**Set Query Timeout (SQL1)**

Preconditions:

* SQL Server 2008

Main Success Scenario:

1. In Management Studio, click "Tools" menu
2. Choose "Options..."
3. In "Options" window, select Query Execution > SQL Server > General
4. Set "Execution time-out"
5. Click "OK" button

Alternatives:

None

**Set Remote Query Timeout (SQL2)**

Preconditions:

* SQL Server 2008

Main Success Scenario:

1. In Management Studio, click right click server node
2. On “Server Properties” window, select “Connections”
3. Set “Remote query timeout”

Alternatives:

None

**Determine SQL Server Users (SQL3)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Determine SQL Server Users Remotely (SQL4)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Ouput Task Level SSIS Error (SQL5)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**See All Possible Non-Null Values in a Table (SQL6)**

Preconditions:

* Set <table name> to a table name

Main Success Scenario:

1. Run

Declare @LoopCounter Int

Declare @LoopMaxCount Int

Declare @Query Varchar(1000)

Select

@LoopMaxCount = COUNT(1)

From

INFORMATION\_SCHEMA.COLUMNS

Where

TABLE\_NAME = 'Ext\_HSGCRM\_FilteredLead'

Set @LoopCounter = 1

While(@LoopCounter <= @LoopMaxCount)

Begin

Select

@Query = 'Select Distinct ''' + TABLE\_NAME + ''' As TableName,[' + COLUMN\_NAME + '] From [' + TABLE\_NAME + '] Where [' + COLUMN\_NAME + '] Is Not Null Order By [' + COLUMN\_NAME + ']'

From

INFORMATION\_SCHEMA.COLUMNS

Where

TABLE\_NAME = 'Ext\_HSGCRM\_FilteredLead'

And

ORDINAL\_POSITION = @LoopCounter

Execute (@Query)

Set @LoopCounter = @LoopCounter + 1

End

Alternatives:

None

**How to Use Transaction (SQL7)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Merge (SQL8)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Output Clause (SQL9)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Create Data Type (SQL10)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Create Common Table Expression (SQL11)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use FileStream (SQL12)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Service Broker (SQL13)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Full Text Search (SQL14)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Spatial Data Type (SQL15)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Cross Apply (SQL16)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Use Outer Apply (SQL17)**

Preconditions:

* none

Main Success Scenario:

1. xxx

Alternatives:

None

**Determine Multi-Column Match Between Tables (SQL18)**

Preconditions:

* Use Intersect

Main Success Scenario:

1. xxx

Alternatives:

None

**Customize Default Error Messages (SQL19)**

Preconditions:

* Use sp\_AddMessage

Main Success Scenario:

1. xxx

Alternatives:

None

**Relational Table Partitioning (SQL20)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Determine Index to Use (SQL21)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Determine Error Code Meaning (SQL22)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Find Like Fields in Different Tables (SQL23)**

Notes: This will allow you to ensure that the same field in a different table has the same name, data type, and data size.

Preconditions:

* None

Main Success Scenario:

1. Find fields with same name but different specification
2. Find fields with different name but same specification
3. Find fields with same values

Alternatives:

None

**Standardize Like Columns (SQL23.5)**

Preconditions:

* None

Main Success Scenario:

1. Create dbo.Information\_Schema\_Columns table

Create Table dbo.Information\_Schema\_Columns

(

DatabaseName Varchar(50) Null

,SchemaName Varchar(10) Null

,TableName Varchar(50) Null

,ColumnName Varchar(50) Null

,CategoryName Varchar(50) Null

)

1. Create dbo.Information\_Schema\_Columns\_Standard

Create Table dbo.Information\_Schema\_Columns\_Standard

(

CategoryName Varchar(50) Null

,DataType Varchar(30) Null

,CharacterMaximumLength Int Null

,NumericPrecision Int Null

,NumericScale Int Null

,DateTimePrecision Int Null

)

1. Create Standard Column Categories

Insert Into

dbo.Information\_Schema\_Columns\_Standard

(

CategoryName

,DataType

,CharacterMaximumLength

,NumericPrecision

,NumericScale

,DateTimePrecision

)

Values

('PersonFullName','Varchar',30,null,null,null)

,('Flag','Char',3,null,null,null)

,('VisibleId','Int',null,null,null,null)

,('HiddenId','UniqueIdentifier',null,null,null,null)

,('WholeNumberCount','Int',null,null,null,null)

,('FractionalNumberCount','Float',null,null,null,null)

,('Dollars','Money',null,null,null,null)

,('DateTime','DateTime2',null,null,null,7)

,('Date','Date',null,null,null,null)

,('Week','Varchar',40,null,null,null)

,('Month','Varchar',25,null,null,null)

,('Quarter','Varchar',10,null,null,null)

,('Year','Varchar',10,null,null,null)

,('DimKey','Int',null,null,null,null)

,('URL','NVarchar',200,null,null,null)

,('DomainName','Varchar',255,null,null,null)

,('EmailAddress','Varchar',256,null,null,null)

,('PersonAlias','Varchar',64,null,null,null)

,('Description','NVarchar',-1,null,null,null)

,('EnglishOnlyName','Varchar',0,null,null,null)

,('PossibleUnicodeName','NVarchar',0,null,null,null)

,('PhoneNumberWo/Extension','Varchar',13,null,null,null)

-- (xxx)xxx-xxxx

,('PhoneNumberW/Extension','Varchar',22,null,null,null)

-- (xxx)xxx-xxxx Ext.xxxx

,('PhoneExtension','Varchar',4,null,null,null)

-- xxxx

,('StreetAddress','Varchar',xxx,null,null,null)

,('City','Varchar',xxx,null,null,null)

,('State','Varchar',xxx,null,null,null)

,('StateAbbreviation','Varchar',2,null,null,null)

,(‘Country','Varchar',xxx,null,null,null)

,(‘CompanyName','Varchar',100,null,null,null)

1. Create Standard Table Categories

xxx

1. Check what columns are not currently standardized

Select

Table\_Catalog

,Table\_Schema

,Table\_Name

,Column\_Name

From

Information\_Schema.Columns

Except

Select

DatabaseName

,SchemaName

,TableName

,ColumnName

From

dbo.Information\_Schema\_Columns

1. Check what tables are not currently standardized

xxx

1. xxx

Alternatives:

None

**Replace Null in UniqueIdentifier field (SQL24)**

Preconditions:

* None

Main Success Scenario:

1. Create GUID placeholder

Declare @NullAccountId UniqueIdentifier

Set @NullAccountId = NEWID()

1. Use placeholder only where field is null

IsNull(ow.AccountId,@NullAccountId) As AccountId

Alternatives:

None

**Profile SQL Server Query (SQL25)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Profile SQL Server (SQL26)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Profile Analysis Services (SQL27)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Grant IP access to SQL Azure from SQL Client (SQL28)**

Preconditions:

* None

Main Success Scenario:

1. Connect to SQL Azure server from SQL client

Server: aztatbcvb5.database.windows.net

SQL Login: ccsqlsvc@aztatbcvb5

SQL Password: …

1. Open a query window for the Master database
2. Run the query to allow a range of IPs

*EXEC sp\_set\_firewall\_rule @name = N'ContosoFirewallRule', @start\_ip\_address = '192.168.1.1', @end\_ip\_address = '192.168.1.10'*

Alternatives:

None

**Grant IP access to SQL Azure from Azure (SQL29)**

Preconditions:

* See [here](https://msdn.microsoft.com/en-us/library/azure/jj553530.aspx) or [here](https://support.office.com/en-ca/article/Connect-to-a-Microsoft-Azure-SQL-Database-9f621fd9-5f22-4c57-b247-eb0be8b7bac4) on how to do this

Main Success Scenario:

1. Open a private browser if connected to the MS network.
2. Connect to SQL Azure server

Server: <https://aztatbcvb5.database.windows.net/>

SQL Login: ccsqlsvc@aztatbcvb5

SQL Password: …

1. xxx

Alternatives:

None

**Grant IP access to SQL Azure from PowerShell (SQL30)**

Preconditions:

* See [here](https://msdn.microsoft.com/en-us/library/azure/jj553530.aspx) for guidance

Main Success Scenario:

1. Open a PowerShell window
2. xxx

Alternatives:

None

**Why Backup Analysis Services and not just the relational database**

* Security, KPI, or other settings are stored in Analytics
* Easy way to move the cube structure from one server to another
* If the cube processing takes a while due to calculations and aggregations, then it would be faster to restore from a backup
* If MOLAP storage is used then data is duplicated in the cube and is worth being backed up. If ROLAP or HOLAP store is used no data is duplicated in the cube.