessdate  FROM t\_project\_participant A  WHERE EXISTS (  SELECT null  FROM t\_project\_participant B  WHERE A.projectid = B.projectid  AND B.userid = @userid  )    SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_get\_by\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @permission | tinyint | 1 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_get\_by\_projectid]  @projectid int,  @permission tinyint  AS BEGIN  SET NOCOUNT ON  SELECT pp.projectid, pp.userid, pp.permission, pp.lastaccessdate  FROM t\_project\_participant as pp WITH (NOLOCK)  WHERE pp.projectid = @projectid  AND ((@permission is null) or (pp.permission = @permission))  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_get\_by\_projectid\_and\_userid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @permission | tinyint | 1 | Out |
| @lastaccessdate | datetime | 8 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_get\_by\_projectid\_and\_userid]  @projectid INT,  @userid INT,  @permission tinyint = 0 OUTPUT,  @lastaccessdate datetime OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @permission = pp.permission,  @lastaccessdate = pp.lastaccessdate  FROM t\_project\_participant as pp WITH (NOLOCK)  WHERE pp.projectid = @projectid  AND pp.userid = @userid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_get\_by\_userid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @userid | int | 4 |
| @permission | tinyint | 1 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_get\_by\_userid]  @userid int,  @permission tinyint  AS BEGIN  SET NOCOUNT ON  SELECT pp.projectid, pp.userid, pp.permission, pp.lastaccessdate  FROM t\_project\_participant as pp WITH (NOLOCK)  WHERE pp.userid = @userid  AND ((@permission is null) or (pp.permission = @permission))  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @userid | int | 4 |
| @permission | tinyint | 1 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_insert]  @projectid int,  @userid int,  @permission tinyint  AS BEGIN  SET NOCOUNT ON  INSERT INTO t\_project\_participant  (  projectid,  userid,  permission,  lastaccessdate  )  VALUES  (  @projectid,  @userid,  @permission,  GETUTCDATE()  )  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_participant\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @permission | tinyint | 1 |  |
| @lastaccessdate | datetime | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_participant\_update]  @projectid int,  @userid int,  @permission tinyint,  @lastaccessdate datetime,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE t\_project\_participant  SET permission = @permission,  lastaccessdate = @lastaccessdate  WHERE projectid = @projectid  AND userid = @userid    set @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_remove\_user\_as\_presenter\_in\_all] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @user­Name | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_remove\_user\_as\_presenter\_in\_all]  @user­Name NVARCHAR(250)  AS  BEGIN  SELECT \* INTO #temp FROM (  SELECT projectid  FROM dbo.t\_project  WHERE LOWER(presenter) = LOWER(@user­Name)  )  AS all­Ids  UPDATE dbo.t\_project  SET presenter = NULL  WHERE projectid IN (SELECT projectid FROM #temp)  SELECT projectid FROM #temp  DROP TABLE #temp    END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_search\_backend] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @participant­Id | int | 4 |  |
| @participant­Permission | tinyint | 1 |  |
| @description­Filter | nvarchar(500) | 1000 |  |
| @is­Deleted | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_search\_backend]  @project­Id int,  @participant­Id int,  @participant­Permission tinyint,  @description­Filter nvarchar(500),  @is­Deleted int,  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT p.projectid, p.[description], p.backgroundcolor,  p.containeruri, p.projecttypeid, p.thumbnail, p.createdby,  p.createdbyuniquemachineid, p.creationdate, p.updatedby,  p.lastupdate, p.presenter,p.upgradeversion, p.islocked, p.deleted,  created.username as createdbyuser, updated.username as updatedbyuser  FROM t\_project as p WITH (NOLOCK)  LEFT JOIN t\_user as created on createdby = created.userid  LEFT JOIN t\_user as updated on updatedby = updated.userid  WHERE ((@project­Id is null) or (p.projectid = @project­Id))  AND ((@participant­Id is null) OR EXISTS (  SELECT NULL FROM t\_project\_participant as pp WITH (NOLOCK)  WHERE pp.projectid = p.projectid  AND pp.userid = @participant­Id  AND ((@participant­Permission is null) OR (pp.permission = @participant­Permission))  ))  AND ((@is­Deleted is null) OR (p.deleted = @is­Deleted))  AND ((@description­Filter is null) OR (p.[description] like @description­Filter))  ORDER BY p.projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_project as p WITH (NOLOCK)  WHERE ((@project­Id is null) or (p.projectid = @project­Id))  AND ((@participant­Id is null) OR EXISTS (  SELECT NULL FROM t\_project\_participant as pp WITH (NOLOCK)  WHERE pp.projectid = p.projectid  AND pp.userid = @participant­Id  AND ((@participant­Permission is null) OR (pp.permission = @participant­Permission))  ))  AND ((@is­Deleted is null) OR (p.deleted = @is­Deleted))  AND ((@description­Filter is null) OR (p.[description] like @description­Filter))  END ELSE BEGIN  SET @totalrows = 0  END  RETURN  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_project\_set\_presenter] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @project­Id | int | 4 |
| @presenter | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_set\_presenter]  @project­Id INT NULL,  @presenter NVARCHAR(250) NULL  AS  BEGIN  UPDATE dbo.t\_project  SET presenter = @presenter  WHERE projectid = @project­Id  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_setlock] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @islocked | bit | 1 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_setlock]  @projectid int,  @islocked bit  AS  UPDATE [dbo].[t\_project]  SET islocked = @islocked  WHERE projectid = @projectid  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_setversion] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @upgradeversion | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_setversion]  @projectid int,  @upgradeversion int  AS  UPDATE [dbo].[t\_project]  SET upgradeversion = @upgradeversion  WHERE projectid = @projectid  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_tag\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @tag | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_tag\_delete]  @projectid int,  @tag nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    DELETE  FROM t\_project\_tag  WHERE projectid = @projectid  AND tag = @tag  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

|  |
| --- |
| [dbo].[p\_project\_tag\_get\_by\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_tag\_get\_by\_projectid]  @projectid int  AS BEGIN  SET NOCOUNT ON  SELECT tag  FROM t\_project\_tag WITH (NOLOCK)  WHERE projectid = @projectid  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

|  |
| --- |
| [dbo].[p\_project\_tag\_get\_by\_projectids] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @p0 | int | 4 |
| @p1 | int | 4 |
| @p2 | int | 4 |
| @p3 | int | 4 |
| @p4 | int | 4 |
| @p5 | int | 4 |
| @p6 | int | 4 |
| @p7 | int | 4 |
| @p8 | int | 4 |
| @p9 | int | 4 |
| @p10 | int | 4 |
| @p11 | int | 4 |
| @p12 | int | 4 |
| @p13 | int | 4 |
| @p14 | int | 4 |
| @p15 | int | 4 |
| @p16 | int | 4 |
| @p17 | int | 4 |
| @p18 | int | 4 |
| @p19 | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_tag\_get\_by\_projectids]  @p0 int = null,  @p1 int = null,  @p2 int = null,  @p3 int = null,  @p4 int = null,  @p5 int = null,  @p6 int = null,  @p7 int = null,  @p8 int = null,  @p9 int = null,  @p10 int = null,  @p11 int = null,  @p12 int = null,  @p13 int = null,  @p14 int = null,  @p15 int = null,  @p16 int = null,  @p17 int = null,  @p18 int = null,  @p19 int = null  AS BEGIN  SET NOCOUNT ON  SELECT projectid, tag  FROM t\_project\_tag WITH (NOLOCK)  WHERE projectid in (@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7, @p8, @p9, @p10, @p11, @p12, @p13, @p14, @p15, @p16, @p17, @p18, @p19)  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

|  |
| --- |
| [dbo].[p\_project\_tag\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @tag | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_tag\_insert]  @projectid int,  @tag nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO t\_project\_tag  (  projectid,  tag  )  VALUES  (  @projectid,  @tag  )    set @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_tag]](#NVtxRSVjLeXCqup7XcDqnAEmXn0=)

|  |
| --- |
| [dbo].[p\_project\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @description | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update]  @projectid int,  @userid int,  @description nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE t\_project  SET [description] = @description,  updatedby = @userid,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_background] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @backgroundcolor | nvarchar(7) | 14 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update\_background]  @projectid int,  @userid int,  @backgroundcolor nvarchar(7),  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET backgroundcolor = @backgroundcolor,  updatedby = @userid,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_canvas\_size\_ratio] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @size\_ratio | nvarchar(7) | 14 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update\_canvas\_size\_ratio]  @projectid int,  @userid int,  @size\_ratio nvarchar(7),  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET canvas\_size\_ratio = @size\_ratio,  updatedby = @userid,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_deleted] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @deleted | bit | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_project\_update\_deleted]  @projectid int,  @userid int,  @deleted bit,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET deleted = @deleted,  updatedby = @userid,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_owner] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @newowner | int | 4 |  |
| @userid | int | 4 |  |
| @createdbyuniquemachineid | uniqueidentifier | 16 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update\_owner]  @projectid int,  @newowner int,  @userid int,  @createdbyuniquemachineid uniqueidentifier,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET createdby = @newowner,  createdbyuniquemachineid = @createdbyuniquemachineid,  updatedby = @userid,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  -- new owner must be deleted from t\_project\_participant  DELETE  FROM t\_project\_participant  WHERE projectid = @projectid  and userid = @newowner  -- old owner is now a user participating to the project  INSERT INTO t\_project\_participant (projectid, userid)  VALUES (@projectid, @userid)  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

|  |
| --- |
| [dbo].[p\_project\_update\_sas\_token] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @containeruri | nvarchar(500) | 1000 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update\_sas\_token]  @projectid int,  @containeruri nvarchar (500),  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET containeruri = @containeruri,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_template] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Id | int | 4 |  |
| @project­Type­Id | int | 4 |  |
| @language­Id | int | 4 |  |
| @description | nvarchar(250) | 500 |  |
| @extended­Description | nvarchar(max) | max |  |
| @more­Info­Url | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_update\_template]  @project­Id INT,  @project­Type­Id INT,  @language­Id INT,  @description NVARCHAR(250),  @extended­Description NVARCHAR(MAX),  @more­Info­Url NVARCHAR(250),  @rows INT OUTPUT  AS BEGIN  SET NOCOUNT ON  UPDATE dbo.t\_project  SET [projecttypeid] = @project­Type­Id  WHERE [projectid] = @project­Id  IF EXISTS (SELECT NULL FROM dbo.t\_template\_translation WHERE [projectid] = @project­Id AND [languageid] = @language­Id) BEGIN  UPDATE dbo.t\_template\_translation  SET [description] = @description,  [moreinfourl] = @more­Info­Url,  [extendeddescription] = @extended­Description,  [projectid] = @project­Id  WHERE [projectid] = @project­Id  AND [languageid] = @language­Id  END ELSE BEGIN  INSERT INTO dbo.t\_template\_translation ([projectid], [languageid], [description], [extendeddescription], [moreinfourl])  VALUES (@project­Id, @language­Id, @description, @extended­Description, @more­Info­Url)  END  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

|  |
| --- |
| [dbo].[p\_project\_update\_thumbnail] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @thumbnail | varbinary(max) | max |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_project\_update\_thumbnail]  @projectid int,  @thumbnail varbinary(max),  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET thumbnail = @thumbnail,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_update\_zoom\_offset] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @zoomfactor | float | 8 |  |
| @offsetx | float | 8 |  |
| @offsety | float | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_project\_update\_zoom\_offset]  @projectid int,  @zoomfactor float,  @offsetx float,  @offsety float,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_project  SET zoomfactor = @zoomfactor,  offsetx = @offsetx,  offsety = @offsety,  lastupdate = GETUTCDATE()  WHERE projectid = @projectid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

|  |
| --- |
| [dbo].[p\_project\_upgrade\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_upgrade\_book]  AS BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) dbo.t\_project\_upgrade\_booking  SET bookedby=@booked\_by,  bookingdate = @booking\_date,  lastupdate = @booking\_date,  statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('working')  WHERE statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('ready')  IF @@ROWCOUNT > 0  BEGIN  SELECT [Id],  [creationdate],  [username],  [projectid],  [currentversion],  [desiredversion],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry]  FROM dbo.t\_project\_upgrade\_booking  WHERE bookedby=@booked\_by  END    SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_upgrade\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_upgrade\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE dbo.t\_project\_upgrade\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN dbo.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  dbo.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec    END  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_upgrade\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_upgrade\_history]  AS  BEGIN  INSERT [dbo].[t\_project\_upgrade\_booking\_history]  (  [Id],  [creationdate],  [username],  [projectid],  [currentversion],  [desiredversion],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  )  SELECT  [Id],  [creationdate],  [username],  [projectid],  [currentversion],  [desiredversion],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  FROM  (  DELETE [dbo].[t\_project\_upgrade\_booking]  OUTPUT  DELETED.[Id],  DELETED.[creationdate],  DELETED.[username],  DELETED.[projectid],  DELETED.[currentversion],  DELETED.[desiredversion],  DELETED.[uniquedeviceid],  DELETED.[bookedby],  DELETED.[lastupdate],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[statemachinestatusid]  WHERE [statemachinestatusid] >= [dbo].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[t\_project\_upgrade\_booking\_history]](#lorYCHQjcbwJVwleNpEFauH2JOY=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_upgrade\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(200) | 400 |
| @projectid | int | 4 |
| @currentversion | int | 4 |
| @desiredversion | int | 4 |
| @uniquedeviceid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_upgrade\_insert]    @username NVARCHAR(200),  @projectid INT,  @currentversion INT,  @desiredversion INT,  @uniquedeviceid UNIQUEIDENTIFIER    AS  BEGIN  DECLARE @statemachinestatusid INT  SET @statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('ready')  INSERT INTO [dbo].[t\_project\_upgrade\_booking]  (  [creationdate],  [username],  [projectid],  [currentversion],  [desiredversion],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  )  VALUES  (  GETUTCDATE(),  @username,  @projectid,  @currentversion,  @desiredversion,  @uniquedeviceid,  NULL,  GETUTCDATE(),  NULL,  0,  @statemachinestatusid  )  END  RETURN 0  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_project\_upgrade\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_project\_upgrade\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  UPDATE dbo.t\_project\_upgrade\_booking  SET  bookedby=NULL,  bookingdate=NULL,  retry=@retries,  statemachinestatusid=@statemachinestatusid,  lastupdate = GETUTCDATE()  WHERE  Id=@id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_project\_upgrade\_booking]](#gfuU7fskjWXvIyRe4VWGy2an6Kw=)

|  |
| --- |
| [dbo].[p\_projecttypes\_convert\_to\_template] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @project­Id | int | 4 |
| @project­Type­Id | int | 4 |
| @description | nvarchar(250) | 500 |
| @extended­Description | nvarchar(max) | max |
| @more­Info­Url | nvarchar(250) | 500 |
| @zoom­Factor | float | 8 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_convert\_to\_template]  @project­Id INT,  @project­Type­Id INT,  @description NVARCHAR(250),  @extended­Description NVARCHAR(MAX),  @more­Info­Url NVARCHAR(250),  @zoom­Factor FLOAT  AS BEGIN  SET NOCOUNT ON  INSERT INTO dbo.t\_template\_translation  (  projectid,  languageid,  [description],  extendeddescription,  moreinfourl  )  VALUES  ( @project­Id, -- projectid - int  1, -- Default language  @description, -- description - nvarchar(250)  @extended­Description, -- extendeddescription - nvarchar(max)  @more­Info­Url -- moreinfourl - nvarchar(250)  )  UPDATE dbo.t\_project  SET projecttypeid = @project­Type­Id,  createdby = 1,  zoomfactor = @zoom­Factor  WHERE projectid = @project­Id  -- Remove the participants as well  DELETE FROM dbo.t\_project\_participant  WHERE projectid = @project­Id  END  GO |

Uses

[[dbo].[t\_project]](#7A8kMkf0Z+G9i1FAStnpsh7dJtg=)

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_template\_translation]](#BpdWJknsRmdVD81ROP738LoU6EU=)

|  |
| --- |
| [dbo].[p\_projecttypes\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projecttypeid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_delete]  @projecttypeid INT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  -- Delete the project type translations  DELETE FROM dbo.t\_project\_type\_translation  WHERE projecttypeid = @projecttypeid  -- Delete the project type  DELETE FROM dbo.t\_project\_type  WHERE projecttypeid = @projecttypeid  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

|  |
| --- |
| [dbo].[p\_projecttypes\_get\_all\_template\_types] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @language­Id | int | 4 |  |
| @page­Number | int | 4 |  |
| @page­Size | int | 4 |  |
| @total­Rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_get\_all\_template\_types]  @language­Id INT = 1,  @page­Number INT,  @page­Size INT,  @total­Rows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT  b.description description,  a.projecttypeid projecttypeid ,  b.languageid languageid  FROM dbo.t\_project\_type AS a  LEFT JOIN dbo.t\_project\_type\_translation AS b ON b.projecttypeid = a.projecttypeid  WHERE  (a.projecttypeid = 1 OR (a.projecttypeid & 256 > 0 ))  AND  (b.languageid = @language­Id  OR b.languageid = 1  OR b.languageid IS NULL)  ORDER BY a.projecttypeid, b.languageid  OFFSET @page­Size \* (@page­Number - 1) ROWS  FETCH NEXT @page­Size ROWS ONLY;  SELECT @total­Rows = count(1)  FROM dbo.t\_project\_type AS a  LEFT JOIN dbo.t\_project\_type\_translation AS b ON b.projecttypeid = a.projecttypeid  WHERE  (a.projecttypeid = 1 OR (a.projecttypeid & 256 > 0 ))  AND  (b.languageid = @language­Id  OR b.languageid = 1  OR b.languageid IS NULL)  END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

|  |
| --- |
| [dbo].[p\_projecttypes\_insert\_new\_projecttype] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @description | nvarchar(250) | 500 |  |
| @projecttypeid | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_insert\_new\_projecttype]  @description NVARCHAR(250),  @projecttypeid INT OUTPUT  AS BEGIN  SET NOCOUNT ON  IF EXISTS( SELECT [description] FROM dbo.t\_project\_type WHERE [description] = @description)  BEGIN  SELECT @projecttypeid=projecttypeid FROM dbo.t\_project\_type WHERE [description] = @description  RETURN  END  ELSE  BEGIN  IF NOT EXISTS (SELECT \* FROM dbo.t\_project\_type WHERE projecttypeid>255)  BEGIN  SET @projecttypeid=256  END  ELSE  BEGIN  SELECT @projecttypeid=(MAX(projecttypeid)+1) FROM dbo.t\_project\_type  END  INSERT INTO dbo.t\_project\_type  (  projecttypeid,  description  )  VALUES  ( @projecttypeid,  @description  )    -- Also set the default, English version  INSERT INTO dbo.t\_project\_type\_translation  (  projecttypeid,  languageid,  description  )  VALUES  ( @projecttypeid,  1, -- Default is english  @description  )  END  END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

|  |
| --- |
| [dbo].[p\_projecttypes\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projecttypeid | int | 4 |  |
| @description | nvarchar(250) | 500 |  |
| @language­Id | int | 4 |  |
| @translated­Description | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_update]  @projecttypeid INT,  @description NVARCHAR(250),  @language­Id INT,  @translated­Description NVARCHAR(250),  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  -- Update the project type  UPDATE dbo.t\_project\_type  SET [description] = @description  where projecttypeid = @projecttypeid  SET @rows = @@ROWCOUNT  -- Update the project type translation  IF EXISTS (SELECT NULL FROM dbo.t\_project\_type\_translation WHERE projecttypeid = @projecttypeid and languageid = @language­Id) BEGIN  UPDATE dbo.t\_project\_type\_translation  SET [description] = @translated­Description  WHERE projecttypeid = @projecttypeid  AND languageid = @language­Id  END ELSE BEGIN  INSERT INTO dbo.t\_project\_type\_translation (projecttypeid, languageid, [description])  VALUES (@projecttypeid, @language­Id, @translated­Description)  END  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

[[dbo].[t\_project\_type\_translation]](#2FXHl0VtJaQYGTZWPJRvQ1jyQEw=)

|  |
| --- |
| [dbo].[p\_projecttypes\_verify\_if\_template] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @project­Type­Id | int | 4 |  |
| @result | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_projecttypes\_verify\_if\_template]  @project­Type­Id INT,  @result BIT OUTPUT  AS BEGIN  IF (@project­Type­Id <> 1) AND (@project­Type­Id & 256 = 0)  BEGIN  SET @result=0  END  ELSE  BEGIN  SET @result=1  END  IF(@result=1)  BEGIN  SELECT @result =  CASE  WHEN EXISTS(SELECT \* FROM dbo.t\_project\_type WHERE projecttypeid=@project­Type­Id)  THEN 1  ELSE 0  END  END  END  GO |

Uses

[[dbo].[t\_project\_type]](#GyGD46URFdCG64D0BvNZhMpw4IA=)

|  |
| --- |
| [dbo].[p\_quicklink\_delete] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @quicklinkid | int | 4 |
| @userid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_delete]  @projectid INT,  @quicklinkid INT,  @userid INT    AS  BEGIN  UPDATE dbo.t\_quicklink  SET  isdeleted = 1,  lastupdate= GETUTCDATE(),  lastupdatedby = @userid  WHERE projectid = @projectid  AND quicklinkid = @quicklinkid  RETURN 0  END  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

|  |
| --- |
| [dbo].[p\_quicklink\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @quicklinkid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_get\_by\_id]  @projectid INT,  @quicklinkid INT  AS  SELECT TOP(1)  ql.quicklinkid,  ql.projectid,  ql.name,  ql.createdby,  ql.creationdate,  ql.lastupdate,  ql.lastupdatedby,  ql.thumbnail  FROM  dbo.t\_quicklink ql WITH (NOLOCK)  WHERE  projectid = @projectid  AND  ql.quicklinkid = @quicklinkid  AND  ql.isdeleted = 0  RETURN 0  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

|  |
| --- |
| [dbo].[p\_quicklink\_get\_by\_projectid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @pagesize | int | 4 |  |
| @pagenum | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_get\_by\_projectid]  @projectid INT,  @wants­Count BIT,  @pagesize INT,  @pagenum INT,  @totalrows INT =0 OUTPUT  AS  SELECT  ql.quicklinkid,  ql.projectid,  ql.name,  ql.createdby,  ql.creationdate,  ql.lastupdate,  ql.lastupdatedby,  ql.thumbnail  FROM  dbo.t\_quicklink ql WITH (NOLOCK)  WHERE  projectid = @projectid  AND  isdeleted = 0  ORDER BY  creationdate ASC  OFFSET @pagesize\* (@pagenum-1) ROWS  FETCH NEXT @pagesize ROWS ONLY  IF (@wants­Count = 1)  BEGIN  SELECT  @totalrows = COUNT(0)  FROM  dbo.t\_quicklink WITH (NOLOCK)  WHERE  projectid = @projectid  AND  isdeleted = 0  END  ELSE  BEGIN  SET @totalrows = 0  END  RETURN 0  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

|  |
| --- |
| [dbo].[p\_quicklink\_get\_tiles\_for\_quicklink] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @quicklinkid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_get\_tiles\_for\_quicklink]  @quicklinkid INT  AS  BEGIN  SET NOCOUNT ON  SELECT tileid  FROM dbo.t\_quicklink\_tile WITH (NOLOCK)  WHERE  quicklinkid = @quicklinkid  SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

|  |
| --- |
| [dbo].[p\_quicklink\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @createdby | int | 4 |
| @name | nvarchar(250) | 500 |
| @thumbnail | varbinary(max) | max |
| @uniquedeviceid | uniqueidentifier | 16 |
| @tileids | Tile­Status­Id­Set | max |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_insert]  @projectid INT,  @createdby INT,  @name NVARCHAR(250),  @thumbnail VARBINARY(MAX),  @uniquedeviceid UNIQUEIDENTIFIER,  @tileids dbo.Tile­Status­Id­Set READONLY  AS  BEGIN  SET NOCOUNT ON  DECLARE @creationdate DATETIME = GETUTCDATE()  DECLARE @lastupdate DATETIME = GETUTCDATE()  DECLARE @quicklinkid INT  -- Create the tally table to avoid using (slow!) cursors  DECLARE @Tally­Table TABLE (Iterator INT IDENTITY(1,1) NOT NULL, tile­Id UNIQUEIDENTIFIER NOT NULL);  INSERT INTO @Tally­Table (tile­Id)  SELECT Tile­Id FROM @tileids    -- Create the main quicklink entity  INSERT INTO dbo.t\_quicklink (  projectid,  createdby,  creationdate,  lastupdate,  lastupdatedby,  [name],  thumbnail,  createdbyuniquemachineid,  isdeleted  )  VALUES (  @projectid,  @createdby,  @creationdate,  @lastupdate,  @createdby,  @name,  @thumbnail,  @uniquedeviceid,  0)  SET @quicklinkid = @@IDENTITY  -- Insert all tile­Ids into the table  DECLARE @Row­Count INT  DECLARE @set­Id INT = 1  DECLARE @tile­Id UNIQUEIDENTIFIER  SELECT @Row­Count = COUNT(0) FROM @tileids  -- Loop through all given tile­Ids  WHILE @set­Id <= @Row­Count  BEGIN  SELECT @tile­Id = Tile­Id FROM @Tally­Table  WHERE Iterator = @set­Id  INSERT INTO [dbo].[t\_quicklink\_tile]  (quicklinkid, tileid)  VALUES  (@quicklinkid, @tile­Id)  SET @set­Id = @set­Id+1  END  SET NOCOUNT OFF    END    RETURN 0  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

[dbo].[Tile­Status­Id­Set]

|  |
| --- |
| [dbo].[p\_quicklink\_remove\_tile\_from\_quicklinks] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tileid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_remove\_tile\_from\_quicklinks]  @tileid UNIQUEIDENTIFIER  AS  BEGIN  -- First, select all quicklinks this is part off  SELECT ql.quicklinkid  INTO #affected­Quick­Links  FROM dbo.t\_quicklink ql WITH (NOLOCK)  JOIN dbo.t\_quicklink\_tile qlt ON qlt.quicklinkid = ql.quicklinkid    WHERE qlt.tileid = @tileid  -- Delete all tiles  DELETE FROM dbo.t\_quicklink\_tile WHERE tileid = @tile­ID  -- Now, see if there are quicklinks that have NO tiles. If they do, delete them as well  UPDATE dbo.t\_quicklink  SET isdeleted = 1  WHERE EXISTS (SELECT quicklinkid FROM #affected­Quick­Links WHERE quicklinkid = t\_quicklink.quicklinkid)  AND NOT EXISTS (SELECT quicklinkid FROM dbo.t\_quicklink\_tile WHERE quicklinkid = t\_quicklink.quicklinkid )  END  RETURN 0  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

|  |
| --- |
| [dbo].[p\_quicklink\_update] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @quicklinkid | int | 4 |
| @lastupdatedby | int | 4 |
| @name | nvarchar(250) | 500 |
| @thumbnail | varbinary(max) | max |
| @tileids | Tile­Status­Id­Set | max |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_quicklink\_update]  @projectid INT,  @quicklinkid INT,  @lastupdatedby INT,  @name NVARCHAR(250),  @thumbnail VARBINARY(MAX),  @tileids dbo.Tile­Status­Id­Set READONLY  AS  BEGIN  SET NOCOUNT ON    DECLARE @lastupdate DATETIME = GETUTCDATE()    -- Create the main quicklink entity  UPDATE dbo.t\_quicklink  SET  lastupdate = @lastupdate,  lastupdatedby = @lastupdatedby,  [name]=@name,  thumbnail=@thumbnail  WHERE quicklinkid = @quicklinkid  AND projectid = @projectid  AND isdeleted = 0  IF @@ROWCOUNT > 0  BEGIN  -- Create the tally table to avoid using (slow!) cursors  DECLARE @Tally­Table TABLE (Iterator INT IDENTITY(1,1) NOT NULL, tile­Id UNIQUEIDENTIFIER NOT NULL);  INSERT INTO @Tally­Table (tile­Id)  SELECT Tile­Id FROM @tileids  -- Delete all the old tiles.  DELETE FROM dbo.t\_quicklink\_tile  WHERE quicklinkid = @quicklinkid  -- Insert all tile­Ids into the table  DECLARE @Row­Count INT  DECLARE @set­Id INT = 1  DECLARE @tile­Id UNIQUEIDENTIFIER  SELECT @Row­Count = COUNT(0) FROM @tileids  -- Loop through all given tile­Ids  WHILE @set­Id <= @Row­Count  BEGIN  SELECT @tile­Id = Tile­Id FROM @Tally­Table  WHERE Iterator = @set­Id  INSERT INTO [dbo].[t\_quicklink\_tile]  (quicklinkid, tileid)  VALUES  (@quicklinkid, @tile­Id)  SET @set­Id = @set­Id+1  END  END -- IF @@ROWCOUNT > 0  SET NOCOUNT OFF    END    RETURN 0  GO |

Uses

[[dbo].[t\_quicklink]](#8R5txRxbFoGoF2i7BOacoCDM29c=)

[[dbo].[t\_quicklink\_tile]](#ysQ1DFp6SF/FNK2j6CNl4tj0nZ8=)

[dbo].[Tile­Status­Id­Set]

|  |
| --- |
| [dbo].[p\_storage\_eraser\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_storage\_eraser\_book]  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate smalldatetime  set @bookingdate = GETUTCDATE()  --part of unsed code  --SET NOCOUNT ON  --UPDATE top(5) t\_tilestatus  --SET bookedby = @bookeby,  -- bookingdate = @bookingdate,  -- statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  --WHERE (blobstatusid = dbo.fn\_­Get­Blob­Status('DELETED') or blobstatusid = dbo.fn\_­Get­Blob­Status('UPLOADERROR'))  --AND statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY')  --SELECT tileid, projectid, parentid, tiletype, tileorder, centerx, centery, height, width, zindex, scale, rotation, isactive,  -- isnew, tileshape, tilecontenttype, gridlibrarytype, backgroundcolor, syncid, blobstatusid, header,  -- lockeduser, lockedby, lockeddate, lastupdate, bookedby, bookingdate, retry, statemachinestatusid  --FROM t\_tilestatus --with(nolock)  --WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

|  |
| --- |
| [dbo].[p\_storage\_eraser\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @retry | int | 4 |  |
| @statemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_storage\_eraser\_unbook]  @tileid uniqueidentifier,  @retry int,  @statemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET bookedby = null,  bookingdate = null,  retry = @retry,  statemachinestatusid = @statemachinestatusid  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_telemetry\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @Trace­Data | nvarchar(1024) | 2048 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_telemetry\_insert]  -- Add the parameters for the stored procedure here  (  @Trace­Data nvarchar(1024)  )  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  INSERT INTO [dbo].[t\_telemetrydata]  (Creation­Date­Time, Log­Entry)  VALUES  (GETUTCDATE(), @Trace­Data)  END  GO |

Uses

[[dbo].[t\_telemetrydata]](#cjQE5M9B1IDxcJRkAEXqhevcxLc=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_by\_participating\_tile] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @requested­Ids | Tile­Status­Id­Set | max |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Dennis Vroegop  -- Create Date: 05/07/2020  -- Description: Returns all relations that have one endpoint in a given set  -- =============================================  CREATE PROCEDURE [dbo].[p\_tile\_relation\_by\_participating\_tile]  (  -- Add the parameters for the stored procedure here  @requested­Ids [dbo].Tile­Status­Id­Set READONLY  )  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON  DECLARE @project­Id INT  SELECT @project­Id = projectid FROM dbo.t\_tilestatus WHERE tileid = (SELECT TOP(1) Tile­Id FROM @requested­Ids)  ;WITH tree AS (  SELECT  t1.tileid AS tile­Id,  t1.parentid AS parent­Id,  [level] = 1  FROM dbo.t\_tilestatus t1  WHERE t1.parentid IS NULL  AND t1.projectid = @project­Id  AND EXISTS  (SELECT Tile­Id FROM @requested­Ids WHERE t1.tileid = Tile­Id)  UNION ALL  SELECT  t2.tileid AS tile­ID,  t2.parentid AS parent­Id,  [level] = tree.[level] +1  FROM dbo.t\_tilestatus t2 INNER JOIN  tree ON tree.tile­Id = t2.parentid  WHERE t2.projectid = @project­Id  )  SELECT r.tilerelationid,  r.projectid,  r.tileid,  r.related\_tileid,  r.tilerelationstyleid,  r.tilerelationanchoridorigin,  r.tilerelationanchoriddestination,  r.tilerelationstartendsymbolorigin,  r.tilerelationstartendsymboldestination AS dest,  r.tilerelationlineid,  r.color,  r.thickness  FROM [dbo].t\_tile\_relation r  WHERE  EXISTS (SELECT tile­Id FROM tree WHERE Tile­Id = r.tileid)  INTERSECT  SELECT r.tilerelationid,  r.projectid,  r.tileid,  r.related\_tileid,  r.tilerelationstyleid,  r.tilerelationanchoridorigin,  r.tilerelationanchoriddestination,  r.tilerelationstartendsymbolorigin,  r.tilerelationstartendsymboldestination AS dest,  r.tilerelationlineid,  r.color,  r.thickness  FROM [dbo].t\_tile\_relation r  WHERE  EXISTS (SELECT tile­Id FROM tree WHERE Tile­Id = r.related\_tileid)  -- Insert statements for procedure here  SELECT r.tilerelationid,  r.projectid,  r.tileid,  r.related\_tileid,  r.tilerelationstyleid,  r.tilerelationanchoridorigin,  r.tilerelationanchoriddestination,  r.tilerelationstartendsymbolorigin,  r.tilerelationstartendsymboldestination AS dest,  r.tilerelationlineid,  r.color,  r.thickness FROM [dbo].t\_tile\_relation r  WHERE EXISTS (SELECT Tile­Id FROM @requested­Ids WHERE Tile­Id = r.tileid)  INTERSECT  SELECT r.tilerelationid,  r.projectid,  r.tileid,  r.related\_tileid,  r.tilerelationstyleid,  r.tilerelationanchoridorigin,  r.tilerelationanchoriddestination,  r.tilerelationstartendsymbolorigin,  r.tilerelationstartendsymboldestination AS dest,  r.tilerelationlineid,  r.color,  r.thickness FROM [dbo].t\_tile\_relation r  WHERE EXISTS (SELECT Tile­Id FROM @requested­Ids WHERE Tile­Id = r.related\_tileid)  END  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[dbo].[Tile­Status­Id­Set]

|  |
| --- |
| [dbo].[p\_tile\_relation\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tilerelationid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tile\_relation\_delete]  @tilerelationid int,  @rows int = 0 output  AS  SET NOCOUNT ON  DELETE FROM t\_tile\_relation  WHERE tilerelationid = @tilerelationid    SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_delete\_all\_by\_tileid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tile\_relation\_delete\_all\_by\_tileid]  @tileid uniqueidentifier,  @rows int = 0 output  AS  SET NOCOUNT ON  DELETE FROM t\_tile\_relation  WHERE tileid = @tileid  OR related\_tileid = @tileid  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_get\_by\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_relation\_get\_by\_projectid]  @projectid int  AS  SET NOCOUNT ON  SELECT tilerelationid,  projectid,  tileid,  related\_tileid,  tilerelationstyleid,  tilerelationanchoridorigin,  tilerelationanchoriddestination,  tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination as dest,  tilerelationlineid,  color,  thickness  FROM t\_tile\_relation WITH (NOLOCK)  WHERE projectid = @projectid  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_get\_by\_projectid\_paged] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @tocount | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_relation\_get\_by\_projectid\_paged]  @projectid int,  @pagesize int,  @pagenumber int,  @tocount bit,  @totalrows int = 0 output  AS  SET NOCOUNT ON  SELECT tilerelationid,  projectid,  tileid,  related\_tileid,  tilerelationstyleid,  tilerelationanchoridorigin,  tilerelationanchoriddestination,  tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination as dest,  tilerelationlineid,  color,  thickness  FROM t\_tile\_relation WITH (NOLOCK)  WHERE projectid = @projectid  ORDER BY tilerelationid  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF(@tocount = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_tile\_relation WITH (NOLOCK)  WHERE projectid = @projectid  END  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tilerelationid | int | 4 | Out |
| @projectid | int | 4 |  |
| @tileid | uniqueidentifier | 16 |  |
| @related\_tileid | uniqueidentifier | 16 |  |
| @tilerelationstyleid | int | 4 |  |
| @tilerelationanchoridorigin | int | 4 |  |
| @tilerelationanchoriddestination | int | 4 |  |
| @tilerelationstartendsymbolorigin | int | 4 |  |
| @dest | int | 4 |  |
| @tilerelationlineid | int | 4 |  |
| @color | nvarchar(7) | 14 |  |
| @thickness | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_relation\_insert]  @tilerelationid int output,  @projectid int,  @tileid uniqueidentifier,  @related\_tileid uniqueidentifier,  @tilerelationstyleid int,  @tilerelationanchoridorigin int,  @tilerelationanchoriddestination int,  @tilerelationstartendsymbolorigin int,  @dest int,  @tilerelationlineid int,  @color nvarchar(7),  @thickness int,  @rows int = 0 output  AS  SET NOCOUNT ON  INSERT INTO t\_tile\_relation  (  projectid,  tileid,  related\_tileid,  tilerelationstyleid,  tilerelationanchoridorigin,  tilerelationanchoriddestination,  tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination,  tilerelationlineid,  color,  thickness  )  values  (  @projectid,  @tileid,  @related\_tileid,  @tilerelationstyleid,  @tilerelationanchoridorigin,  @tilerelationanchoriddestination,  @tilerelationstartendsymbolorigin,  @dest,  @tilerelationlineid,  @color,  @thickness  )  set @tilerelationid = @@IDENTITY  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tilerelationid | int | 4 |  |
| @projectid | int | 4 |  |
| @tileid | uniqueidentifier | 16 |  |
| @related\_tileid | uniqueidentifier | 16 |  |
| @tilerelationstyleid | int | 4 |  |
| @tilerelationanchoridorigin | int | 4 |  |
| @tilerelationanchoriddestination | int | 4 |  |
| @tilerelationstartendsymbolorigin | int | 4 |  |
| @dest | int | 4 |  |
| @tilerelationlineid | int | 4 |  |
| @color | nvarchar(7) | 14 |  |
| @thickness | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_relation\_update]  @tilerelationid int,  @projectid int,  @tileid uniqueidentifier,  @related\_tileid uniqueidentifier,  @tilerelationstyleid int,  @tilerelationanchoridorigin int,  @tilerelationanchoriddestination int,  @tilerelationstartendsymbolorigin int,  @dest int,  @tilerelationlineid int,  @color nvarchar(7),  @thickness int,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_tile\_relation  SET tileid = @tileid,  related\_tileid = @related\_tileid,  tilerelationstyleid = @tilerelationstyleid,  tilerelationanchoridorigin = @tilerelationanchoridorigin,  tilerelationanchoriddestination = @tilerelationanchoriddestination,  tilerelationstartendsymbolorigin = @tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination = @dest,  tilerelationlineid = @tilerelationlineid,  color = @color,  thickness = @thickness  WHERE tilerelationid = @tilerelationid  AND projectid = @projectid  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_relation\_with\_tile] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tile­Id | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_relation\_with\_tile]  (  -- Add the parameters for the stored procedure here  @tile­Id uniqueidentifier    )  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON    SELECT tilerelationid,  projectid,  tileid,  related\_tileid,  tilerelationstyleid,  tilerelationanchoridorigin,  tilerelationanchoriddestination,  tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination AS dest,  tilerelationlineid,  color,  thickness FROM dbo.t\_tile\_relation  WHERE tileid = @tile­Id  UNION  SELECT tilerelationid,  projectid,  tileid,  related\_tileid,  tilerelationstyleid,  tilerelationanchoridorigin,  tilerelationanchoriddestination,  tilerelationstartendsymbolorigin,  tilerelationstartendsymboldestination AS dest,  tilerelationlineid,  color,  thickness FROM dbo.t\_tile\_relation  WHERE related\_tileid = @tile­Id  END  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

|  |
| --- |
| [dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |
| @userid | int | 4 |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id]  @projectid int,  @userid int  AS  SET NOCOUNT ON;  SELECT tileid,  rate,  thumb  FROM t\_tile\_voting with(nolock)  WHERE projectid = @projectid  AND userid = @userid  RETURN  GO |

Uses

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

|  |
| --- |
| [dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id\_paged] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @userid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @tocount | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_voting\_get\_by\_projectid\_and\_user\_id\_paged]  @projectid int,  @userid int,  @pagesize int,  @pagenumber int,  @tocount bit,  @totalrows int = 0 output  AS  SET NOCOUNT ON;  SELECT tileid,  rate,  thumb  FROM t\_tile\_voting with(nolock)  WHERE projectid = @projectid  AND userid = @userid  ORDER BY projectid, userid  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF(@tocount = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_tile\_voting with(nolock)  WHERE projectid = @projectid  AND userid = @userid  END  RETURN  GO |

Uses

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

|  |
| --- |
| [dbo].[p\_tile\_voting\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @userid | int | 4 |  |
| @projectid | int | 4 |  |
| @rate | int | 4 |  |
| @thumb | decimal(18,9) | 9 |  |
| @thumbs | decimal(18,9) | 9 | Out |
| @ratingsum | int | 4 | Out |
| @ratingcount | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tile\_voting\_update]  @tileid uniqueidentifier,  @userid int,  @projectid int,  @rate int,  @thumb decimal(18,9),  @thumbs decimal(18,9) output,  @ratingsum int output,  @ratingcount int output,  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_tile\_voting  SET rate = @rate,  thumb = @thumb  WHERE tileid = @tileid  and userid = @userid  SET @rows = @@ROWCOUNT  IF @rows=0  BEGIN  INSERT INTO t\_tile\_voting (tileid, userid, projectid, rate, thumb)  VALUES (@tileid, @userid, @projectid, @rate, @thumb)  SET @rows = @@ROWCOUNT  END  IF @rows=1  BEGIN  SELECT @thumbs = SUM(thumb)  FROM t\_tile\_voting  WHERE tileid = @tileid  SELECT @ratingsum = SUM(rate), @ratingcount = COUNT(tileid)  FROM t\_tile\_voting with (nolock)  WHERE tileid = @tileid  and rate <> 0  END  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

|  |
| --- |
| [dbo].[p\_tilecopy\_cleanup] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_cleanup]  AS  BEGIN  BEGIN TRANSACTION p\_tilecopy\_cleanup;  BEGIN TRY  DECLARE @cutoff­Time DATETIME  SET @cutoff­Time =DATEADD(MINUTE, -5, GETUTCDATE())  DELETE FROM dbo.t\_tilecopy\_sourcetile  WHERE copyfinishedtimestamp < @cutoff­Time  DELETE FROM dbo.t\_tilecopy\_copiedtile  WHERE copyfinishedtimestamp < @cutoff­Time  DELETE FROM dbo.t\_tilecopy\_copiedrelation  WHERE copyfinishedtimestamp < @cutoff­Time  COMMIT TRANSACTION p\_tilecopy\_cleanup;  END TRY  BEGIN CATCH  DECLARE @error INT,  @message VARCHAR(4000)  SELECT  @error = ERROR\_­NUMBER(),  @message = ERROR\_­MESSAGE()      ROLLBACK TRANSACTION p\_tilecopy\_cleanup;  RAISERROR('p\_tilecopy\_cleanup: %d: %s', 16, 1, @error, @message);  END CATCH  END  GO |

Uses

[[dbo].[t\_tilecopy\_copiedrelation]](#TpEVDYGh4J/aV/9zW2Qgqvq2Ku8=)

[[dbo].[t\_tilecopy\_copiedtile]](#z32gliNZ8xOJvfNpb2w8q4TnC08=)

[[dbo].[t\_tilecopy\_sourcetile]](#Q2UYJrfAzp+CQwZGWFDTwEseGMI=)

|  |
| --- |
| [dbo].[p\_tilecopy\_getcopiedtiles\_by\_token] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @page­Size | int | 4 |
| @page­Number | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_getcopiedtiles\_by\_token]  -- Add the parameters for the stored procedure here  @token UNIQUEIDENTIFIER,  @page­Size INT,  @page­Number INT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;    -- if the parameters are null, set them to respectively  -- MAXINT and 0 to not use pages  SET @page­Size = ISNULL(@page­Size, 2147483647)  SET @page­Number = ISNULL(@page­Number, 0)  SELECT [id]  ,[token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tileid]  ,[projectid]  ,[parentid]  ,[tiletype]  ,[tilecontent]  ,[positionx]  ,[positiony]  ,[height]  ,[width]  ,[zindex]  ,[scalex]  ,[scaley]  ,[rotation]  ,[totalthumbup]  ,[totlathumbdown]  ,[overallrating]  ,[totalrates]  ,[ispinned]  ,[backgroundcolor]  ,[originalfilename]  ,[lockeduser]  ,[lockedby]  ,[lockeddate]  ,[lastupdate]  ,[blobstatusid]  ,[old­Id]  ,[bookedby]  ,[bookingdate]  ,[retry]  ,[statemachinestatusid]  FROM [dbo].[t\_tilecopy\_copiedtile]  WHERE [token] = @token  ORDER BY [copyfinishedtimestamp]  OFFSET @page­Size \* (@page­Number-1) ROWS FETCH NEXT @page­Size ROWS ONLY  END  GO |

Uses

[[dbo].[t\_tilecopy\_copiedtile]](#z32gliNZ8xOJvfNpb2w8q4TnC08=)

|  |
| --- |
| [dbo].[p\_tilecopy\_getrelations\_by\_token] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @page­Size | int | 4 |
| @page­Number | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_getrelations\_by\_token]  -- Add the parameters for the stored procedure here  @token UNIQUEIDENTIFIER,  @page­Size INT,  @page­Number INT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- if the parameters are null, set them to respectively  -- MAXINT and 0 to not use pages  SET @page­Size = ISNULL(@page­Size, 2147483647)  SET @page­Number = ISNULL(@page­Number, 0)  SELECT [id]  ,[token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tilerelationid]  ,[projectid]  ,[tileid]  ,[related\_tileid]  ,[tilerelationstyleid]  ,[tilerelationanchoridorigin]  ,[tilerelationanchoriddestination]  ,[tilerelationstartendsymbolorigin]  ,[tilerelationstartendsymboldestination]  ,[tilerelationlineid]  ,[color]  ,[thickness]  FROM [dbo].[t\_tilecopy\_copiedrelation]  WHERE [token] = @token  ORDER BY [copyfinishedtimestamp]  OFFSET @page­Size \* (@page­Number-1) ROWS FETCH NEXT @page­Size ROWS ONLY  END  GO |

Uses

[[dbo].[t\_tilecopy\_copiedrelation]](#TpEVDYGh4J/aV/9zW2Qgqvq2Ku8=)

|  |
| --- |
| [dbo].[p\_tilecopy\_getsourcetiles\_by\_token] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @page­Size | int | 4 |
| @page­Number | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_getsourcetiles\_by\_token]  -- Add the parameters for the stored procedure here  @token UNIQUEIDENTIFIER,  @page­Size INT,  @page­Number INT  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- if the parameters are null, set them to respectively  -- MAXINT and 0 to not use pages  SET @page­Size = ISNULL(@page­Size, 2147483647)  SET @page­Number = ISNULL(@page­Number, 0)  SELECT [id]  ,[token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tileid]  FROM [dbo].[t\_tilecopy\_sourcetile]  WHERE [token] = @token  ORDER BY [copyfinishedtimestamp]  OFFSET @page­Size \* (@page­Number-1) ROWS FETCH NEXT @page­Size ROWS ONLY  END  GO |

Uses

[[dbo].[t\_tilecopy\_sourcetile]](#Q2UYJrfAzp+CQwZGWFDTwEseGMI=)

|  |
| --- |
| [dbo].[p\_tilecopy\_insertcopiedtile] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @copyfinishedtimestamp | datetime | 8 |
| @usernamecreator | nvarchar(50) | 100 |
| @sourceprojectid | int | 4 |
| @targetprojectid | int | 4 |
| @tileid | uniqueidentifier | 16 |
| @projectid | int | 4 |
| @parentid | uniqueidentifier | 16 |
| @tiletype | int | 4 |
| @tilecontent | nvarchar(max) | max |
| @positionx | float | 8 |
| @positiony | float | 8 |
| @height | float | 8 |
| @width | float | 8 |
| @zindex | int | 4 |
| @scalex | float | 8 |
| @scaley | float | 8 |
| @rotation | float | 8 |
| @totalthumbup | int | 4 |
| @totalthumbdown | int | 4 |
| @overallrating | decimal(2,1) | 5 |
| @totalrates | int | 4 |
| @ispinned | bit | 1 |
| @backgroundcolor | nvarchar(7) | 14 |
| @originalfilename | nvarchar(255) | 510 |
| @lockeduser | nvarchar(50) | 100 |
| @lockedby | nvarchar(50) | 100 |
| @lockeddate | datetime | 8 |
| @lastupdate | datetime | 8 |
| @blobstatusid | int | 4 |
| @old­Id | uniqueidentifier | 16 |
| @bookedby | uniqueidentifier | 16 |
| @bookingdate | datetime | 8 |
| @retry | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_insertcopiedtile]  -- Add the parameters for the stored procedure here  @token [UNIQUEIDENTIFIER],  @copyfinishedtimestamp [DATETIME],  @usernamecreator [NVARCHAR](50),  @sourceprojectid [INT],  @targetprojectid [INT],  @tileid [UNIQUEIDENTIFIER],  @projectid [INT],  @parentid [UNIQUEIDENTIFIER],  @tiletype [INT],  @tilecontent [NVARCHAR](MAX),  @positionx [FLOAT],  @positiony [FLOAT],  @height [FLOAT],  @width [FLOAT],  @zindex [INT],  @scalex [FLOAT],  @scaley [FLOAT],  @rotation [FLOAT],  @totalthumbup [INT],  @totalthumbdown [INT],  @overallrating [DECIMAL](2, 1),  @totalrates [INT],  @ispinned [BIT],  @backgroundcolor [NVARCHAR](7),  @originalfilename [NVARCHAR](255),  @lockeduser [NVARCHAR](50),  @lockedby [NVARCHAR](50),  @lockeddate [DATETIME],  @lastupdate [DATETIME],  @blobstatusid [INT],  @old­Id UNIQUEIDENTIFIER,  @bookedby [UNIQUEIDENTIFIER],  @bookingdate [DATETIME],  @retry [INT],  @statemachinestatusid [INT]  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  INSERT INTO [dbo].[t\_tilecopy\_copiedtile]  ([token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tileid]  ,[projectid]  ,[parentid]  ,[tiletype]  ,[tilecontent]  ,[positionx]  ,[positiony]  ,[height]  ,[width]  ,[zindex]  ,[scalex]  ,[scaley]  ,[rotation]  ,[totalthumbup]  ,[totlathumbdown]  ,[overallrating]  ,[totalrates]  ,[ispinned]  ,[backgroundcolor]  ,[originalfilename]  ,[lockeduser]  ,[lockedby]  ,[lockeddate]  ,[lastupdate]  ,[blobstatusid]  ,[old­Id]  ,[bookedby]  ,[bookingdate]  ,[retry]  ,[statemachinestatusid])  VALUES  (  @token  ,@copyfinishedtimestamp  ,@usernamecreator  ,@sourceprojectid  ,@targetprojectid  ,@tileid  ,@projectid  ,@parentid  ,@tiletype  ,@tilecontent  ,@positionx  ,@positiony  ,@height  ,@width  ,@zindex  ,@scalex  ,@scaley  ,@rotation  ,@totalthumbup  ,@totalthumbdown  ,@overallrating  ,@totalrates  ,@ispinned  ,@backgroundcolor  ,@originalfilename  ,@lockeduser  ,@lockedby  ,@lockeddate  ,@lastupdate  ,@blobstatusid  ,@old­Id  ,@bookedby  ,@bookingdate  ,@retry  ,@statemachinestatusid  )    END  GO |

Uses

[[dbo].[t\_tilecopy\_copiedtile]](#z32gliNZ8xOJvfNpb2w8q4TnC08=)

|  |
| --- |
| [dbo].[p\_tilecopy\_insertrelation] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @copyfinishedtimestamp | datetime | 8 |
| @usernamecreator | nvarchar(50) | 100 |
| @sourceprojectid | int | 4 |
| @targetprojectid | int | 4 |
| @tilerelationid | int | 4 |
| @projectid | int | 4 |
| @tileid | uniqueidentifier | 16 |
| @related\_tileid | uniqueidentifier | 16 |
| @tilerelationstyleid | int | 4 |
| @tilerelationanchoridorigin | int | 4 |
| @tilerelationanchoriddestination | int | 4 |
| @tilerelationstartendsymbolorigin | int | 4 |
| @tilerelationstartendsymboldestination | int | 4 |
| @tilerelationlineid | int | 4 |
| @color | nvarchar(7) | 14 |
| @thickness | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_insertrelation]  @token [UNIQUEIDENTIFIER] ,  @copyfinishedtimestamp [DATETIME] ,  @usernamecreator [NVARCHAR](50),  @sourceprojectid [INT] ,  @targetprojectid [INT] ,  @tilerelationid [INT] ,  @projectid [INT] ,  @tileid [UNIQUEIDENTIFIER] ,  @related\_tileid [UNIQUEIDENTIFIER] ,  @tilerelationstyleid [INT] ,  @tilerelationanchoridorigin [INT] ,  @tilerelationanchoriddestination [INT] ,  @tilerelationstartendsymbolorigin [INT] ,  @tilerelationstartendsymboldestination [INT] ,  @tilerelationlineid [INT] ,  @color [NVARCHAR](7) ,  @thickness [INT]  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  INSERT INTO [dbo].[t\_tilecopy\_copiedrelation]  (  [token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tilerelationid]  ,[projectid]  ,[tileid]  ,[related\_tileid]  ,[tilerelationstyleid]  ,[tilerelationanchoridorigin]  ,[tilerelationanchoriddestination]  ,[tilerelationstartendsymbolorigin]  ,[tilerelationstartendsymboldestination]  ,[tilerelationlineid]  ,[color]  ,[thickness]  )  VALUES  (  @token,  @copyfinishedtimestamp,  @usernamecreator,  @sourceprojectid,  @targetprojectid,  @tilerelationid,  @projectid ,  @tileid ,  @related\_tileid ,  @tilerelationstyleid ,  @tilerelationanchoridorigin ,  @tilerelationanchoriddestination ,  @tilerelationstartendsymbolorigin ,  @tilerelationstartendsymboldestination ,  @tilerelationlineid ,  @color ,  @thickness  )  END  GO |

Uses

[[dbo].[t\_tilecopy\_copiedrelation]](#TpEVDYGh4J/aV/9zW2Qgqvq2Ku8=)

|  |
| --- |
| [dbo].[p\_tilecopy\_insertsourcetile] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @token | uniqueidentifier | 16 |
| @copyfinishedtimestamp | datetime | 8 |
| @usernamecreator | nvarchar(50) | 100 |
| @sourceprojectid | int | 4 |
| @targetprojectid | int | 4 |
| @tileid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilecopy\_insertsourcetile]  -- Add the parameters for the stored procedure here  @token [uniqueidentifier],  @copyfinishedtimestamp [datetime],  @usernamecreator [nvarchar](50),  @sourceprojectid [int],  @targetprojectid [int],  @tileid [uniqueidentifier]  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  INSERT INTO [dbo].[t\_tilecopy\_sourcetile]  ([token]  ,[copyfinishedtimestamp]  ,[usernamecreator]  ,[sourceprojectid]  ,[targetprojectid]  ,[tileid])  VALUES  (  @token,  @copyfinishedtimestamp,  @usernamecreator,  @sourceprojectid,  @targetprojectid,  @tileid  )  END  GO |

Uses

[[dbo].[t\_tilecopy\_sourcetile]](#Q2UYJrfAzp+CQwZGWFDTwEseGMI=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_book]  AS BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) dbo.t\_tilestatus\_copy\_booking  SET bookedby=@booked\_by,  bookingdate = @booking\_date,  lastupdate = @booking\_date,  statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('working')  WHERE statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('ready')  IF @@ROWCOUNT > 0  BEGIN  SELECT Id,  username,  sourceprojectid,  destinationprojectid,  tilesandpositions,  tileoffsetx,  tileoffsety,  clientconnectionid,  appver,  uniquedeviceid,  bookedby,  bookingdate,  retry  FROM dbo.t\_tilestatus\_copy\_booking  WHERE bookedby=@booked\_by  END    SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_fallback] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @max\_retries | int | 4 |
| @booking\_elapsed\_time\_in\_sec | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_fallback]  @max\_retries int,  @booking\_elapsed\_time\_in\_sec int  AS  BEGIN  DECLARE @lastupdate DATETIME  SET @lastupdate = GETUTCDATE()    SET NOCOUNT ON  UPDATE dbo.t\_tilestatus\_copy\_booking  SET bookedby=null,  bookingdate = null,  lastupdate = @lastupdate,  retry = retry + 1,  statemachinestatusid = CASE WHEN (retry >= @max\_retries)  THEN dbo.fn\_­Get­State­Machine­Status­Id('ERROR')  ELSE  dbo.fn\_­Get­State­Machine­Status­Id('READY')  END  WHERE statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('WORKING')  AND DATEDIFF(SECOND, bookingdate, GETUTCDATE()) >= @booking\_elapsed\_time\_in\_sec    END  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_history] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_history]  AS  BEGIN  INSERT [dbo].[t\_tilestatus\_copy\_booking\_history]  (  [Id],  [creationdate],  [username],  [sourceprojectid],  [destinationprojectid],  [tilesandpositions],  [tileoffsetx],  [tileoffsety],  [clientconnectionid],  [appver],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  )  SELECT  [Id],  [creationdate],  [username],  [sourceprojectid],  [destinationprojectid],  [tilesandpositions],  [tileoffsetx],  [tileoffsety],  [clientconnectionid],  [appver],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  FROM  (  DELETE [dbo].[t\_tilestatus\_copy\_booking]  OUTPUT  DELETED.[Id],  DELETED.[creationdate],  DELETED.[username],  DELETED.[sourceprojectid],  DELETED.[destinationprojectid],  DELETED.[tilesandpositions],  DELETED.[tileoffsetx],  DELETED.[tileoffsety],  DELETED.[clientconnectionid],  DELETED.[appver],  DELETED.[uniquedeviceid],  DELETED.[bookedby],  DELETED.[lastupdate],  DELETED.[bookingdate],  DELETED.[retry],  DELETED.[statemachinestatusid]  WHERE [statemachinestatusid] >= [dbo].fn\_­Get­State­Machine­Status­Id('FINISHED')  ) AS Rows­To­Move ;  END  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

[[dbo].[t\_tilestatus\_copy\_booking\_history]](#c+tNoP0kvrC9ukkhGNCEvWTioxY=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_history\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @source­Project­Id | int | 4 |  |
| @destination­Project­Id | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_history\_search]  @source­Project­Id int,  @destination­Project­Id int,  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT [Id],  [creationdate],  [username],  [sourceprojectid],  [destinationprojectid],  [tilesandpositions],  [tileoffsetx],  [tileoffsety],  [clientconnectionid],  [appver],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  FROM [dbo].[t\_tilestatus\_copy\_booking\_history] WITH (NOLOCK)  WHERE ((@source­Project­Id IS NULL) OR (sourceprojectid = @source­Project­Id))  AND ((@destination­Project­Id IS NULL) OR (destinationprojectid = @destination­Project­Id))  ORDER BY [id]  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM [dbo].[t\_tilestatus\_copy\_booking\_history] WITH (NOLOCK)  WHERE ((@source­Project­Id IS NULL) OR (sourceprojectid = @source­Project­Id))  AND ((@destination­Project­Id IS NULL) OR (destinationprojectid = @destination­Project­Id))  END ELSE BEGIN  SET @totalrows = 0  END  END  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking\_history]](#c+tNoP0kvrC9ukkhGNCEvWTioxY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(200) | 400 |
| @sourceprojectid | int | 4 |
| @destinationprojectid | int | 4 |
| @tilesandpositions | nvarchar(max) | max |
| @tileoffsetx | float | 8 |
| @tileoffsety | float | 8 |
| @clientconnectionid | nvarchar(50) | 100 |
| @appver | nvarchar(50) | 100 |
| @device­Id | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_insert]  @username NVARCHAR(200),  @sourceprojectid INT,  @destinationprojectid INT,  @tilesandpositions NVARCHAR(MAX),  @tileoffsetx float,  @tileoffsety float,  @clientconnectionid NVARCHAR(50),  @appver NVARCHAR(50),  @device­Id UNIQUEIDENTIFIER  AS  BEGIN  DECLARE @statemachinestatusid INT  SET @statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('ready')    INSERT INTO [t\_tilestatus\_copy\_booking]  (  [username],  [sourceprojectid],  [destinationprojectid],  [tilesandpositions],  [tileoffsetx],  [tileoffsety],  [clientconnectionid],  [appver],  [uniquedeviceid],  [lastupdate],  [bookedby],  [bookingdate],  [retry],  [statemachinestatusid]  )  VALUES  (  @username,  @sourceprojectid,  @destinationprojectid,  @tilesandpositions,  @tileoffsetx,  @tileoffsety,  @clientconnectionid,  @appver,  @device­Id,  GETUTCDATE(),  NULL,  NULL,  0,  @statemachinestatusid  )  END  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_copy\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_copy\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  UPDATE dbo.t\_tilestatus\_copy\_booking  SET  bookedby=NULL,  bookingdate=NULL,  retry=@retries,  statemachinestatusid=@statemachinestatusid,  lastupdate = GETUTCDATE()  WHERE  Id=@id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_copy\_booking]](#RVjh+ATj4D3aPXIlnTnzWVjMatE=)

|  |
| --- |
| [dbo].[p\_tilestatus\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_delete]  @tileid uniqueidentifier,  @rows int = 0 output  AS  --SET NOCOUNT ON  --DELETE  --FROM t\_tilestatus  -- WHERE tileid = @tileid  --set @rows = @@ROWCOUNT  --SET NOCOUNT OFF  --this stored procedure it is not used any more we shall clean up...  SET NOCOUNT ON  BEGIN TRANSACTION;  DELETE  FROM t\_tile\_relation  WHERE tileid = @tileid  DELETE  FROM t\_tile\_relation  WHERE related\_tileid = @tileid  DELETE  FROM t\_tile\_voting  WHERE tileid = @tileid  DELETE  FROM t\_tilestatus  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_relation]](#078YtCFOxZtrTr+ji99XjRUc40A=)

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tileid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_by\_id]  @tileid UNIQUEIDENTIFIER  AS  SET NOCOUNT ON;  SELECT tileid,  projectid,  parentid,  tiletype,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  totalthumbup,  totlathumbdown,  overallrating,  totalrates,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus WITH(NOLOCK)  WHERE tileid = @tileid  RETURN  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_by\_parenttile] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @parent­Id | uniqueidentifier | 16 |
| @tile­Type | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_by\_parenttile]  @parent­Id UNIQUEIDENTIFIER,  @tile­Type INT  AS  SET NOCOUNT ON;  SELECT tileid,  parentid,  projectid,  tiletype,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  totalthumbup,  totlathumbdown,  overallrating,  totalrates,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus WITH(NOLOCK)  WHERE parentid = @parent­Id  AND tiletype = @tile­Type  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_by\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_by\_projectid]  @projectid int  AS  SET NOCOUNT ON;  SELECT tileid,  parentid,  tiletype,  projectid,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  totalthumbup,  totlathumbdown,  overallrating,  totalrates,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus with(nolock)  WHERE projectid = @projectid  AND blobstatusid < dbo.fn\_­Get­Blob­Status('Upload­Error')  AND tiletype <> dbo.fn\_­Get­Tile­Type('Thumbnail')  RETURN  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[[dbo].[fn\_­Get­Tile­Type]](#fGE7XnuGzg29WlhvQEuR0BLPbL8=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_by\_projectid\_paged] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @tocount | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_by\_projectid\_paged]  @projectid int,  @pagesize int,  @pagenumber int,  @tocount bit,  @totalrows int = 0 output  AS  SET NOCOUNT ON;  SELECT tileid,  parentid,  tiletype,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  totalthumbup,  totlathumbdown,  overallrating,  totalrates,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus with(nolock)  WHERE projectid = @projectid  AND blobstatusid < dbo.fn\_­Get­Blob­Status('Upload­Error')  AND tiletype <> dbo.fn\_­Get­Tile­Type('Thumbnail')  ORDER BY tileid  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF(@tocount = 1)  BEGIN  SELECT @totalrows = count(1)  FROM t\_tilestatus with(nolock)  WHERE projectid = @projectid  AND blobstatusid < dbo.fn\_­Get­Blob­Status('Upload­Error')  AND tiletype <> dbo.fn\_­Get­Tile­Type('Thumbnail')  END  RETURN  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[[dbo].[fn\_­Get­Tile­Type]](#fGE7XnuGzg29WlhvQEuR0BLPbL8=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_by\_projectid\_with\_thumbnails] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_by\_projectid\_with\_thumbnails]  @projectid INT  AS  SET NOCOUNT ON;  SELECT tileid,  parentid,  tiletype,  projectid,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  totalthumbup,  totlathumbdown,  overallrating,  totalrates,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus WITH(NOLOCK)  WHERE projectid = @projectid  AND blobstatusid < dbo.fn\_­Get­Blob­Status('Upload­Error')  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_max\_zindex] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @project­Id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_max\_zindex]  @project­Id INT  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 MAX(zindex) FROM dbo.t\_tilestatus  WHERE projectid=@project­Id  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @parent­Id | uniqueidentifier | 16 |  |
| @start | int | 4 |  |
| @end | int | 4 |  |
| @total­Thumbnails | int | 4 | Out |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Dennis Vroegop  -- Create date: 04/05/2020  -- Description: Returns a subset of the thumbnails for a tile  -- =============================================  CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile]  -- Add the parameters for the stored procedure here  @parent­Id UNIQUEIDENTIFIER,  @start INT = 0,  @end INT = 0,  @total­Thumbnails INT OUTPUT  AS  BEGIN  SET NOCOUNT ON  SELECT tileid, JSON\_­VALUE(tilecontent, '$.Sequence­Number') [index] FROM dbo.t\_tilestatus  WHERE parentid = @parent­Id  AND JSON\_­VALUE(tilecontent, '$.Sequence­Number') >= @start  AND JSON\_­VALUE(tilecontent, '$.Sequence­Number') <= @end  AND tiletype = [dbo].fn\_­Get­Tile­Type('Thumbnail')  ORDER BY JSON\_­VALUE(tilecontent, '$.Sequence­Number') ASC  SET NOCOUNT OFF  SELECT @total­Thumbnails = COUNT(\*) FROM dbo.t\_tilestatus WITH (NOLOCK) WHERE parentid = @parent­Id AND blobstatusid <> [dbo].fn\_­Get­Blob­Status('Deleted')  RETURN 0  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[[dbo].[fn\_­Get­Tile­Type]](#fGE7XnuGzg29WlhvQEuR0BLPbL8=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_toplevel\_by\_projectid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_toplevel\_by\_projectid]  @projectid INT  AS  BEGIN  SET NOCOUNT ON  SELECT  tileid  FROM  dbo.t\_tilestatus WITH (NOLOCK)  WHERE  projectid = @projectid  AND parentid IS NULL  RETURN 0  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_get\_upgradable\_for\_project] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @projectid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_get\_upgradable\_for\_project]  @projectid INT  AS  BEGIN  SELECT tileid,  tiletype,  tilecontent,  originalfilename FROM dbo.t\_tilestatus WITH (NOLOCK)  WHERE  projectid=@projectid  AND  blobstatusid <> dbo.fn\_­Get­Blob­Status('deleted')  AND  (  tiletype = dbo.fn\_­Get­Tile­Type('Image')  OR  tiletype = dbo.fn\_­Get­Tile­Type('PDF')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Video')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Audio')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Word')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Excel')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Power­Point')  OR  tiletype = dbo.fn\_­Get­Tile­Type('Thumbnail')  )  END  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[[dbo].[fn\_­Get­Tile­Type]](#fGE7XnuGzg29WlhvQEuR0BLPbL8=)

|  |
| --- |
| [dbo].[p\_tilestatus\_gettree\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @requested­Ids | Tile­Status­Id­Set | max |
| @include­Deleted | bit | 1 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: Dennis Vroegop  -- Create Date: 07/05/2020  -- Description: Returns a tile and it's children  -- =============================================  CREATE PROCEDURE [dbo].[p\_tilestatus\_gettree\_by\_id]  (  -- Add the parameters for the stored procedure here  @requested­Ids [dbo].Tile­Status­Id­Set READONLY ,  @include­Deleted BIT  )  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON  DECLARE @project­Id INT  SELECT @project­Id = projectid FROM dbo.t\_tilestatus WHERE tileid = (SELECT TOP(1) Tile­Id FROM @requested­Ids)  ;WITH tree AS (  SELECT  t1.tileid,  t1.projectid,  t1.parentid,  t1.tiletype,  t1.tilecontent,  t1.positionx,  t1.positiony,  t1.height,  t1.width,  t1.zindex,  t1.scalex,  t1.scaley,  t1.rotation,  t1.totalthumbup,  t1.totlathumbdown,  t1.overallrating,  t1.totalrates,  t1.ispinned,  t1.backgroundcolor,  t1.lockeduser,  t1.lockedby,  t1.lockeddate,  t1.lastupdate,  t1.blobstatusid,  t1.bookedby,  t1.bookingdate,  t1.retry,  t1.statemachinestatusid ,  t1.originalfilename,  [level] = 1  FROM dbo.t\_tilestatus t1  WHERE t1.parentid IS NULL  AND t1.projectid = @project­Id  AND EXISTS  (SELECT Tile­Id FROM @requested­Ids WHERE t1.tileid = Tile­Id)  UNION ALL  SELECT  t2.tileid,  t2.projectid,  t2.parentid,  t2.tiletype,  t2.tilecontent,  t2.positionx,  t2.positiony,  t2.height,  t2.width,  t2.zindex,  t2.scalex,  t2.scaley,  t2.rotation,  t2.totalthumbup,  t2.totlathumbdown,  t2.overallrating,  t2.totalrates,  t2.ispinned,  t2.backgroundcolor,  t2.lockeduser,  t2.lockedby,  t2.lockeddate,  t2.lastupdate,  t2.blobstatusid,  t2.bookedby,  t2.bookingdate,  t2.retry,  t2.statemachinestatusid,  t2.originalfilename,  [level] = tree.[level] +1  FROM dbo.t\_tilestatus t2 INNER JOIN  tree ON tree.tileid = t2.parentid  WHERE t2.projectid = @project­Id  )  -- Check to see if we need to include the deleted ones as well or not.  SELECT  tree.tileid,  tree.projectid,  tree.parentid,  tree.tiletype,  tree.tilecontent,  tree.positionx,  tree.positiony,  tree.height,  tree.width,  tree.zindex,  tree.scalex,  tree.scaley,  tree.rotation,  tree.totalthumbup,  tree.totlathumbdown,  tree.overallrating,  tree.totalrates,  tree.ispinned,  tree.backgroundcolor,  tree.lockeduser,  tree.lockedby,  tree.lockeddate,  tree.lastupdate,  tree.blobstatusid,  tree.bookedby,  tree.bookingdate,  tree.retry,  tree.statemachinestatusid,  tree.originalfilename  FROM tree  WHERE  @include­Deleted = 1  OR (@include­Deleted = 0 AND tree.blobstatusid <> dbo.fn\_­Get­Blob­Status('deleted'))  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[dbo].[Tile­Status­Id­Set]

|  |
| --- |
| [dbo].[p\_tilestatus\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @projectid | int | 4 |  |
| @parentid | uniqueidentifier | 16 |  |
| @tiletype | int | 4 |  |
| @tilecontent | nvarchar(max) | max |  |
| @positionx | float | 8 |  |
| @positiony | float | 8 |  |
| @height | float | 8 |  |
| @width | float | 8 |  |
| @zindex | int | 4 |  |
| @scalex | float | 8 |  |
| @scaley | float | 8 |  |
| @rotation | float | 8 |  |
| @ispinned | bit | 1 |  |
| @backgroundcolor | nvarchar(7) | 14 |  |
| @blobstatusid | int | 4 |  |
| @lockeduser | nvarchar(50) | 100 |  |
| @lockedby | nvarchar(50) | 100 |  |
| @lockeddate | datetime | 8 |  |
| @originalfilename | nvarchar(255) | 510 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_insert]  @tileid uniqueidentifier,  @projectid int,  @parentid uniqueidentifier,  @tiletype int,  @tilecontent nvarchar(max),  @positionx float,  @positiony float,  @height float,  @width float,  @zindex int,  @scalex float,  @scaley float,  @rotation float,  @ispinned bit,  @backgroundcolor nvarchar(7),  @blobstatusid int,  @lockeduser nvarchar(50),  @lockedby nvarchar(50),  @lockeddate datetime,  @originalfilename NVARCHAR(255),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  declare @RET\_­VAL INT  SET @RET\_­VAL = 0  SET NOCOUNT ON  SET @rows = 0  BEGIN  IF NOT EXISTS ( SELECT 1  FROM t\_tilestatus  WHERE tileid = @tileid)  BEGIN  INSERT INTO t\_tilestatus  (  tileid,  projectid,  parentid,  tiletype,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  lockeduser,  lockedby,  lockeddate,  lastupdate,  originalfilename  )  --OUTPUT inserted.projectid  VALUES  (  @tileid,  @projectid,  @parentid,  @tiletype,  @tilecontent,  @positionx,  @positiony,  @height,  @width,  @zindex,  @scalex,  @scaley,  @rotation,  @ispinned,  @backgroundcolor,  @blobstatusid,  @lockeduser,  @lockedby,  @lockeddate,  @lastupdate,  @originalfilename  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    END  ELSE  BEGIN  SET @RET\_­VAL = -1  END  END  RETURN @RET\_­VAL  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_stroke\_by\_tileid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tileid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_stroke\_by\_tileid]  @tileid uniqueidentifier  AS  SET NOCOUNT ON;  SELECT tilestatusstrokeid,  stroke  FROM t\_tilestatus\_stroke with(nolock)  WHERE tileid = @tileid  RETURN  GO |

Uses

[[dbo].[t\_tilestatus\_stroke]](#eNq8iC4zx70GGM6EXbHSL++uzlU=)

|  |
| --- |
| [dbo].[p\_tilestatus\_stroke\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tilestatusstrokeid | uniqueidentifier | 16 |  |
| @tileid | uniqueidentifier | 16 |  |
| @stroke | varbinary(max) | max |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_stroke\_insert]  @tilestatusstrokeid uniqueidentifier,  @tileid uniqueidentifier,  @stroke varbinary(max),  @rows int = 0 output  AS  SET NOCOUNT ON  SET @rows = 0  INSERT INTO t\_tilestatus\_stroke  (  tilestatusstrokeid,  tileid,  stroke  )  VALUES  (  @tilestatusstrokeid,  @tileid,  @stroke  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_stroke]](#eNq8iC4zx70GGM6EXbHSL++uzlU=)

|  |
| --- |
| [dbo].[p\_tilestatus\_stroke\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tilestatusstrokeid | uniqueidentifier | 16 |  |
| @tileid | uniqueidentifier | 16 |  |
| @stroke | varbinary(max) | max |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_stroke\_update]  @tilestatusstrokeid uniqueidentifier,  @tileid uniqueidentifier,  @stroke varbinary(max),  @rows int = 0 output  AS  SET NOCOUNT ON  UPDATE t\_tilestatus\_stroke  SET stroke = @stroke  WHERE tilestatusstrokeid = @tilestatusstrokeid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_stroke]](#eNq8iC4zx70GGM6EXbHSL++uzlU=)

|  |
| --- |
| [dbo].[p\_tilestatus\_unlock\_by\_ids] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @ids | Tile­Status­Id­Set | max |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_unlock\_by\_ids]  -- Add the parameters for the stored procedure here  @ids [dbo].Tile­Status­Id­Set READONLY  AS  BEGIN  -- SET NOCOUNT ON added to prevent extra result sets from  -- interfering with SELECT statements.  SET NOCOUNT ON;  -- Insert statements for procedure here  UPDATE dbo.t\_tilestatus  SET lockeduser=NULL,  lockedby=NULL,  lockeddate= NULL  FROM dbo.t\_tilestatus AS ts  INNER JOIN @ids AS p ON ts.tileid = p.Tile­Id  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[dbo].[Tile­Status­Id­Set]

|  |
| --- |
| [dbo].[p\_tilestatus\_unlock\_by\_lockedby] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @lockedby | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_unlock\_by\_lockedby]  @lockedby nvarchar(50)  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET lockeduser = null,  lockedby = null,  lockeddate = null,  lastupdate = @lastupdate  OUTPUT INSERTED.tileid  WHERE lockedby = @lockedby  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_unlock\_by\_project\_lockedby] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @lockedby | nvarchar(50) | 100 |  |
| @projectid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_unlock\_by\_project\_lockedby]  @lockedby nvarchar(50),  @projectid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SELECT tileid,  parentid,  tiletype,  tilecontent,  positionx,  positiony,  height,  width,  zindex,  scalex,  scaley,  rotation,  ispinned,  backgroundcolor,  blobstatusid,  lockeduser,  lockedby,  lockeddate,  lastupdate,  bookedby,  bookingdate,  retry,  statemachinestatusid,  originalfilename  FROM t\_tilestatus  WHERE projectid = @projectid  AND lockedby = @lockedby  SET NOCOUNT ON  UPDATE t\_tilestatus  SET lockeduser = null,  lockedby = null,  lockeddate = null,  lastupdate = @lastupdate  WHERE projectid = @projectid  AND lockedby = @lockedby  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_unlock\_forgot] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_unlock\_forgot]  @rows int = 0 output  AS  declare @lastupdate smalldatetime  set @lastupdate = GETUTCDATE()    UPDATE t\_tilestatus  SET lastupdate = @lastupdate,  lockeduser = null,  lockedby = null,  lockeddate = null  WHERE lockedby is not null  AND lockedby not in  (  SELECT connectionid FROM t\_online\_user with (nolock)  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @projectid | int | 4 |  |
| @parentid | uniqueidentifier | 16 |  |
| @tiletype | int | 4 |  |
| @tilecontent | nvarchar(max) | max |  |
| @positionx | float | 8 |  |
| @positiony | float | 8 |  |
| @height | float | 8 |  |
| @width | float | 8 |  |
| @zindex | int | 4 |  |
| @scalex | float | 8 |  |
| @scaley | float | 8 |  |
| @rotation | float | 8 |  |
| @ispinned | bit | 1 |  |
| @backgroundcolor | nvarchar(7) | 14 |  |
| @blobstatusid | int | 4 |  |
| @lockeduser | nvarchar(50) | 100 |  |
| @lockedby | nvarchar(50) | 100 |  |
| @lockeddate | datetime | 8 |  |
| @connectionid | nvarchar(50) | 100 |  |
| @originalfilename | nvarchar(255) | 510 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_update]  @tileid uniqueidentifier,  @projectid int,  @parentid uniqueidentifier,  @tiletype int,  @tilecontent nvarchar(max),  @positionx float,  @positiony float,  @height float,  @width float,  @zindex int,  @scalex float,  @scaley float,  @rotation float,  @ispinned bit,  @backgroundcolor nvarchar(7),  @blobstatusid int,  @lockeduser nvarchar(50),  @lockedby nvarchar(50),  @lockeddate datetime,  @connectionid nvarchar(50),  @originalfilename NVARCHAR(255),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET parentid = @parentid,  tiletype = @tiletype,  tilecontent = @tilecontent,  positionx = @positionx,  positiony = @positiony,  height = @height,  width = @width,  zindex = @zindex,  scalex = @scalex,  scaley = @scaley,  rotation = @rotation,  ispinned = @ispinned,  backgroundcolor = @backgroundcolor,  --IMPORTANT blobstatus never go back  blobstatusid = case when @blobstatusid < blobstatusid then blobstatusid else @blobstatusid end,  lockeduser = @lockeduser,  lockedby = @lockedby,  lockeddate = @lockeddate,  lastupdate = @lastupdate,  originalfilename = @originalfilename  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  UPDATE t\_online\_user  SET lastupdate = @lastupdate  WHERE connectionid = @connectionid  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_blobstatusid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @blobstatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_update\_blobstatusid]  @tileid uniqueidentifier,  @blobstatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  declare @statemachinestatusid int  if(@blobstatusid = dbo.fn\_­Get­Blob­Status('UPLOADERROR') or @blobstatusid = dbo.fn\_­Get­Blob­Status('DELETED'))  begin  set @statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('READY')  end  else  begin  set @statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('NOT READY')  end  SET NOCOUNT ON  UPDATE t\_tilestatus  SET --IMPORTANT blobstatus never go back  blobstatusid = case when @blobstatusid < blobstatusid then blobstatusid else @blobstatusid end,  lastupdate = @lastupdate,  statemachinestatusid = @statemachinestatusid  WHERE tileid = @tileid  set @rows = @@ROWCOUNT    -- second time for updating parent  UPDATE t\_tilestatus  SET --IMPORTANT blobstatus never go back  blobstatusid = case when @blobstatusid < blobstatusid then blobstatusid else @blobstatusid end,  lastupdate = @lastupdate,  statemachinestatusid = @statemachinestatusid  WHERE parentid = @tileid  IF(@blobstatusid = dbo.fn\_­Get­Blob­Status('DELETED'))  BEGIN  DELETE  FROM t\_tile\_voting  WHERE tileid = @tileid  END    SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tile\_voting]](#apFAn+5wzEjWRUBbGPwi/lWybWQ=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

[[dbo].[fn\_­Get­Blob­Status]](#MrHRuJLPZf16ktgVr1fQoxOAtko=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_containertype] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @tiletype | int | 4 |  |
| @positionx | float | 8 |  |
| @positiony | float | 8 |  |
| @height | float | 8 |  |
| @width | float | 8 |  |
| @zindex | int | 4 |  |
| @scalex | float | 8 |  |
| @scaley | float | 8 |  |
| @rotation | float | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_update\_containertype]  @tileid uniqueidentifier,  @tiletype int,  @positionx float,  @positiony float,  @height float,  @width float,  @zindex int,  @scalex float,  @scaley float,  @rotation float,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET tiletype = @tiletype,  positionx = @positionx,  positiony = @positiony,  height = @height,  width = @width,  zindex = @zindex,  scalex = @scalex,  scaley = @scaley,  rotation = @rotation,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_content] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @tilecontent | nvarchar(max) | max |  |
| @backgroundcolor | nvarchar(7) | 14 |  |
| @connectionid | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_update\_content]  @tileid uniqueidentifier,  @tilecontent nvarchar(max),  @backgroundcolor nvarchar(7),  @connectionid nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET tilecontent = @tilecontent,  backgroundcolor = @backgroundcolor,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  UPDATE t\_online\_user  SET lastupdate = @lastupdate  WHERE connectionid = @connectionid  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_grouped] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @parentid | uniqueidentifier | 16 |  |
| @positionx | float | 8 |  |
| @positiony | float | 8 |  |
| @height | float | 8 |  |
| @width | float | 8 |  |
| @zindex | int | 4 |  |
| @scalex | float | 8 |  |
| @scaley | float | 8 |  |
| @rotation | float | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_update\_grouped]  @tileid uniqueidentifier,  @parentid uniqueidentifier,  @positionx float,  @positiony float,  @height float,  @width float,  @zindex int,  @scalex float,  @scaley float,  @rotation float,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET parentid = @parentid,  positionx = @positionx,  positiony = @positiony,  height = @height,  width = @width,  zindex = @zindex,  scalex = @scalex,  scaley = @scaley,  rotation = @rotation,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_lock] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @lockeduser | nvarchar(50) | 100 |  |
| @lockedby | nvarchar(50) | 100 |  |
| @lockeddate | datetime | 8 |  |
| @connectionid | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_update\_lock]  @tileid uniqueidentifier,  @lockeduser nvarchar(50),  @lockedby nvarchar(50),  @lockeddate datetime,  @connectionid nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET lockeduser = @lockeduser,  lockedby = @lockedby,  lockeddate = @lockeddate,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  UPDATE t\_online\_user  SET lastupdate = @lastupdate  WHERE connectionid = @connectionid  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_pinning] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @ispinned | bit | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_update\_pinning]  @tileid uniqueidentifier,  @ispinned bit,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET ispinned = @ispinned,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_position] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @positionx | float | 8 |  |
| @positiony | float | 8 |  |
| @height | float | 8 |  |
| @width | float | 8 |  |
| @zindex | int | 4 |  |
| @scalex | float | 8 |  |
| @scaley | float | 8 |  |
| @rotation | float | 8 |  |
| @connectionid | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_update\_position]  @tileid uniqueidentifier,  @positionx float,  @positiony float,  @height float,  @width float,  @zindex int,  @scalex float,  @scaley float,  @rotation float,  @connectionid nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET positionx = @positionx,  positiony = @positiony,  height = @height,  width = @width,  zindex = @zindex,  scalex = @scalex,  scaley = @scaley,  rotation = @rotation,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  UPDATE t\_online\_user  SET lastupdate = @lastupdate  WHERE connectionid = @connectionid  RETURN 0  GO |

Uses

[[dbo].[t\_online\_user]](#oTKipXz9q0i7/ho1w/MFw/f36nA=)

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_voting] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @totalthumbup | int | 4 |  |
| @totlathumbdown | int | 4 |  |
| @overallrating | decimal(2,1) | 5 |  |
| @totalrates | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create procedure [dbo].[p\_tilestatus\_update\_voting]  @tileid uniqueidentifier,  @totalthumbup int,  @totlathumbdown int,  @overallrating decimal(2,1),  @totalrates int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE t\_tilestatus  SET totalthumbup = @totalthumbup,  totlathumbdown = @totlathumbdown,  overallrating = @overallrating,  totalrates = @totalrates,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_update\_zindex] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @zindex | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_update\_zindex]  @tileid uniqueidentifier,  @zindex int,  @rows int = 0 output  AS    SET NOCOUNT ON  UPDATE t\_tilestatus  SET zindex = @zindex  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_upgrade\_book] |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_upgrade\_book]  AS BEGIN  DECLARE @booked\_by UNIQUEIDENTIFIER  SET @booked\_by = NEWID()    DECLARE @booking\_date DATETIME  SET @booking\_date = GETUTCDATE()    SET NOCOUNT ON  UPDATE TOP(1) dbo.t\_tilestatus\_upgrade\_booking  SET bookedby=@booked\_by,  bookingdate = @booking\_date,  lastupdate = @booking\_date,  statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('working')  WHERE statemachinestatusid = [dbo].fn\_­Get­State­Machine­Status­Id('ready')  IF @@ROWCOUNT > 0  BEGIN  SELECT [Id],  [creationdate],  [username],  [projectupgradeid]  [tileid],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry]  FROM dbo.t\_tilestatus\_upgrade\_booking  WHERE bookedby=@booked\_by  END    SET NOCOUNT OFF  END  GO |

Uses

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_upgrade\_file] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @tileid | uniqueidentifier | 16 |  |
| @tilecontent | nvarchar(max) | max |  |
| @originalfilename | nvarchar(255) | 510 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| create PROCEDURE [dbo].[p\_tilestatus\_upgrade\_file]  @tileid uniqueidentifier,  @tilecontent nvarchar(max),  @originalfilename NVARCHAR(255),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE dbo.t\_tilestatus  SET tilecontent = @tilecontent,  originalfilename = @originalfilename,  lastupdate = @lastupdate  WHERE tileid = @tileid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_tilestatus\_upgrade\_insert] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(200) | 400 |
| @projectupgradeid | int | 4 |
| @tileid | uniqueidentifier | 16 |
| @uniquedeviceid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_upgrade\_insert]  @username NVARCHAR(200),  @projectupgradeid INT,  @tileid UNIQUEIDENTIFIER,  @uniquedeviceid UNIQUEIDENTIFIER    AS  BEGIN  DECLARE @statemachinestatusid INT  SET @statemachinestatusid = dbo.fn\_­Get­State­Machine­Status­Id('ready')  INSERT INTO [dbo].[t\_tilestatus\_upgrade\_booking]  (  [creationdate],  [username],  [projectupgradeid],  [tileid],  [uniquedeviceid],  [bookedby],  [lastupdate],  [bookingdate],  [retry],  [statemachinestatusid]  )  VALUES  (  GETUTCDATE(),  @username,  @projectupgradeid,  @tileid,  @uniquedeviceid,  NULL,  GETUTCDATE(),  NULL,  0,  @statemachinestatusid  )  END  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

[[dbo].[fn\_­Get­State­Machine­Status­Id]](#OR1i7a+8r+s6UWatraats0CjTwY=)

|  |
| --- |
| [dbo].[p\_tilestatus\_upgrade\_unbook] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |
| @retries | int | 4 |
| @statemachinestatusid | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_tilestatus\_upgrade\_unbook]  @id INT,  @retries INT,  @statemachinestatusid INT  AS  SET NOCOUNT ON  UPDATE dbo.t\_tilestatus\_upgrade\_booking  SET  bookedby=NULL,  bookingdate=NULL,  retry=@retries,  statemachinestatusid=@statemachinestatusid,  lastupdate = GETUTCDATE()  WHERE  Id=@id  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[dbo].[t\_tilestatus\_upgrade\_booking]](#NfeAlTE7NimcNyx/T33qC/risAg=)

|  |
| --- |
| [dbo].[p\_update\_blobstatus\_for\_thumbnails] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tile­Id | uniqueidentifier | 16 |
| @blobstatus | int | 4 |

SQL Script

|  |
| --- |
| -- Stored Procedure  -- =============================================  -- Author: Dennis Vroegop  -- Create date: 02 04 2020  -- Description: Updates the blob status only for the thumbnail creation  -- =============================================  CREATE PROCEDURE [dbo].[p\_update\_blobstatus\_for\_thumbnails]  @tile­Id uniqueidentifier,  @blobstatus int = 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  UPDATE dbo.t\_tilestatus  SET blobstatusid = @blobstatus  WHERE tileid= @tile­Id  END  GO |

Uses

[[dbo].[t\_tilestatus]](#oLs0t3srd6yHmujlsbYzAtzC4Zs=)

|  |
| --- |
| [dbo].[p\_user\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT userid, username, firstname, lastname, photourl, is\_guest, creationdate, lastupdate, lastactivitydate  FROM t\_user with (nolock)  ORDER BY userid  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_user with (nolock)  RETURN 0  END  GO |

Uses

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_get\_by\_project] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @is\_guest | bit | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_get\_by\_project]  @projectid int,  @is\_guest bit,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT u.userid, u.username, u.firstname, u.lastname, u.photourl, u.is\_guest, u.creationdate, u.lastupdate, u.lastactivitydate  FROM t\_user as u WITH (NOLOCK)  LEFT JOIN t\_project\_participant as p WITH (NOLOCK) on p.userid = u.userid  WHERE p.projectid = @projectid  AND ((@is\_guest is null) or (u.is\_guest = @is\_guest))  ORDER BY projectid desc  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM t\_user as u WITH (NOLOCK)  LEFT JOIN t\_project\_participant as p WITH (NOLOCK) on p.userid = u.userid  WHERE p.projectid = @projectid  AND ((@is\_guest is null) or (u.is\_guest = @is\_guest))  RETURN  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_get\_by\_project\_count] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @projectid | int | 4 |  |
| @permission | tinyint | 1 |  |
| @is\_guest | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_get\_by\_project\_count]  @projectid int,  @permission tinyint,  @is\_guest bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON;  SELECT @totalrows = count(1)  FROM t\_user as u WITH (NOLOCK)  LEFT JOIN t\_project\_participant as pp WITH (NOLOCK) on pp.userid = u.userid  WHERE pp.projectid = @projectid  AND ((@permission is null) or (pp.permission = @permission))  AND ((@is\_guest is null) or (u.is\_guest = @is\_guest))  RETURN  END  GO |

Uses

[[dbo].[t\_project\_participant]](#20iVIf+vwX3+XobN2ZpMOACBq4o=)

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_get\_by\_userid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @username | nvarchar(250) | 500 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_get\_by\_userid]  @userid INT,  @username NVARCHAR (250) OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @username = username  FROM t\_user WITH (NOLOCK)  WHERE userid = @userid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_get\_by\_username] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @input\_username | nvarchar(250) | 500 |  |
| @userid | int | 4 | Out |
| @username | nvarchar(250) | 500 | Out |
| @firstname | nvarchar(50) | 100 | Out |
| @lastname | nvarchar(50) | 100 | Out |
| @photourl | nvarchar(250) | 500 | Out |
| @companyname | nvarchar(250) | 500 | Out |
| @phonenumber | nvarchar(50) | 100 | Out |
| @language | nvarchar(50) | 100 | Out |
| @country | nvarchar(5) | 10 | Out |
| @companysize | tinyint | 1 | Out |
| @companyindustry | tinyint | 1 | Out |
| @companyrole | nvarchar(250) | 500 | Out |
| @usecase | nvarchar(100) | 200 | Out |
| @terms\_accepted | datetime | 8 | Out |
| @receive\_news | bit | 1 | Out |
| @is\_guest | bit | 1 | Out |
| @creationdate | datetime | 8 | Out |
| @lastupdate | datetime | 8 | Out |
| @lastactivitydate | datetime | 8 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_get\_by\_username]  @input\_username NVARCHAR (250),  @userid INT = 0 OUTPUT,  @username NVARCHAR (250) OUTPUT,  @firstname NVARCHAR (50) OUTPUT,  @lastname NVARCHAR (50) OUTPUT,  @photourl NVARCHAR (250) OUTPUT,  @companyname NVARCHAR (250) OUTPUT,  @phonenumber NVARCHAR (50) OUTPUT,  @language NVARCHAR (50) OUTPUT,  @country NVARCHAR (5) OUTPUT,  @companysize TINYINT OUTPUT,  @companyindustry TINYINT OUTPUT,  @companyrole NVARCHAR (250) OUTPUT,  @usecase NVARCHAR (100) OUTPUT,  @terms\_accepted DATETIME OUTPUT,  @receive\_news BIT OUTPUT,  @is\_guest BIT OUTPUT,  @creationdate DATETIME OUTPUT,  @lastupdate DATETIME OUTPUT,  @lastactivitydate DATETIME OUTPUT,  @rows INT = 0 OUTPUT  AS BEGIN  SET NOCOUNT ON  SELECT @userid = userid,  @username = username,  @firstname = firstname,  @lastname = lastname,  @photourl = photourl,  @companyname = companyname,  @phonenumber = phonenumber,  @language = [language],  @country = country,  @companysize = companysize,  @companyindustry = companyindustry,  @companyrole = companyrole,  @usecase = usecase,  @terms\_accepted = terms\_accepted,  @receive\_news = receive\_news,  @is\_guest = is\_guest,  @creationdate = creationdate,  @lastupdate = lastupdate,  @lastactivitydate = lastactivitydate  FROM t\_user WITH (NOLOCK)  WHERE username = @input\_username  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @firstname | nvarchar(50) | 100 |  |
| @lastname | nvarchar(50) | 100 |  |
| @photourl | nvarchar(250) | 500 |  |
| @companyname | nvarchar(250) | 500 |  |
| @phonenumber | nvarchar(50) | 100 |  |
| @language | nvarchar(50) | 100 |  |
| @country | nvarchar(5) | 10 |  |
| @companysize | tinyint | 1 |  |
| @companyindustry | tinyint | 1 |  |
| @companyrole | nvarchar(250) | 500 |  |
| @usecase | nvarchar(100) | 200 |  |
| @is\_guest | bit | 1 |  |
| @terms\_accepted | datetime | 8 |  |
| @receive\_news | bit | 1 |  |
| @userid | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_insert]  @username NVARCHAR(250),  @firstname NVARCHAR(50),  @lastname NVARCHAR(50),  @photourl NVARCHAR(250),  @companyname NVARCHAR (250),  @phonenumber NVARCHAR (50),  @language NVARCHAR (50),  @country NVARCHAR (5),  @companysize TINYINT,  @companyindustry TINYINT,  @companyrole NVARCHAR (250),  @usecase NVARCHAR (100),  @is\_guest BIT,  @terms\_accepted DATETIME,  @receive\_news BIT,  @userid int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    INSERT INTO t\_user  (  username,  firstname,  lastname,  photourl,  companyname,  phonenumber,  [language],  country,  companysize,  companyindustry,  companyrole,  usecase,  terms\_accepted,  receive\_news,  is\_guest,  creationdate  )  VALUES  (  @username,  @firstname,  @lastname,  @photourl,  @companyname,  @phonenumber,  @language,  @country,  @companysize,  @companyindustry,  @companyrole,  @usecase,  @terms\_accepted,  @receive\_news,  @is\_guest,  GETUTCDATE()  )  set @userid = @@IDENTITY  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [dbo].[p\_user\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @userid | int | 4 |  |
| @username | nvarchar(250) | 500 |  |
| @firstname | nvarchar(50) | 100 |  |
| @lastname | nvarchar(50) | 100 |  |
| @photourl | nvarchar(250) | 500 |  |
| @companyname | nvarchar(250) | 500 |  |
| @phonenumber | nvarchar(50) | 100 |  |
| @language | nvarchar(50) | 100 |  |
| @country | nvarchar(5) | 10 |  |
| @companysize | tinyint | 1 |  |
| @companyindustry | tinyint | 1 |  |
| @companyrole | nvarchar(250) | 500 |  |
| @usecase | nvarchar(100) | 200 |  |
| @terms\_accepted | datetime | 8 |  |
| @receive\_news | bit | 1 |  |
| @is\_guest | bit | 1 |  |
| @lastactivitydate | datetime | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [dbo].[p\_user\_update]  @userid int,  @username nvarchar(250),  @firstname nvarchar(50),  @lastname nvarchar(50),  @photourl nvarchar(250),  @companyname NVARCHAR (250),  @phonenumber NVARCHAR (50),  @language NVARCHAR (50),  @country NVARCHAR (5),  @companysize TINYINT,  @companyindustry TINYINT,  @companyrole NVARCHAR (250),  @usecase NVARCHAR (100),  @terms\_accepted datetime,  @receive\_news bit,  @is\_guest BIT,  @lastactivitydate datetime,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON    UPDATE t\_user  SET username = @username,  firstname = @firstname,  lastname = @lastname,  photourl = @photourl,  companyname = @companyname,  phonenumber = @phonenumber,  [language] = @language,  country = @country,  companysize = @companysize,  companyindustry = @companyindustry,  companyrole = @companyrole,  usecase = @usecase,  terms\_accepted = COALESCE(terms\_accepted, @terms\_accepted),  receive\_news = @receive\_news,  is\_guest = @is\_guest,  lastactivitydate = @lastactivitydate,  lastupdate = GETUTCDATE()  WHERE userid = @userid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF    RETURN 0  END  GO |

Uses

[[dbo].[t\_user]](#YQmJ42veNpoHVYwizgZ/g+LxtRs=)

|  |
| --- |
| [licensing].[p\_billing\_type\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_billing\_type\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_billing\_type]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[licensing]

|  |
| --- |
| [licensing].[p\_billing\_type\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_billing\_type\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_billing\_type] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_billing\_type] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[licensing]

|  |
| --- |
| [licensing].[p\_billing\_type\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_billing\_type\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_billing\_type] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[licensing]

|  |
| --- |
| [licensing].[p\_billing\_type\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(250) | 500 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_billing\_type\_insert]  @name nvarchar(250),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_billing\_type] (  [name]  )  VALUES (  @name  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[licensing]

|  |
| --- |
| [licensing].[p\_billing\_type\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_billing\_type\_update]  @id int,  @name nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_billing\_type] SET  [name] = @name  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_billing\_type]](#h41ZR/NYzHzoN/1iCLZE78KXY2Q=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_customer]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [username]  FROM [licensing].[t\_customer] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_customer] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [username]  FROM [licensing].[t\_customer] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_get\_by\_username] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @username | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_get\_by\_username]  @username nvarchar(250)  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [username]  FROM [licensing].[t\_customer] WITH(NOLOCK)  WHERE [username] = @username  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(500) | 1000 |  |
| @username | nvarchar(250) | 500 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_insert]  @name nvarchar(500),  @username nvarchar(250),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_customer] (  [name],  [username]  )  VALUES (  @name,  @username  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name­Filter | nvarchar(500) | 1000 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_search]  @name­Filter nvarchar(500),  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [username]  FROM [licensing].[t\_customer] WITH(NOLOCK)  WHERE ((@name­Filter is null) or ([name] like @name­Filter))  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_customer] WITH(NOLOCK)  WHERE ((@name­Filter is null) or ([name] like @name­Filter))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_customer\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(500) | 1000 |  |
| @username | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_customer\_update]  @id int,  @name nvarchar(500),  @username nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_customer] SET  [name] = @name,  [username] = @username  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[licensing]

|  |
| --- |
| [licensing].[p\_import\_data\_book] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @type | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_import\_data\_book]  @type int  AS BEGIN  declare @booked\_by uniqueidentifier  set @booked\_by = NEWID()    declare @booking\_date smalldatetime  set @booking\_date = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(5) t\_import\_data  SET booked\_by = @booked\_by,  booking\_date = @booking\_date,  state\_machine\_status = 2 /\*Working\*/  WHERE [type] = @type  AND (state\_machine\_status = 1 /\*Ready\*/)  IF @@ROWCOUNT > 0  BEGIN  SELECT [id],  [type],  [payload],  [creationdate],  [lastupdate],  [booked\_by],  [booking\_date],  [retry],  [state\_machine\_status]  FROM t\_import\_data with(nolock)  WHERE [type] = @type  AND booked\_by = @booked\_by  END  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_import\_data]](#yOjz4MRQxXPPTp9qgVyGmTRcv5k=)

[licensing]

|  |
| --- |
| [licensing].[p\_import\_data\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_import\_data\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_import\_data]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_import\_data]](#yOjz4MRQxXPPTp9qgVyGmTRcv5k=)

[licensing]

|  |
| --- |
| [licensing].[p\_import\_data\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @type | int | 4 |  |
| @payload | nvarchar(max) | max |  |
| @state\_machine\_status | int | 4 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_import\_data\_insert]  @type int,  @payload nvarchar(MAX),  @state\_machine\_status int,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_import\_data] (  [type],  [payload],  [lastupdate],  [booked\_by],  [booking\_date],  [state\_machine\_status]  )  VALUES (  @type,  @payload,  getutcdate(),  null,  null,  @state\_machine\_status  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_import\_data]](#yOjz4MRQxXPPTp9qgVyGmTRcv5k=)

[licensing]

|  |
| --- |
| [licensing].[p\_import\_data\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @number\_of\_retries | int | 4 |  |
| @state\_machine\_status | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_import\_data\_unbook]  @id int,  @number\_of\_retries int,  @state\_machine\_status int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE t\_import\_data  SET booked\_by = null,  booking\_date = null,  lastupdate = GETUTCDATE(),  retry = @number\_of\_retries,  state\_machine\_status = @state\_machine\_status  WHERE id = @id  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_import\_data]](#yOjz4MRQxXPPTp9qgVyGmTRcv5k=)

[licensing]

|  |
| --- |
| [licensing].[p\_license\_file\_history\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_license\_file\_history\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer],  [license\_data],  [created\_by],  [date\_created]  FROM [licensing].[t\_license\_file\_history] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_license\_file\_history]](#SIlmTMR49Mk3ZHwNuyVJLwgmdAI=)

[licensing]

|  |
| --- |
| [licensing].[p\_license\_file\_history\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer | nvarchar(500) | 1000 |  |
| @license\_data | varbinary(8000) | 8000 |  |
| @created\_by | nvarchar(250) | 500 |  |
| @date\_created | datetime | 8 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_license\_file\_history\_insert]  @customer nvarchar(500),  @license\_data varbinary(8000),  @created\_by nvarchar(250),  @date\_created datetime,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_license\_file\_history] (  [customer],  [license\_data],  [created\_by],  [date\_created]  )  VALUES (  @customer,  @license\_data,  @created\_by,  @date\_created  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_license\_file\_history]](#SIlmTMR49Mk3ZHwNuyVJLwgmdAI=)

[licensing]

|  |
| --- |
| [licensing].[p\_license\_file\_history\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer­Filter | nvarchar(500) | 1000 |  |
| @date\_created\_from | datetime | 8 |  |
| @date\_created\_to | datetime | 8 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_license\_file\_history\_search]  @customer­Filter nvarchar(500),  @date\_created\_from datetime,  @date\_created\_to datetime,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer],  [license\_data],  [created\_by],  [date\_created]  FROM [licensing].[t\_license\_file\_history] WITH(NOLOCK)  WHERE ((@customer­Filter is null) or ([customer] like @customer­Filter))  AND ((@date\_created\_from is null) or ([date\_created] >= @date\_created\_from))  AND ((@date\_created\_to is null) or ([date\_created] <= @date\_created\_to))  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_license\_file\_history] WITH(NOLOCK)  WHERE ((@customer­Filter is null) or ([customer] like @customer­Filter))  AND ((@date\_created\_from is null) or ([date\_created] >= @date\_created\_from))  AND ((@date\_created\_to is null) or ([date\_created] <= @date\_created\_to))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_license\_file\_history]](#SIlmTMR49Mk3ZHwNuyVJLwgmdAI=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_order]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [currency],  [promo\_code\_id],  [payment\_method\_id],  [created\_by\_username],  [date\_created],  [status],  [external\_reference]  FROM [licensing].[t\_order] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_order] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [currency],  [promo\_code\_id],  [payment\_method\_id],  [created\_by\_username],  [date\_created],  [status],  [external\_reference]  FROM [licensing].[t\_order] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @currency | nvarchar(5) | 10 |  |
| @promo\_code\_id | int | 4 |  |
| @payment\_method\_id | int | 4 |  |
| @created\_by\_username | nvarchar(250) | 500 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_insert]  @subscription\_id int,  @currency nvarchar(5),  @promo\_code\_id int,  @payment\_method\_id int,  @created\_by\_username nvarchar(250),  @date\_created datetime,  @status tinyint,  @external\_reference nvarchar(500),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_order] (  [subscription\_id],  [currency],  [promo\_code\_id],  [payment\_method\_id],  [created\_by\_username],  [date\_created],  [status],  [external\_reference]  )  VALUES (  @subscription\_id,  @currency,  @promo\_code\_id,  @payment\_method\_id,  @created\_by\_username,  @date\_created,  @status,  @external\_reference  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @order\_status | int | 4 |  |
| @subscription\_activeto\_start | datetime | 8 |  |
| @subscription\_activeto\_end | datetime | 8 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_search]  @subscription\_id int,  @product\_id int,  @order\_status int,  @subscription\_activeto\_start datetime,  @subscription\_activeto\_end datetime,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [currency],  [promo\_code\_id],  [payment\_method\_id],  [created\_by\_username],  [date\_created],  [status],  [external\_reference]  FROM [licensing].[t\_order] WITH(NOLOCK)  WHERE ((@subscription\_id is null) or ([subscription\_id] = @subscription\_id))  AND ((@order\_status is null) or ([status] = @order\_status))  AND ((@product\_id is null) or (EXISTS (  SELECT NULL  FROM [licensing].[t\_orderline]  WHERE [t\_orderline].[order\_id] = [t\_order].[id]  AND [t\_orderline].[product\_id] = @product\_id  )))  AND (((@subscription\_activeto\_start is null) and (@subscription\_activeto\_end is null)) or (EXISTS (  SELECT NULL  FROM [licensing].[t\_subscription]  WHERE [t\_subscription].[id] = [t\_order].[subscription\_id]  AND ((@subscription\_activeto\_start is null) or ([t\_subscription].[active\_to] >= @subscription\_activeto\_start))  AND ((@subscription\_activeto\_end is null) or ([t\_subscription].[active\_to] <= @subscription\_activeto\_end))  )))  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_order] WITH(NOLOCK)  WHERE ((@subscription\_id is null) or ([subscription\_id] = @subscription\_id))  AND ((@order\_status is null) or ([status] = @order\_status))  AND ((@product\_id is null) or (EXISTS (  SELECT NULL  FROM [licensing].[t\_orderline]  WHERE [t\_orderline].[order\_id] = [t\_order].[id]  AND [t\_orderline].[product\_id] = @product\_id  )))  AND (((@subscription\_activeto\_start is null) and (@subscription\_activeto\_end is null)) or (EXISTS (  SELECT NULL  FROM [licensing].[t\_subscription]  WHERE [t\_subscription].[id] = [t\_order].[subscription\_id]  AND ((@subscription\_activeto\_start is null) or ([t\_subscription].[active\_to] >= @subscription\_activeto\_start))  AND ((@subscription\_activeto\_end is null) or ([t\_subscription].[active\_to] <= @subscription\_activeto\_end))  )))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_order\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @subscription\_id | int | 4 |  |
| @currency | nvarchar(5) | 10 |  |
| @promo\_code\_id | int | 4 |  |
| @payment\_method\_id | int | 4 |  |
| @created\_by\_username | nvarchar(250) | 500 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_order\_update]  @id int,  @subscription\_id int,  @currency nvarchar(5),  @promo\_code\_id int,  @payment\_method\_id int,  @created\_by\_username nvarchar(250),  @date\_created datetime,  @status tinyint,  @external\_reference nvarchar(500),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_order] SET  [subscription\_id] = @subscription\_id,  [currency] = @currency,  [promo\_code\_id] = @promo\_code\_id,  [payment\_method\_id] = @payment\_method\_id,  [created\_by\_username] = @created\_by\_username,  [date\_created] = @date\_created,  [status] = @status,  [external\_reference] = @external\_reference  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_order]](#Abm+/exewd8b+kfSgBGP3mREH2k=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_orderline]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [product\_id],  [number\_of\_users],  [number\_of\_devices],  [billing\_type\_id],  [amount],  [status]  FROM [licensing].[t\_orderline] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_orderline] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [product\_id],  [number\_of\_users],  [number\_of\_devices],  [billing\_type\_id],  [amount],  [status]  FROM [licensing].[t\_orderline] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_get\_by\_orderid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @order\_id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_get\_by\_orderid]  @order\_id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [product\_id],  [number\_of\_users],  [number\_of\_devices],  [billing\_type\_id],  [amount],  [status]  FROM [licensing].[t\_orderline] WITH(NOLOCK)  WHERE [order\_id] = @order\_id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @order\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @amount | money | 8 |  |
| @status | tinyint | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_insert]  @order\_id int,  @product\_id int,  @number\_of\_users int,  @number\_of\_devices int,  @billing\_type\_id int,  @amount money,  @status tinyint,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_orderline] (  [order\_id],  [product\_id],  [number\_of\_users],  [number\_of\_devices],  [billing\_type\_id],  [amount],  [status]  )  VALUES (  @order\_id,  @product\_id,  @number\_of\_users,  @number\_of\_devices,  @billing\_type\_id,  @amount,  @status  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_orderline\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @order\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @amount | money | 8 |  |
| @status | tinyint | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_orderline\_update]  @id int,  @order\_id int,  @product\_id int,  @number\_of\_users int,  @number\_of\_devices int,  @billing\_type\_id int,  @amount money,  @status tinyint,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_orderline] SET  [order\_id] = @order\_id,  [product\_id] = @product\_id,  [number\_of\_users] = @number\_of\_users,  [number\_of\_devices] = @number\_of\_devices,  [billing\_type\_id] = @billing\_type\_id,  [amount] = @amount,  [status] = @status  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_orderline]](#gW/8v+TiudxKbw4411ArMKtXy2Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_payment]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [amount],  [external\_reference],  [date\_created],  [status]  FROM [licensing].[t\_payment] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_payment] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [amount],  [external\_reference],  [date\_created],  [status]  FROM [licensing].[t\_payment] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @order\_id | int | 4 |  |
| @amount | money | 8 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_insert]  @order\_id int,  @amount money,  @external\_reference nvarchar(500),  @date\_created datetime,  @status tinyint,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_payment] (  [order\_id],  [amount],  [external\_reference],  [date\_created],  [status]  )  VALUES (  @order\_id,  @amount,  @external\_reference,  @date\_created,  @status  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_method\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_method\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_payment\_method]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_method\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_method\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_payment\_method] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_payment\_method] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_method\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_method\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_payment\_method] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_method\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(250) | 500 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_method\_insert]  @name nvarchar(250),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_payment\_method] (  [name]  )  VALUES (  @name  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_method\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_method\_update]  @id int,  @name nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_payment\_method] SET  [name] = @name  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment\_method]](#bq8EjbGuD1o8wQd83KAB4kZD7RE=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @order\_id | int | 4 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @payment\_status | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_search]  @order\_id int,  @external\_reference nvarchar(500),  @payment\_status int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [order\_id],  [amount],  [external\_reference],  [date\_created],  [status]  FROM [licensing].[t\_payment] WITH(NOLOCK)  WHERE ((@order\_id is null) or ([order\_id] = @order\_id))  AND ((@external\_reference is null) or ([external\_reference] = @external\_reference))  AND ((@payment\_status is null) or ([status] = @payment\_status))  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_payment] WITH(NOLOCK)  WHERE ((@order\_id is null) or ([order\_id] = @order\_id))  AND ((@external\_reference is null) or ([external\_reference] = @external\_reference))  AND ((@payment\_status is null) or ([status] = @payment\_status))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_payment\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @order\_id | int | 4 |  |
| @amount | money | 8 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_payment\_update]  @id int,  @order\_id int,  @amount money,  @external\_reference nvarchar(500),  @date\_created datetime,  @status tinyint,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_payment] SET  [order\_id] = @order\_id,  [amount] = @amount,  [external\_reference] = @external\_reference,  [date\_created] = @date\_created,  [status] = @status  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_payment]](#hYxAzoyBQRnrW5Nz12ue3oquKm4=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_price]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [pricelist\_id],  [billing\_type\_id],  [product\_id],  [amount\_initial],  [amount\_per\_user],  [amount\_per\_device]  FROM [licensing].[t\_price] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_price] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [pricelist\_id],  [billing\_type\_id],  [product\_id],  [amount\_initial],  [amount\_per\_user],  [amount\_per\_device]  FROM [licensing].[t\_price] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_get\_by\_pricelistid] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @pricelist\_id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_get\_by\_pricelistid]  @pricelist\_id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [pricelist\_id],  [billing\_type\_id],  [product\_id],  [amount\_initial],  [amount\_per\_user],  [amount\_per\_device]  FROM [licensing].[t\_price] WITH(NOLOCK)  WHERE [pricelist\_id] = @pricelist\_id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pricelist\_id | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @amount\_initial | money | 8 |  |
| @amount\_per\_user | money | 8 |  |
| @amount\_per\_device | money | 8 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_insert]  @pricelist\_id int,  @billing\_type\_id int,  @product\_id int,  @amount\_initial money,  @amount\_per\_user money,  @amount\_per\_device money,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_price] (  [pricelist\_id],  [billing\_type\_id],  [product\_id],  [amount\_initial],  [amount\_per\_user],  [amount\_per\_device]  )  VALUES (  @pricelist\_id,  @billing\_type\_id,  @product\_id,  @amount\_initial,  @amount\_per\_user,  @amount\_per\_device  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_price\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @pricelist\_id | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @amount\_initial | money | 8 |  |
| @amount\_per\_user | money | 8 |  |
| @amount\_per\_device | money | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_price\_update]  @id int,  @pricelist\_id int,  @billing\_type\_id int,  @product\_id int,  @amount\_initial money,  @amount\_per\_user money,  @amount\_per\_device money,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_price] SET  [pricelist\_id] = @pricelist\_id,  [billing\_type\_id] = @billing\_type\_id,  [product\_id] = @product\_id,  [amount\_initial] = @amount\_initial,  [amount\_per\_user] = @amount\_per\_user,  [amount\_per\_device] = @amount\_per\_device  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_price]](#3xfuQEL8DMsOwiVfhBOSiA8gHPA=)

[licensing]

|  |
| --- |
| [licensing].[p\_pricelist\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_pricelist\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_pricelist]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[licensing]

|  |
| --- |
| [licensing].[p\_pricelist\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_pricelist\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [currency],  [active\_from],  [active\_to],  [is\_active]  FROM [licensing].[t\_pricelist] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_pricelist] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[licensing]

|  |
| --- |
| [licensing].[p\_pricelist\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_pricelist\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [currency],  [active\_from],  [active\_to],  [is\_active]  FROM [licensing].[t\_pricelist] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[licensing]

|  |
| --- |
| [licensing].[p\_pricelist\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(500) | 1000 |  |
| @currency | nvarchar(5) | 10 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @is\_active | bit | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_pricelist\_insert]  @name nvarchar(500),  @currency nvarchar(5),  @active\_from datetime,  @active\_to datetime,  @is\_active bit,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_pricelist] (  [name],  [currency],  [active\_from],  [active\_to],  [is\_active]  )  VALUES (  @name,  @currency,  @active\_from,  @active\_to,  @is\_active  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[licensing]

|  |
| --- |
| [licensing].[p\_pricelist\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(500) | 1000 |  |
| @currency | nvarchar(5) | 10 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @is\_active | bit | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_pricelist\_update]  @id int,  @name nvarchar(500),  @currency nvarchar(5),  @active\_from datetime,  @active\_to datetime,  @is\_active bit,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_pricelist] SET  [name] = @name,  [currency] = @currency,  [active\_from] = @active\_from,  [active\_to] = @active\_to,  [is\_active] = @is\_active  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_pricelist]](#lfqnc4nKM7PvrMGyuhtCjbwwf44=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_product]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [description],  [product\_family\_code],  [min\_number\_of\_users],  [max\_number\_of\_users],  [min\_number\_of\_devices],  [max\_number\_of\_devices],  [external\_reference]  FROM [licensing].[t\_product] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_product] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [description],  [product\_family\_code],  [min\_number\_of\_users],  [max\_number\_of\_users],  [min\_number\_of\_devices],  [max\_number\_of\_devices],  [external\_reference]  FROM [licensing].[t\_product] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_get\_by\_productfamily] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @product\_family\_code | nvarchar(100) | 200 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_get\_by\_productfamily]  @product\_family\_code nvarchar(100)  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name],  [description],  [product\_family\_code],  [min\_number\_of\_users],  [max\_number\_of\_users],  [min\_number\_of\_devices],  [max\_number\_of\_devices],  [external\_reference]  FROM [licensing].[t\_product] WITH(NOLOCK)  WHERE [product\_family\_code] = @product\_family\_code  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(500) | 1000 |  |
| @description | nvarchar(max) | max |  |
| @product\_family\_code | nvarchar(100) | 200 |  |
| @min\_number\_of\_users | int | 4 |  |
| @max\_number\_of\_users | int | 4 |  |
| @min\_number\_of\_devices | int | 4 |  |
| @max\_number\_of\_devices | int | 4 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_insert]  @name nvarchar(500),  @description nvarchar(max),  @product\_family\_code nvarchar(100),  @min\_number\_of\_users int,  @max\_number\_of\_users int,  @min\_number\_of\_devices int,  @max\_number\_of\_devices int,  @external\_reference nvarchar(500),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_product] (  [name],  [description],  [product\_family\_code],  [min\_number\_of\_users],  [max\_number\_of\_users],  [min\_number\_of\_devices],  [max\_number\_of\_devices],  [external\_reference]  )  VALUES (  @name,  @description,  @product\_family\_code,  @min\_number\_of\_users,  @max\_number\_of\_users,  @min\_number\_of\_devices,  @max\_number\_of\_devices,  @external\_reference  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_product\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(500) | 1000 |  |
| @description | nvarchar(max) | max |  |
| @product\_family\_code | nvarchar(100) | 200 |  |
| @min\_number\_of\_users | int | 4 |  |
| @max\_number\_of\_users | int | 4 |  |
| @min\_number\_of\_devices | int | 4 |  |
| @max\_number\_of\_devices | int | 4 |  |
| @external\_reference | nvarchar(500) | 1000 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_product\_update]  @id int,  @name nvarchar(500),  @description nvarchar(max),  @product\_family\_code nvarchar(100),  @min\_number\_of\_users int,  @max\_number\_of\_users int,  @min\_number\_of\_devices int,  @max\_number\_of\_devices int,  @external\_reference nvarchar(500),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_product] SET  [name] = @name,  [description] = @description,  [product\_family\_code] = @product\_family\_code,  [min\_number\_of\_users] = @min\_number\_of\_users,  [max\_number\_of\_users] = @max\_number\_of\_users,  [min\_number\_of\_devices] = @min\_number\_of\_devices,  [max\_number\_of\_devices] = @max\_number\_of\_devices,  [external\_reference] = external\_reference  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_promo\_code]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [code],  [promo\_type\_id],  [description],  [product\_id],  [max\_number\_of\_uses],  [number\_of\_users],  [number\_of\_devices],  [number\_of\_free\_months],  [active\_from],  [active\_to],  [discount\_amount],  [discount\_percent],  [usage\_count]  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_get\_by\_code] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @code | nvarchar(100) | 200 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_get\_by\_code]  @code nvarchar(100)  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [code],  [promo\_type\_id],  [description],  [product\_id],  [max\_number\_of\_uses],  [number\_of\_users],  [number\_of\_devices],  [number\_of\_free\_months],  [active\_from],  [active\_to],  [discount\_amount],  [discount\_percent],  [usage\_count]  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  WHERE [code] = @code  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [code],  [promo\_type\_id],  [description],  [product\_id],  [max\_number\_of\_uses],  [number\_of\_users],  [number\_of\_devices],  [number\_of\_free\_months],  [active\_from],  [active\_to],  [discount\_amount],  [discount\_percent],  [usage\_count]  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_get\_by\_promotypeid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @promo\_type\_id | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_get\_by\_promotypeid]  @promo\_type\_id int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [code],  [promo\_type\_id],  [description],  [product\_id],  [max\_number\_of\_uses],  [number\_of\_users],  [number\_of\_devices],  [number\_of\_free\_months],  [active\_from],  [active\_to],  [discount\_amount],  [discount\_percent],  [usage\_count]  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  WHERE [promo\_type\_id] = @promo\_type\_id  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_promo\_code] WITH(NOLOCK)  WHERE [promo\_type\_id] = @promo\_type\_id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @code | nvarchar(100) | 200 |  |
| @promo\_type\_id | int | 4 |  |
| @description | nvarchar(1000) | 2000 |  |
| @product\_id | int | 4 |  |
| @max\_number\_of\_uses | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @number\_of\_free\_months | int | 4 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @discount\_amount | money | 8 |  |
| @discount\_percent | decimal(18,0) | 9 |  |
| @usage\_count | int | 4 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_insert]  @code nvarchar(100),  @promo\_type\_id int,  @description nvarchar(1000),  @product\_id int,  @max\_number\_of\_uses int,  @number\_of\_users int,  @number\_of\_devices int,  @number\_of\_free\_months int,  @active\_from datetime,  @active\_to datetime,  @discount\_amount money,  @discount\_percent decimal,  @usage\_count int,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_promo\_code] (  [code],  [promo\_type\_id],  [description],  [product\_id],  [max\_number\_of\_uses],  [number\_of\_users],  [number\_of\_devices],  [number\_of\_free\_months],  [active\_from],  [active\_to],  [discount\_amount],  [discount\_percent],  [usage\_count]  )  VALUES (  @code,  @promo\_type\_id,  @description,  @product\_id,  @max\_number\_of\_uses,  @number\_of\_users,  @number\_of\_devices,  @number\_of\_free\_months,  @active\_from,  @active\_to,  @discount\_amount,  @discount\_percent,  @usage\_count  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_code\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @promo\_type\_id | int | 4 |  |
| @code | nvarchar(100) | 200 |  |
| @description | nvarchar(1000) | 2000 |  |
| @product\_id | int | 4 |  |
| @max\_number\_of\_uses | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @number\_of\_free\_months | int | 4 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @discount\_amount | money | 8 |  |
| @discount\_percent | decimal(18,0) | 9 |  |
| @usage\_count | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_code\_update]  @id int,  @promo\_type\_id int,  @code nvarchar(100),  @description nvarchar(1000),  @product\_id int,  @max\_number\_of\_uses int,  @number\_of\_users int,  @number\_of\_devices int,  @number\_of\_free\_months int,  @active\_from datetime,  @active\_to datetime,  @discount\_amount money,  @discount\_percent decimal,  @usage\_count int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_promo\_code] SET  [code] = @code,  [promo\_type\_id] = @promo\_type\_id,  [description] = @description,  [product\_id] = @product\_id,  [max\_number\_of\_uses] = @max\_number\_of\_uses,  [number\_of\_users] = @number\_of\_users,  [number\_of\_devices] = @number\_of\_devices,  [number\_of\_free\_months] = @number\_of\_free\_months,  [active\_from] = @active\_from,  [active\_to] = @active\_to,  [discount\_amount] = @discount\_amount,  [discount\_percent] = @discount\_percent,  [usage\_count] = @usage\_count  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_code]](#6oqplzrZY3UacWTAEhXbFt60GSQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_type\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_type\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_promo\_type]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_type\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_type\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_promo\_type] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_promo\_type] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_type\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_type\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [name]  FROM [licensing].[t\_promo\_type] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_type\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @name | nvarchar(250) | 500 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_type\_insert]  @name nvarchar(250),  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_promo\_type] (  [name]  )  VALUES (  @name  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

|  |
| --- |
| [licensing].[p\_promo\_type\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @name | nvarchar(250) | 500 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_promo\_type\_update]  @id int,  @name nvarchar(250),  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_promo\_type] SET  [name] = @name  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_promo\_type]](#XWhkhxJf9mvneWNL4gIOGe4ENts=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_subscription]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_subscription\_device]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [device\_code],  [device\_name],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [device\_code],  [device\_name],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_get\_by\_subscriptionid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @status | tinyint | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_get\_by\_subscriptionid]  @subscription\_id int,  @status tinyint,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [device\_code],  [device\_name],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [subscription\_id] = @subscription\_id  AND ((@status is null) or ([status] = @status))  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [subscription\_id] = @subscription\_id  AND ((@status is null) or ([status] = @status))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_get\_by\_subscriptionid\_devicecode] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @subscription\_id | int | 4 |
| @device\_code | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_get\_by\_subscriptionid\_devicecode]  @subscription\_id int,  @device\_code nvarchar(250)  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [device\_code],  [device\_name],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [subscription\_id] = @subscription\_id  AND [device\_code] = @device\_code  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @device\_code | nvarchar(250) | 500 |  |
| @device\_name | nvarchar(500) | 1000 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_insert]  @subscription\_id int,  @device\_code nvarchar(250),  @device\_name nvarchar(500),  @date\_created datetime,  @status tinyint,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_subscription\_device] (  [subscription\_id],  [device\_code],  [device\_name],  [date\_created],  [status]  )  VALUES (  @subscription\_id,  @device\_code,  @device\_name,  @date\_created,  @status  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_device\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @subscription\_id | int | 4 |  |
| @device\_code | nvarchar(250) | 500 |  |
| @device\_name | nvarchar(500) | 1000 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_device\_update]  @id int,  @subscription\_id int,  @device\_code nvarchar(250),  @device\_name nvarchar(500),  @date\_created datetime,  @status tinyint,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_subscription\_device] SET  [subscription\_id] = @subscription\_id,  [device\_code] = @device\_code,  [device\_name] = @device\_name,  [date\_created] = @date\_created,  [status] = @status  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_by\_customerid] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer\_id | int | 4 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_by\_customerid]  @customer\_id int,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE [customer\_id] = @customer\_id  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE [customer\_id] = @customer\_id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_by\_customerid\_productfamilycode] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer\_id | int | 4 |  |
| @product\_family\_code | nvarchar(100) | 200 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_by\_customerid\_productfamilycode]  @customer\_id int,  @product\_family\_code nvarchar(100),  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE ((@customer\_id is null) or ([customer\_id] = @customer\_id))  AND ((@product\_family\_code is null) OR EXISTS (  SELECT NULL FROM [licensing].[t\_product]  WHERE [product\_family\_code] = @product\_family\_code  AND [product\_id] = [t\_subscription].product\_id  ))  ORDER BY [date\_created] DESC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE ((@customer\_id is null) or ([customer\_id] = @customer\_id))  AND ((@product\_family\_code is null) OR EXISTS (  SELECT NULL FROM [licensing].[t\_product]  WHERE [product\_family\_code] = @product\_family\_code  AND [product\_id] = [t\_subscription].product\_id  ))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_by\_subscriptiondevice] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @device\_code | nvarchar(250) | 500 |  |
| @device\_status | tinyint | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_by\_subscriptiondevice]  @device\_code nvarchar(250),  @device\_status tinyint,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE EXISTS (  SELECT null  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [subscription\_id] = [t\_subscription].id  AND [device\_code] = @device\_code  AND ((@device\_status is null) or ([status] = @device\_status))  )  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE EXISTS (  SELECT null  FROM [licensing].[t\_subscription\_device] WITH(NOLOCK)  WHERE [subscription\_id] = [t\_subscription].id  AND [device\_code] = @device\_code  AND ((@device\_status is null) or ([status] = @device\_status))  )  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[t\_subscription\_device]](#lJFbLWHid2xndcdtbsATRo54g/w=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_get\_by\_subscriptionuser] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @username | nvarchar(250) | 500 |  |
| @user\_status | tinyint | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_get\_by\_subscriptionuser]  @username nvarchar(250),  @user\_status tinyint,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE EXISTS (  SELECT null  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE [subscription\_id] = [t\_subscription].id  AND [username] = @username  AND ((@user\_status is null) or ([status] = @user\_status))  )  ORDER BY [id] ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  WHERE EXISTS (  SELECT null  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE [subscription\_id] = [t\_subscription].id  AND [username] = @username  AND ((@user\_status is null) or ([status] = @user\_status))  )  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @date\_created | datetime | 8 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_insert]  @customer\_id int,  @product\_id int,  @billing\_type\_id int,  @number\_of\_users int,  @number\_of\_devices int,  @active\_from datetime,  @active\_to datetime,  @status tinyint,  @date\_created datetime,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_subscription] (  [customer\_id],  [product\_id],  [billing\_type\_id],  [number\_of\_users],  [number\_of\_devices],  [active\_from],  [active\_to],  [status],  [date\_created]  )  VALUES (  @customer\_id,  @product\_id,  @billing\_type\_id,  @number\_of\_users,  @number\_of\_devices,  @active\_from,  @active\_to,  @status,  @date\_created  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @customer­Username­Filter | nvarchar(500) | 1000 |  |
| @active­From­Date­Filter | nvarchar(500) | 1000 |  |
| @active­To­Date­Filter | nvarchar(500) | 1000 |  |
| @status­Filter | nvarchar(500) | 1000 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @wants­Count | bit | 1 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_search]  @customer­Username­Filter nvarchar(500),  @active­From­Date­Filter nvarchar(500),  @active­To­Date­Filter nvarchar(500),  @status­Filter nvarchar(500),  @pagesize int,  @pagenumber int,  @wants­Count bit,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [t\_subscription].[id],  [t\_subscription].[customer\_id],  [t\_subscription].[product\_id],  [t\_subscription].[billing\_type\_id],  [t\_subscription].[number\_of\_users],  [t\_subscription].[number\_of\_devices],  [t\_subscription].[active\_from],  [t\_subscription].[active\_to],  [t\_subscription].[status],  [t\_subscription].[date\_created],  [t\_customer].username as customer\_username,  [t\_customer].[name] as customer\_name  FROM [licensing].[t\_subscription] WITH(NOLOCK)  INNER JOIN [licensing].[t\_customer] WITH(NOLOCK) ON [t\_subscription].customer\_id = [t\_customer].id  WHERE ((@customer­Username­Filter is null) or ([t\_customer].username like @customer­Username­Filter))  AND ((@active­From­Date­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[active\_from], SUBSTRING(@active­From­Date­Filter, 1, 2), CONVERT(DATETIME, SUBSTRING(@active­From­Date­Filter, 3, 1000), 120)) = 1))  AND ((@active­To­Date­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[active\_to], SUBSTRING(@active­To­Date­Filter, 1, 2), CONVERT(DATETIME, SUBSTRING(@active­To­Date­Filter, 3, 1000), 120)) = 1))  AND ((@status­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[status], SUBSTRING(@status­Filter, 1, 2), CONVERT(TINYINT, SUBSTRING(@status­Filter, 3, 1000))) = 1))  ORDER BY [date\_created] DESC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  IF (@wants­Count = 1)  BEGIN  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription] WITH(NOLOCK)  INNER JOIN [licensing].[t\_customer] WITH(NOLOCK) ON [t\_subscription].customer\_id = [t\_customer].id  WHERE ((@customer­Username­Filter is null) or ([t\_customer].username like @customer­Username­Filter))  AND ((@active­From­Date­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[active\_from], SUBSTRING(@active­From­Date­Filter, 1, 2), CONVERT(DATETIME, SUBSTRING(@active­From­Date­Filter, 3, 1000), 120)) = 1))  AND ((@active­To­Date­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[active\_to], SUBSTRING(@active­To­Date­Filter, 1, 2), CONVERT(DATETIME, SUBSTRING(@active­To­Date­Filter, 3, 1000), 120)) = 1))  AND ((@status­Filter is null) or ([licensing].[fn\_­Apply­Filter]([t\_subscription].[status], SUBSTRING(@status­Filter, 1, 2), CONVERT(TINYINT, SUBSTRING(@status­Filter, 3, 1000))) = 1))  END ELSE BEGIN  SET @totalrows = 0  END  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_customer]](#+gwO/kV6vlFfFAxpzjU3+jxS/Pc=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[fn\_­Apply­Filter]](#++oVkkv6GS0hNkB/MhyJdusLT+Y=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @customer\_id | int | 4 |  |
| @product\_id | int | 4 |  |
| @billing\_type\_id | int | 4 |  |
| @number\_of\_users | int | 4 |  |
| @number\_of\_devices | int | 4 |  |
| @active\_from | datetime | 8 |  |
| @active\_to | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @date\_created | datetime | 8 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_update]  @id int,  @customer\_id int,  @product\_id int,  @billing\_type\_id int,  @number\_of\_users int,  @number\_of\_devices int,  @active\_from datetime,  @active\_to datetime,  @status tinyint,  @date\_created datetime,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_subscription] SET  [customer\_id] = @customer\_id,  [product\_id] = @product\_id,  [billing\_type\_id] = @billing\_type\_id,  [number\_of\_users] = @number\_of\_users,  [number\_of\_devices] = @number\_of\_devices,  [active\_from] = @active\_from,  [active\_to] = @active\_to,  [status] = @status,  [date\_created] = @date\_created  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_delete] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_delete]  @id int,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  DELETE FROM [licensing].[t\_subscription\_user]  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_get\_all] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_get\_all]  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [username],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  ORDER BY id ASC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_get\_by\_id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @id | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_get\_by\_id]  @id int  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [username],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE [id] = @id  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_get\_by\_subscriptionid\_username] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @subscription\_id | int | 4 |
| @username | nvarchar(250) | 500 |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_get\_by\_subscriptionid\_username]  @subscription\_id int,  @username nvarchar(250)  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [username],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE [subscription\_id] = @subscription\_id  AND [username] = @username  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @username | nvarchar(250) | 500 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @id | int | 4 | Out |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_insert]  @subscription\_id int,  @username nvarchar(250),  @date\_created datetime,  @status tinyint,  @id int output,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  INSERT INTO [licensing].[t\_subscription\_user] (  [subscription\_id],  [username],  [date\_created],  [status]  )  VALUES (  @subscription\_id,  @username,  @date\_created,  @status  )  SET @id = SCOPE\_­IDENTITY()  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_search] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @subscription\_id | int | 4 |  |
| @username | nvarchar(250) | 500 |  |
| @product\_family\_code | nvarchar(100) | 200 |  |
| @status | tinyint | 1 |  |
| @pagesize | int | 4 |  |
| @pagenumber | int | 4 |  |
| @totalrows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_search]  @subscription\_id int,  @username nvarchar(250),  @product\_family\_code nvarchar(100),  @status tinyint,  @pagesize int,  @pagenumber int,  @totalrows int = 0 output  AS BEGIN  SET NOCOUNT ON  SELECT [id],  [subscription\_id],  [username],  [date\_created],  [status]  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE ((@subscription\_id is null) OR ([subscription\_id] = @subscription\_id))  AND ((@username is null) OR ([username] = @username))  AND ((@status is null) OR ([status] = @status))  AND ((@product\_family\_code is null) OR EXISTS (  SELECT NULL  FROM [licensing].[t\_subscription]  INNER JOIN [licensing].[t\_product] ON [t\_subscription].product\_id = [t\_product].id  WHERE [product\_family\_code] = @product\_family\_code  AND [t\_subscription].id = [t\_subscription\_user].[subscription\_id]  ))  ORDER BY [id] DESC  OFFSET @pagesize \* (@pagenumber - 1) ROWS  FETCH NEXT @pagesize ROWS ONLY;  SELECT @totalrows = count(1)  FROM [licensing].[t\_subscription\_user] WITH(NOLOCK)  WHERE ((@subscription\_id is null) OR ([subscription\_id] = @subscription\_id))  AND ((@username is null) OR ([username] = @username))  AND ((@status is null) OR ([status] = @status))  AND ((@product\_family\_code is null) OR EXISTS (  SELECT NULL  FROM [licensing].[t\_subscription]  INNER JOIN [licensing].[t\_product] ON [t\_subscription].product\_id = [t\_product].id  WHERE [product\_family\_code] = @product\_family\_code  AND [t\_subscription].id = [t\_subscription\_user].[subscription\_id]  ))  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_product]](#6EBiFkvfbRTw3F555fk44DFA3ks=)

[[licensing].[t\_subscription]](#0fbEPQn2ckpupg1/9FJCsZJQ3CQ=)

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [licensing].[p\_subscription\_user\_update] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @id | int | 4 |  |
| @subscription\_id | int | 4 |  |
| @username | nvarchar(250) | 500 |  |
| @date\_created | datetime | 8 |  |
| @status | tinyint | 1 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROC [licensing].[p\_subscription\_user\_update]  @id int,  @subscription\_id int,  @username nvarchar(250),  @date\_created datetime,  @status tinyint,  @rows int = 0 output  AS BEGIN  SET NOCOUNT ON  UPDATE [licensing].[t\_subscription\_user] SET  [subscription\_id] = @subscription\_id,  [username] = @username,  [date\_created] = @date\_created,  [status] = @status  WHERE [id] = @id  SET @rows = @@ROWCOUNT  SET NOCOUNT OFF  END  GO |

Uses

[[licensing].[t\_subscription\_user]](#jrEJsKBnLMDw32yaeZrtVt+09KU=)

[licensing]

|  |
| --- |
| [mft].[p\_check\_download\_available] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @requestid | uniqueidentifier | 16 |  |
| @isavailable | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_check\_download\_available]  @requestid uniqueidentifier,  @isavailable bit OUTPUT  AS  BEGIN  DECLARE @count INT  SELECT @count = COUNT(\*) FROM mft.t\_chunkingmachine with (nolock)  WHERE chunkingmachineid = @requestid  AND chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('ready to download')  AND chunks > 0  IF @count > 0  SET @isavailable = 1  ELSE  SET @isavailable = 0  END  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_check\_storageoperation\_finished] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @requestid | uniqueidentifier | 16 |  |
| @isfinished | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_check\_storageoperation\_finished]  @requestid UNIQUEIDENTIFIER,  @isfinished BIT OUTPUT  AS  BEGIN  DECLARE @count INT  SELECT @count = COUNT(\*) FROM mft.t\_storageoperations with (nolock)  WHERE storageoperationsid = @requestid  AND storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('completed')  IF @count > 0  SET @isfinished = 1  ELSE  SET @isfinished = 0  END  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[fn\_­Get­Storage­Machine­Status­Id]](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=)

[mft]

|  |
| --- |
| [mft].[p\_check\_upload\_finished] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @requestid | uniqueidentifier | 16 |  |
| @isfinished | bit | 1 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_check\_upload\_finished]  @requestid UNIQUEIDENTIFIER,  @isfinished bit OUTPUT  AS  BEGIN  DECLARE @count INT  SELECT @count = COUNT(\*) FROM mft.t\_mergemachine with (nolock)  WHERE mergemachineid = @requestid  AND mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('merge completed')  AND chunks > 0  IF @count > 0  SET @isfinished = 1  ELSE  SET @isfinished = 0  END  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_chunkingmachine\_book] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_chunkingmachine\_book]  @updateby nvarchar(50)  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('PROCESSING')  WHERE chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('READY TO PROCESS')  AND bookedby is null  AND bookingdate is null  SELECT chunkingmachineid, chunkingmachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, buffersize, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_chunkingmachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_chunkingmachine\_completed] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @chunkingmachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_chunkingmachine\_completed]  @chunkingmachineid uniqueidentifier,  @updateby nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @lastupdate,  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('DOWNLOAD COMPLETED')  WHERE chunkingmachineid = @chunkingmachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_chunkingmachine\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @chunkingmachineid | uniqueidentifier | 16 |  |
| @domain | nvarchar(250) | 500 |  |
| @username | nvarchar(250) | 500 |  |
| @password | nvarchar(500) | 1000 |  |
| @machineid | uniqueidentifier | 16 |  |
| @filepath | nvarchar(500) | 1000 |  |
| @filename | nvarchar(500) | 1000 |  |
| @filecheksum | nvarchar(128) | 256 |  |
| @temppath | nvarchar(500) | 1000 |  |
| @buffersize | int | 4 |  |
| @updateby | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_chunkingmachine\_insert]  @chunkingmachineid uniqueidentifier,  @domain nvarchar(250),  @username nvarchar(250),  @password nvarchar(500),  @machineid uniqueidentifier,  @filepath nvarchar(500),  @filename nvarchar(500),  @filecheksum nvarchar(128),  @temppath nvarchar(500),  @buffersize int,  @updateby nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  INSERT INTO mft.t\_chunkingmachine  (  chunkingmachineid,  chunkingmachinestatusid,  domain,  username,  [password],  machineid,  filepath,  [filename],  filecheksum,  temppath,  buffersize,  updateby,  lastupdate  )  VALUES  (  @chunkingmachineid,  mft.fn\_­Get­Chunking­Machine­Status­Id('READY TO PROCESS'),  @domain,  @username,  @password,  @machineid,  @filepath,  @filename,  @filecheksum,  @temppath,  @buffersize,  @updateby,  @lastupdate  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_chunkingmachine\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @chunkingmachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @chunkingmachinestatusid | int | 4 |  |
| @chunks | int | 4 |  |
| @filecheksum | nvarchar(128) | 256 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_chunkingmachine\_unbook]  @chunkingmachineid uniqueidentifier,  @updateby nvarchar(50),  @retry int,  @chunkingmachinestatusid int,  @chunks int,  @filecheksum nvarchar(128),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  chunks = @chunks,  retry = @retry,  filecheksum = @filecheksum,  chunkingmachinestatusid = @chunkingmachinestatusid  WHERE chunkingmachineid = @chunkingmachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_bookchunking] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_bookchunking]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('CLEANING')  WHERE (chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('DOWNLOAD COMPLETED') OR chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('READY TO CLEAN'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT chunkingmachineid, chunkingmachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, buffersize, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_chunkingmachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_bookmergemoving] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_bookmergemoving]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate  WHERE (mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('CLEAN COMPLETED') OR  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('OPERATION CANCELLED') OR  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('ERROR'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT mergemachineid, mergemachinestatusid,  domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_mergemachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_bookmerging] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_bookmerging]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('CLEANING')  WHERE (mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('MERGE COMPLETED') OR mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('READY TO CLEAN'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT mergemachineid, mergemachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_mergemachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_move\_chunkinghistory] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @chunkingmachineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_move\_chunkinghistory]  @chunkingmachineid uniqueidentifier  AS  BEGIN  SET NOCOUNT ON;  INSERT INTO mft.t\_chunkingmachine\_history  SELECT \*, GETUTCDATE() as moved\_on from mft.t\_chunkingmachine  WHERE mft.t\_chunkingmachine.chunkingmachineid = @chunkingmachineid  DELETE FROM mft.t\_chunkingmachine  WHERE mft.t\_chunkingmachine.chunkingmachineid = @chunkingmachineid  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[t\_chunkingmachine\_history]](#aU/tVVaTKU0VGvpN8lOEMo7OrWQ=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_move\_mergehistory] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @mergemachineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_move\_mergehistory]  @mergemachineid uniqueidentifier  AS  BEGIN  SET NOCOUNT ON;  INSERT INTO mft.t\_mergemachine\_history  SELECT \*, GETUTCDATE() as moved\_on from mft.t\_mergemachine  WHERE mft.t\_mergemachine.mergemachineid = @mergemachineid  DELETE FROM mft.t\_mergemachine  WHERE mft.t\_mergemachine.mergemachineid = @mergemachineid  SET NOCOUNT OFF  RETURN 0  END  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[t\_mergemachine\_history]](#A4S2F2HPlFcxaD6FNedLE5yGu/A=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_unbookchunking] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @chunkingmachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @chunkingmachinestatusid | int | 4 |  |
| @chunks | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_unbookchunking]  @chunkingmachineid uniqueidentifier,  @updateby nvarchar(50),  @retry int,  @chunkingmachinestatusid int,  @chunks int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  chunks = @chunks,  retry = @retry,  chunkingmachinestatusid = @chunkingmachinestatusid  WHERE chunkingmachineid = @chunkingmachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachine\_unbookmerging] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @mergemachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @mergemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachine\_unbookmerging]  @mergemachineid uniqueidentifier,  @updateby nvarchar(50),  @retry int,  @mergemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = @retry,  mergemachinestatusid = @mergemachinestatusid  WHERE mergemachineid = @mergemachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachinenotcompleted\_bookchunking] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachinenotcompleted\_bookchunking]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('CLEANING')  WHERE (chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('READY TO DOWNLOAD'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT chunkingmachineid, chunkingmachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, buffersize, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_chunkingmachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachinenotcompleted\_bookchunkingmoving] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachinenotcompleted\_bookchunkingmoving]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_chunkingmachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate  WHERE (chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('CLEAN COMPLETED') OR  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('OPERATION CANCELLED') OR  chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('ERROR'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT chunkingmachineid, chunkingmachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, buffersize, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_chunkingmachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_cleaningmachinenotcompleted\_bookmerging] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |
| @minutesfromstart | int | 4 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_cleaningmachinenotcompleted\_bookmerging]  @updateby nvarchar(50),  @minutesfromstart int  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('CLEANING')  WHERE (mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('READY TO MERGE'))  AND bookedby is null  AND bookingdate is null  AND lastupdate < DATEADD(minute, -@minutesfromstart, GETUTCDATE())  SELECT mergemachineid, mergemachinestatusid,  domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_mergemachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_configuration\_load\_documentexplorer] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @publishid | uniqueidentifier | 16 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_configuration\_load\_documentexplorer]  @publishid uniqueidentifier output  AS  SET NOCOUNT ON;  SELECT [key], [value]  FROM mft.t\_configuration\_documentexplorer with (nolock)  SELECT @publishid = publishid  FROM mft.t\_caches with (nolock)  WHERE cacheid = mft.fn\_­Get­Cache­Id('configuration\_documentexplorer')  RETURN  GO |

Uses

[mft]

|  |
| --- |
| [mft].[p\_get\_activity\_from\_chunking] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @machineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_get\_activity\_from\_chunking]  @machineid uniqueidentifier  AS  select chunkingmachineid, chunks, filecheksum,chunkingmachinestatusid  from mft.t\_chunkingmachine with (nolock)  where machineid = @machineid  and (chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('READY TO DOWNLOAD') or chunkingmachinestatusid = mft.fn\_­Get­Chunking­Machine­Status­Id('ERROR'))  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_chunkingmachine]](#L+ecWckMLzfF0Qc6dF3FQEWqMkI=)

[[mft].[fn\_­Get­Chunking­Machine­Status­Id]](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=)

[mft]

|  |
| --- |
| [mft].[p\_get\_activity\_from\_merge] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @machineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_get\_activity\_from\_merge]  @machineid uniqueidentifier  AS  select mergemachineid, filecheksum, mergemachinestatusid  from mft.t\_mergemachine with (nolock)  where machineid = @machineid  and (mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('MERGE COMPLETED') or mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('ERROR'))  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_get\_activity\_from\_storage] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @machineid | uniqueidentifier | 16 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_get\_activity\_from\_storage]  @machineid uniqueidentifier  AS  select storageoperationsid, storageoperationsmachinestatusid  from [mft].t\_storageoperations with (nolock)  where machineid = @machineid  and (storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('COMPLETED') OR storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('ERROR'))  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[fn\_­Get­Storage­Machine­Status­Id]](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=)

[mft]

|  |
| --- |
| [mft].[p\_mergemachine\_book] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_mergemachine\_book]  @updateby nvarchar(50)  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('PROCESSING')  WHERE mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('READY TO PROCESS')  AND bookedby is null  AND bookingdate is null  SELECT mergemachineid, mergemachinestatusid, domain, username, [password], machineid,  filepath, [filename], filecheksum, temppath, chunks, updateby, lastupdate,  bookedby, bookingdate, retry  FROM mft.t\_mergemachine  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_mergemachine\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @mergemachineid | uniqueidentifier | 16 |  |
| @domain | nvarchar(250) | 500 |  |
| @username | nvarchar(250) | 500 |  |
| @password | nvarchar(500) | 1000 |  |
| @machineid | uniqueidentifier | 16 |  |
| @filepath | nvarchar(500) | 1000 |  |
| @filename | nvarchar(500) | 1000 |  |
| @filecheksum | nvarchar(128) | 256 |  |
| @temppath | nvarchar(500) | 1000 |  |
| @updateby | nvarchar(50) | 100 |  |
| @chunks | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_mergemachine\_insert]  @mergemachineid uniqueidentifier,  @domain nvarchar(250),  @username nvarchar(250),  @password nvarchar(500),  @machineid uniqueidentifier,  @filepath nvarchar(500),  @filename nvarchar(500),  @filecheksum nvarchar(128),  @temppath nvarchar(500),  @updateby nvarchar(50),  @chunks int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  INSERT INTO mft.t\_mergemachine  (  mergemachineid,  mergemachinestatusid,  domain,  username,  [password],  machineid,  filepath,  [filename],  filecheksum,  temppath,  updateby,  chunks,  lastupdate  )  VALUES  (  @mergemachineid,  mft.fn\_­Get­Merge­Machine­Status­Id('CLIENT UPLOADING'),  @domain,  @username,  @password,  @machineid,  @filepath,  @filename,  @filecheksum,  @temppath,  @updateby,  @chunks,  @lastupdate  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_mergemachine\_ready\_to\_process] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @mergemachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_mergemachine\_ready\_to\_process]  @mergemachineid uniqueidentifier,  @updateby nvarchar(50),  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @lastupdate,  mergemachinestatusid = mft.fn\_­Get­Merge­Machine­Status­Id('READY TO PROCESS')  WHERE mergemachineid = @mergemachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[[mft].[fn\_­Get­Merge­Machine­Status­Id]](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=)

[mft]

|  |
| --- |
| [mft].[p\_mergemachine\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @mergemachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @mergemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_mergemachine\_unbook]  @mergemachineid uniqueidentifier,  @updateby nvarchar(50),  @retry int,  @mergemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) mft.t\_mergemachine  SET updateby = @updateby,  lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = @retry,  mergemachinestatusid = @mergemachinestatusid  WHERE mergemachineid = @mergemachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_mergemachine]](#bu7q7+v0YyUsCxS8oi40DUowbds=)

[mft]

|  |
| --- |
| [mft].[p\_storagemachine\_book] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @updateby | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_storagemachine\_book]  @updateby nvarchar(50)  AS  declare @bookeby uniqueidentifier  set @bookeby = NEWID()    declare @bookingdate datetime  set @bookingdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) [mft].t\_storageoperations  SET updatedby = @updateby,  lastupdate = @bookingdate,  bookedby = @bookeby,  bookingdate = @bookingdate,  storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('PROCESSING')  WHERE storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('READY TO PROCESS')  AND bookedby is null  AND bookingdate is null  SELECT storageoperationsid, storageoperationsmachinestatusid,storageoperationtype, updatedby, lastupdate,  bookedby, bookingdate, retry, domain, username, password, action1, action2, machineid  FROM [mft].t\_storageoperations  WHERE bookedby = @bookeby  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[fn\_­Get­Storage­Machine­Status­Id]](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=)

[mft]

|  |
| --- |
| [mft].[p\_storagemachine\_insert] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @storageoperationsid | uniqueidentifier | 16 |  |
| @domain | nvarchar(250) | 500 |  |
| @username | nvarchar(250) | 500 |  |
| @password | nvarchar(500) | 1000 |  |
| @storageoperationtype | int | 4 |  |
| @action1 | nvarchar(500) | 1000 |  |
| @action2 | nvarchar(500) | 1000 |  |
| @updateby | nvarchar(50) | 100 |  |
| @machineid | uniqueidentifier | 16 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_storagemachine\_insert]  @storageoperationsid uniqueidentifier,  @domain nvarchar(250),  @username nvarchar(250),  @password nvarchar(500),  @storageoperationtype int,  @action1 nvarchar(500),  @action2 nvarchar(500),  @updateby nvarchar(50),  @machineid uniqueidentifier,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  INSERT INTO [mft].[t\_storageoperations]  ([storageoperationsid]  ,[storageoperationsmachinestatusid]  ,[storageoperationtype]  ,[updatedby]  ,[lastupdate]  ,[action1]  ,[action2]  ,[username]  ,[password]  ,[machineid]  ,[domain])  VALUES  (  @storageoperationsid,  mft.fn\_­Get­Storage­Machine­Status­Id('READY TO PROCESS'),  @storageoperationtype,  @updateby,  @lastupdate,  @action1,  @action2,  @username,  @password,  @machineid,  @domain  )  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[fn\_­Get­Storage­Machine­Status­Id]](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=)

[mft]

|  |
| --- |
| [mft].[p\_storagemachine\_unbook] |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Max Length (Bytes) | Direction |
| @storagemachineid | uniqueidentifier | 16 |  |
| @updateby | nvarchar(50) | 100 |  |
| @retry | int | 4 |  |
| @storagemachinestatusid | int | 4 |  |
| @rows | int | 4 | Out |

SQL Script

|  |
| --- |
| CREATE PROCEDURE [mft].[p\_storagemachine\_unbook]  @storagemachineid uniqueidentifier,  @updateby nvarchar(50),  @retry int,  @storagemachinestatusid int,  @rows int = 0 output  AS  declare @lastupdate datetime  set @lastupdate = GETUTCDATE()  SET NOCOUNT ON  UPDATE top(1) [mft].[t\_storageoperations]  SET updatedby = @updateby,  lastupdate = @lastupdate,  bookedby = null,  bookingdate = null,  retry = @retry,  storageoperationsmachinestatusid = @storagemachinestatusid  WHERE storageoperationsid = @storagemachineid  set @rows = @@ROWCOUNT  SET NOCOUNT OFF  RETURN 0  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[mft]

|  |
| --- |
| [mft].[p\_storageoperations\_move\_history] |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: <Author,,Name>  -- Create date: <Create Date,,>  -- Description: <Description,,>  -- =============================================  CREATE PROCEDURE [mft].[p\_storageoperations\_move\_history]  AS  BEGIN  SET NOCOUNT ON;  INSERT INTO [mft].[t\_storageoperations\_history]  SELECT \* from [mft].[t\_storageoperations]  WHERE [mft].[t\_storageoperations].storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('COMPLETED')  OR [mft].t\_storageoperations.storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('OPERATION CANCELLED')  OR [mft].t\_storageoperations.storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('ERROR')  DELETE FROM [mft].t\_storageoperations  WHERE [mft].t\_storageoperations.storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('COMPLETED')  OR [mft].t\_storageoperations.storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('OPERATION CANCELLED')  OR [mft].t\_storageoperations.storageoperationsmachinestatusid = mft.fn\_­Get­Storage­Machine­Status­Id('ERROR')  END  GO |

Uses

[[mft].[t\_storageoperations]](#Y7425KmgHwq7Czxr/cHhDctuQW8=)

[[mft].[t\_storageoperations\_history]](#Wikc4aMSZrWQB6JETj0eWwnYvU0=)

[[mft].[fn\_­Get­Storage­Machine­Status­Id]](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=)

[mft]

|  |
| --- |
| Scalar-valued Functions |

Objects

|  |
| --- |
| Name |
| [conversion.fn\_­Get­Convertable­File­Type­Id](#wjO5ra2h68BacsEBVBMZO9ZUOCM=) |
| [conversion.fn\_­Get­State­Machine­Status­Id](#RsSpSE3LsfeWt/t7ng/gELcm9i0=) |
| [dbo.fn\_­Get­Blob­Status](#MrHRuJLPZf16ktgVr1fQoxOAtko=) |
| [dbo.fn\_­Get­Cache­Id](#O/aCmRIgLeKi5tcqjHQHnuXfp00=) |
| [dbo.fn\_­Get­Officeconverter­Machine­Status­Id](#XvEdlll/Q2Y8fou9CM7GbGxRCQA=) |
| [dbo.fn\_­Get­State­Machine­Status­Id](#OR1i7a+8r+s6UWatraats0CjTwY=) |
| [dbo.fn\_­Get­Tile­Type](#fGE7XnuGzg29WlhvQEuR0BLPbL8=) |
| [licensing.fn\_­Apply­Filter](#++oVkkv6GS0hNkB/MhyJdusLT+Y=) |
| [mft.fn\_­Get­Chunking­Machine­Status­Id](#CMIbqxRCuJm2NtAUYHHqZZkXAR0=) |
| [mft.fn\_­Get­Merge­Machine­Status­Id](#UuZJAA6onh5lt3Wq/ZTg4wh4gSE=) |
| [mft.fn\_­Get­State­Machine­Status­Id](#jjtZXPovRmvqhClb0TsUrLbTKEs=) |
| [mft.fn\_­Get­Storage­Machine­Status­Id](#bKKDMfJ/9OWj2U4ZVZB+zaF01q4=) |

|  |
| --- |
| [conversion].[fn\_­Get­Convertable­File­Type­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @convertablefiletype | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [conversion].[fn\_­Get­Convertable­File­Type­Id]  (  @convertablefiletype NVARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @convertablefiletypeid INT  SET @convertablefiletype = UPPER(@convertablefiletype)  IF @convertablefiletype = 'WORDDOCUMENT'  SET @convertablefiletypeid = 1  ELSE IF @convertablefiletype = 'EXCELDOCUMENT'  SET @convertablefiletypeid = 2  ELSE IF @convertablefiletype = 'POWERPOINTDOCUMENT'  SET @convertablefiletypeid = 3  ELSE IF @convertablefiletype = 'PDFFILE'  SET @convertablefiletypeid = 4  ELSE IF @convertablefiletype = 'UNSUPPORTED'  SET @convertablefiletypeid = 99  ELSE  SET @convertablefiletypeid = -1  RETURN @convertablefiletypeid  END  GO |

Uses

[conversion]

|  |
| --- |
| [conversion].[fn\_­Get­State­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @statemachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [conversion].[fn\_­Get­State­Machine­Status­Id]  (  @statemachinestatusdescription NVARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @statemachinestatusid INT  SET @statemachinestatusdescription = UPPER(@statemachinestatusdescription)  IF @statemachinestatusdescription = 'NOT READY'  SET @statemachinestatusid = 0  ELSE IF @statemachinestatusdescription = 'READY'  SET @statemachinestatusid = 1  ELSE IF @statemachinestatusdescription = 'WORKING'  SET @statemachinestatusid = 2  ELSE IF @statemachinestatusdescription = 'FINISHED'  SET @statemachinestatusid = 10  ELSE IF @statemachinestatusdescription = 'ERROR'  SET @statemachinestatusid = 11  ELSE  SET @statemachinestatusid = -1  RETURN @statemachinestatusid  END  GO |

Uses

[conversion]

Used By

[[conversion].[p\_conversion\_book]](#2ky7nZ3brsF0UI6fkxfmgT8GZqI=)

[[conversion].[p\_conversion\_book\_fallback]](#N/g2rojWTtdssTZ449KkIIx/4B8=)

[[conversion].[p\_conversion\_book\_history]](#Fndejxz7Ohz6SURqC1Fm5gQZiYQ=)

[[conversion].[p\_conversion\_insert]](#ZAAI0TdWV+PMPe0e2gKthXBu6VY=)

[[conversion].[p\_image\_resizing\_book]](#8H9bGc3/Fy7+H4uFbIgZhvy1agY=)

[[conversion].[p\_image\_resizing\_book\_fallback]](#s8FUIxTi+6LBh8mYurqnSND9xLQ=)

[[conversion].[p\_image\_resizing\_book\_history]](#1xpeiE5gtXIQzKAuzLeH5MY+gHc=)

[[conversion].[p\_image\_resizing\_booking\_insert]](#lJxGVVKhAjqYDYdtS1TeWbGi0Ko=)

|  |
| --- |
| [dbo].[fn\_­Get­Blob­Status] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @blobldescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[fn\_­Get­Blob­Status]  (  @blobldescription NVARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @blobstatusid INT  SET @blobldescription = UPPER(@blobldescription)  IF @blobldescription = 'NOTONAZURE'  SET @blobstatusid = 1  ELSE IF @blobldescription = 'ONAZURE'  SET @blobstatusid = 2  ELSE IF @blobldescription='CREATINGTHUMBNAILS'  SET @blobstatusid = 5  ELSE IF @blobldescription = 'THUMBNAILSAVAILABLE'  SET @blobstatusid = 6  ELSE IF @blobldescription = 'THUMBNAILERROR'  SET @blobstatusid = 7  ELSE IF @blobldescription = 'UPLOADERROR'  SET @blobstatusid = 10  ELSE IF @blobldescription = 'DELETED'  SET @blobstatusid = 11  ELSE  SET @blobstatusid = -1  RETURN @blobstatusid  END  GO |

Used By

[[dbo].[p\_tilestatus\_get\_by\_projectid]](#vjVE9Go4aDkBtZQTZoJWex+rCTw=)

[[dbo].[p\_tilestatus\_get\_by\_projectid\_paged]](#ZP9t21UM3jQkhPAjnryIGcmsaG0=)

[[dbo].[p\_tilestatus\_get\_by\_projectid\_with\_thumbnails]](#MMa4x8Vn5ChNWEY98YLQDM4wcRk=)

[[dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile]](#onMohemUZN4w8Ujofa/vIkPMLus=)

[[dbo].[p\_tilestatus\_get\_upgradable\_for\_project]](#eyljmpfDG1RdeUTtqr/BN0GN86U=)

[[dbo].[p\_tilestatus\_gettree\_by\_id]](#6VlWIweS4UGBOSOX46QlUJpkZ+Y=)

[[dbo].[p\_tilestatus\_update\_blobstatusid]](#rKXiHuqjFMw5RE7XRuNRz7uwpBs=)

|  |
| --- |
| [dbo].[fn\_­Get­Cache­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @cachename | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[fn\_­Get­Cache­Id]  (  @cachename nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @cache­Id int  set @cachename = UPPER(@cachename)  if @cachename = 'CACHE'  set @cache­Id = 0  else if @cachename = 'CONFIGURATION'  set @cache­Id = 1  else  set @cache­Id = -1  RETURN @cache­Id  END  GO |

Used By

[[dbo].[p\_cache\_load]](#x5NpJBEk3hMDw/mO6nOObbXG3r4=)

[[dbo].[p\_configuration\_load]](#4A8WPG9EXmbdkLYr/w11wlL7oSw=)

[[dbo].[p\_configuration\_set]](#DeIdr/PSElcRt6bzhuI+IlR1rlU=)

|  |
| --- |
| [dbo].[fn\_­Get­Officeconverter­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @officeconvertermachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: <Dennis Vroegop>  -- Create date: <27/02/2020>  -- Description: <Returns the available statusses >  -- =============================================  CREATE FUNCTION [dbo].[fn\_­Get­Officeconverter­Machine­Status­Id]  (  @officeconvertermachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @officeconvertermachinestatusid int  set @officeconvertermachinestatusdescription = UPPER(@officeconvertermachinestatusdescription)  if @officeconvertermachinestatusdescription = 'READY TO PROCESS'  set @officeconvertermachinestatusid = 0  else if @officeconvertermachinestatusdescription = 'PROCESSING'  set @officeconvertermachinestatusid = 1  else if @officeconvertermachinestatusdescription = 'COMPLETED'  set @officeconvertermachinestatusid = 2  else if @officeconvertermachinestatusdescription = 'OPERATION CANCELLED'  set @officeconvertermachinestatusid = 20  else if @officeconvertermachinestatusdescription = 'ERROR'  set @officeconvertermachinestatusid = 30  else  set @officeconvertermachinestatusid = -1  RETURN @officeconvertermachinestatusid  END  GO |

|  |
| --- |
| [dbo].[fn\_­Get­State­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @statemachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| create FUNCTION [dbo].[fn\_­Get­State­Machine­Status­Id]  (  @statemachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @statemachinestatusid int  set @statemachinestatusdescription = UPPER(@statemachinestatusdescription)  if @statemachinestatusdescription = 'NOT READY'  set @statemachinestatusid = 0  else if @statemachinestatusdescription = 'READY'  set @statemachinestatusid = 1  else if @statemachinestatusdescription = 'WORKING'  set @statemachinestatusid = 2  else if @statemachinestatusdescription = 'FINISHED'  set @statemachinestatusid = 10  else if @statemachinestatusdescription = 'ERROR'  set @statemachinestatusid = 11  else  set @statemachinestatusid = -1  RETURN @statemachinestatusid  END  GO |

Used By

[[dbo].[p\_canvasshotter\_book]](#GBecoS5NNIUwnYY10IG98Y4ntws=)

[[dbo].[p\_canvasshotter\_booking\_insert\_conditional]](#5iHeTvzxxnH/YtLAt0wKOhWh8Ls=)

[[dbo].[p\_canvasshotter\_fallback]](#w7g5EjCSqGzhgN064FP5YdNO6GM=)

[[dbo].[p\_canvasshotter\_history]](#N8r5uy8daNeo8TkSqrrpdiwTXaU=)

[[dbo].[p\_online\_user\_book]](#c5iw4Jj06eWGv8RAQS1V5t4MAv8=)

[[dbo].[p\_online\_user\_unbook\_forgot]](#CR8dfDUDNElJfvYHsUNNh1QwKbk=)

[[dbo].[p\_online\_user\_update\_keepalive]](#RGffZq+MHbZHzhLGgmYB7+RxXLQ=)

[[dbo].[p\_project\_copy\_book]](#Xc2uZUUHsCqQzKVu6toSxfLcKL8=)

[[dbo].[p\_project\_copy\_fallback]](#SZSN716jO+FTCfcstEifkqpd04k=)

[[dbo].[p\_project\_copy\_history]](#iP9nGEt2cxagal93V/gZLHBLNeA=)

[[dbo].[p\_project\_copy\_insert]](#IElOCneHQ59gypn3QPnkEKG6pSo=)

[[dbo].[p\_project\_upgrade\_book]](#ZtIHMxzkIaQqTAum/wUN5RfUzwk=)

[[dbo].[p\_project\_upgrade\_fallback]](#eyFMA0mMdoB7kA1BgRbWM2aeqpE=)

[[dbo].[p\_project\_upgrade\_history]](#TRfRFOGkYYT9dyRx+1BlOHqMJnk=)

[[dbo].[p\_project\_upgrade\_insert]](#2XDAj09j2hETAOl2Ohb7ZFRhJUU=)

[[dbo].[p\_tilestatus\_copy\_book]](#kzIlhS2mJFiJma1P3FyuIngtmj4=)

[[dbo].[p\_tilestatus\_copy\_fallback]](#FAozwoeeZq/SBEnuUGe8CMluFkI=)

[[dbo].[p\_tilestatus\_copy\_history]](#jaRkA76y/rG7Lw8O/YMA4asdxIs=)

[[dbo].[p\_tilestatus\_copy\_insert]](#dhfHL2WtimjmNW26GV2CNafaNn0=)

[[dbo].[p\_tilestatus\_update\_blobstatusid]](#rKXiHuqjFMw5RE7XRuNRz7uwpBs=)

[[dbo].[p\_tilestatus\_upgrade\_book]](#ld1zkb/fqpBa8wnM58UqebM1F4Q=)

[[dbo].[p\_tilestatus\_upgrade\_insert]](#cutcOfLrdt2GiPnsGtG+EKSAy/A=)

|  |
| --- |
| [dbo].[fn\_­Get­Tile­Type] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @tile­Type | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [dbo].[fn\_­Get­Tile­Type]  (  @tile­Type NVARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @tile­Typeid INT  SET @tile­Type = UPPER(@tile­Type)  IF @tile­Type = 'UNDEFINED'  SET @tile­Typeid = 0  ELSE IF @tile­Type = 'STICKY'  SET @tile­Typeid = 1  ELSE IF @tile­Type='IMAGE'  SET @tile­Typeid = 2  ELSE IF @tile­Type = 'PDF'  SET @tile­Typeid = 3  ELSE IF @tile­Type = 'GROUP'  SET @tile­Typeid = 4  ELSE IF @tile­Type = 'VIDEO'  SET @tile­Typeid = 5  ELSE IF @tile­Type = 'AUDIO'  SET @tile­Typeid = 6  ELSE IF @tile­Type = 'STACK'  SET @tile­Typeid = 8  ELSE IF @tile­Type = 'YOUTUBEVIDEO'  SET @tile­Typeid = 9  ELSE IF @tile­Type = 'WORD'  SET @tile­Typeid = 11  ELSE IF @tile­Type = 'EXCEL'  SET @tile­Typeid = 12  ELSE IF @tile­Type = 'POWERPOINT'  SET @tile­Typeid = 13  ELSE IF @tile­Type = 'TEXT'  SET @tile­Typeid = 16  ELSE IF @tile­Type = 'SHAPE'  SET @tile­Typeid = 17  ELSE IF @tile­Type = 'THUMBNAIL'  SET @tile­Typeid = 18  ELSE IF @tile­Type = 'INK'  SET @tile­Typeid = 19  ELSE IF @tile­Type = 'EMBED'  SET @tile­Typeid = 20  ELSE  SET @tile­Typeid = -1  RETURN @tile­Typeid  END  GO |

Used By

[[dbo].[p\_tilestatus\_get\_by\_projectid]](#vjVE9Go4aDkBtZQTZoJWex+rCTw=)

[[dbo].[p\_tilestatus\_get\_by\_projectid\_paged]](#ZP9t21UM3jQkhPAjnryIGcmsaG0=)

[[dbo].[p\_tilestatus\_get\_thumbnails\_for\_tile]](#onMohemUZN4w8Ujofa/vIkPMLus=)

[[dbo].[p\_tilestatus\_get\_upgradable\_for\_project]](#eyljmpfDG1RdeUTtqr/BN0GN86U=)

|  |
| --- |
| [licensing].[fn\_­Apply­Filter] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @operand1 | sql\_variant | 8016 |
| @operator | nvarchar(2) | 4 |
| @operand2 | sql\_variant | 8016 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [licensing].[fn\_­Apply­Filter]  (  @operand1 sql\_variant,  @operator nvarchar(2),  @operand2 sql\_variant  )  RETURNS BIT  AS  BEGIN  IF (@operator = '= ' AND @operand1 = @operand2) RETURN 1  ELSE IF (@operator = '> ' AND @operand1 > @operand2) RETURN 1  ELSE IF (@operator = '>=' AND @operand1 >= @operand2) RETURN 1  ELSE IF (@operator = '< ' AND @operand1 < @operand2) RETURN 1  ELSE IF (@operator = '<=' AND @operand1 <= @operand2) RETURN 1  ELSE IF (@operator = '<>' AND @operand1 <> @operand2) RETURN 1    RETURN 0  END  GO |

Uses

[licensing]

Used By

[[licensing].[p\_subscription\_search]](#lwhn7riTIvypHY0gkd39j/IiQ4w=)

|  |
| --- |
| [mft].[fn\_­Get­Chunking­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @chunkingmachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [mft].[fn\_­Get­Chunking­Machine­Status­Id]  (  @chunkingmachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @chunkingmachinestatusid int  set @chunkingmachinestatusdescription = UPPER(@chunkingmachinestatusdescription)  if @chunkingmachinestatusdescription = 'READY TO PROCESS'  set @chunkingmachinestatusid = 0  else if @chunkingmachinestatusdescription = 'PROCESSING'  set @chunkingmachinestatusid = 1  else if @chunkingmachinestatusdescription = 'READY TO DOWNLOAD'  set @chunkingmachinestatusid = 2  else if @chunkingmachinestatusdescription = 'DOWNLOAD COMPLETED'  set @chunkingmachinestatusid = 10  else if @chunkingmachinestatusdescription = 'READY TO CLEAN'  set @chunkingmachinestatusid = 11  else if @chunkingmachinestatusdescription = 'CLEANING'  set @chunkingmachinestatusid = 12  else if @chunkingmachinestatusdescription = 'CLEAN COMPLETED'  set @chunkingmachinestatusid = 13  else if @chunkingmachinestatusdescription = 'OPERATION CANCELLED'  set @chunkingmachinestatusid = 20  else if @chunkingmachinestatusdescription = 'ERROR'  set @chunkingmachinestatusid = 30  else  set @chunkingmachinestatusid = -1  RETURN @chunkingmachinestatusid  END  GO |

Uses

[mft]

Used By

[[mft].[p\_check\_download\_available]](#9OQhqSWAVHD+UBy5XRVnUeTSWWU=)

[[mft].[p\_chunkingmachine\_book]](#PnS53PCRBYBj/nxp1iARCISBVY8=)

[[mft].[p\_chunkingmachine\_completed]](#GbqFV8ZfXcU29GDG9v8bxnOw5Us=)

[[mft].[p\_chunkingmachine\_insert]](#f6LK3IajM3RnECpMSzPHWBs1WoU=)

[[mft].[p\_cleaningmachine\_bookchunking]](#MwISX6HBX5uwfbo4eD1sPM5Ba+w=)

[[mft].[p\_cleaningmachinenotcompleted\_bookchunking]](#DQUu0qPhdSDXCubN8KtH3sGAOos=)

[[mft].[p\_cleaningmachinenotcompleted\_bookchunkingmoving]](#BlF5NLkxVeX1k7IbVyiELdR7Yzc=)

[[mft].[p\_get\_activity\_from\_chunking]](#qXRdjSXddfxXIHJ7Eym6hAXQJDM=)

|  |
| --- |
| [mft].[fn\_­Get­Merge­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @mergemachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| CREATE FUNCTION [mft].[fn\_­Get­Merge­Machine­Status­Id]  (  @mergemachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @mergemachinestatusid int  set @mergemachinestatusdescription = UPPER(@mergemachinestatusdescription)  if @mergemachinestatusdescription = 'CLIENT UPLOADING'  set @mergemachinestatusid = -1  else if @mergemachinestatusdescription = 'READY TO PROCESS'  set @mergemachinestatusid = 0  else if @mergemachinestatusdescription = 'PROCESSING'  set @mergemachinestatusid = 1  else if @mergemachinestatusdescription = 'READY TO MERGE'  set @mergemachinestatusid = 2  else if @mergemachinestatusdescription = 'MERGE COMPLETED'  set @mergemachinestatusid = 10  else if @mergemachinestatusdescription = 'READY TO CLEAN'  set @mergemachinestatusid = 11  else if @mergemachinestatusdescription = 'CLEANING'  set @mergemachinestatusid = 12  else if @mergemachinestatusdescription = 'CLEAN COMPLETED'  set @mergemachinestatusid = 13  else if @mergemachinestatusdescription = 'OPERATION CANCELLED'  set @mergemachinestatusid = 20  else if @mergemachinestatusdescription = 'ERROR'  set @mergemachinestatusid = 30  else  set @mergemachinestatusid = -2  RETURN @mergemachinestatusid  END  GO |

Uses

[mft]

Used By

[[mft].[p\_check\_upload\_finished]](#88BhiFm5uHwIoRoVhAIAiPjw0sw=)

[[mft].[p\_cleaningmachine\_bookmergemoving]](#lr7RkXHqmwkD5T1jKtSL+cyVZn8=)

[[mft].[p\_cleaningmachine\_bookmerging]](#PcBMt0tIPqIsieGQG+6oFWBJhXQ=)

[[mft].[p\_cleaningmachinenotcompleted\_bookmerging]](#BGEQIhIkhDirIWgbHwqOmRwJtIk=)

[[mft].[p\_get\_activity\_from\_merge]](#3yUdhmqUMo+5TUAe4NtlDssCtbs=)

[[mft].[p\_mergemachine\_book]](#pNy6chimtMu1ga2KNNPvnYdyfzA=)

[[mft].[p\_mergemachine\_insert]](#iUuoicTqGrjxPJNYyujMR6ek43U=)

[[mft].[p\_mergemachine\_ready\_to\_process]](#Dh290Ssk5A7Vl7lE4ssUSvb0/To=)

|  |
| --- |
| [mft].[fn\_­Get­State­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @statemachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| create FUNCTION [mft].[fn\_­Get­State­Machine­Status­Id]  (  @statemachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @statemachinestatusid int  set @statemachinestatusdescription = UPPER(@statemachinestatusdescription)  if @statemachinestatusdescription = 'NOT READY'  set @statemachinestatusid = 0  else if @statemachinestatusdescription = 'READY'  set @statemachinestatusid = 1  else if @statemachinestatusdescription = 'WORKING'  set @statemachinestatusid = 2  else if @statemachinestatusdescription = 'FINISHED'  set @statemachinestatusid = 10  else if @statemachinestatusdescription = 'ERROR'  set @statemachinestatusid = 11  else  set @statemachinestatusid = -1  RETURN @statemachinestatusid  END  GO |

Uses

[mft]

|  |
| --- |
| [mft].[fn\_­Get­Storage­Machine­Status­Id] |

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Max Length (Bytes) |
| @storagemachinestatusdescription | nvarchar(50) | 100 |

SQL Script

|  |
| --- |
| -- =============================================  -- Author: <Author,,Name>  -- Create date: <Create Date, ,>  -- Description: <Description, ,>  -- =============================================  CREATE FUNCTION [mft].[fn\_­Get­Storage­Machine­Status­Id]  (  @storagemachinestatusdescription nvarchar(50)  )  RETURNS INT  AS  BEGIN  DECLARE @storagemachinestatusid int  set @storagemachinestatusdescription = UPPER(@storagemachinestatusdescription)  if @storagemachinestatusdescription = 'READY TO PROCESS'  set @storagemachinestatusid = 0  else if @storagemachinestatusdescription = 'PROCESSING'  set @storagemachinestatusid = 1  else if @storagemachinestatusdescription = 'COMPLETED'  set @storagemachinestatusid = 2  else if @storagemachinestatusdescription = 'OPERATION CANCELLED'  set @storagemachinestatusid = 20  else if @storagemachinestatusdescription = 'ERROR'  set @storagemachinestatusid = 30  else  set @storagemachinestatusid = -1  RETURN @storagemachinestatusid  END  GO |

Uses

[mft]

Used By

[[mft].[p\_check\_storageoperation\_finished]](#YWckhcm8yx4WBJg+di4+9EJyP0Y=)

[[mft].[p\_get\_activity\_from\_storage]](#9e7ywF3uhRnjGcJZ4e9+VeUE340=)

[[mft].[p\_storagemachine\_book]](#HjNI/SX3pG0VAK1u/2UQ3RjjiQc=)

[[mft].[p\_storagemachine\_insert]](#WiPK2k7Zanqsa370TGyEzfu2RLc=)

[[mft].[p\_storageoperations\_move\_history]](#HzyPp0rH2W7p+J3+QqOKqUUnxuM=)