

SQL Query Manager ++ (SQM++)

Over time when working with SQL queries and using different data bases, you have to relate to a lot of SQL queries, and probably use different tools to query the different sources. SQM++ is developed to have one central interface into your different data sources.

SQM++ has 2 basic purposes:

1. Help you in the work with SQL queries, especially when working with different data sources.

In this interface you can easily test, save and structure your SQL queries. Then it will also be easier to find your SQLs later, change or use them as samples for new queries.

Basic functions:

- Add SQL's using a defined connection
- Run the SQL to test it
- Save the SQL to keep it
- Structure your SQL's in different Workspaces (File folders)
- Run SQL's to make exports
- Run SQL's to alter databases
- Copy SQL's between Workspaces
- Find SQLs by name or content.

Supported databases are all databases that can be accessed using a connection string from an OLE DB or ADO DB connection.

It is mostly used with MS SQL server and Oracle (where you need the Oracle client).

In addition, it is well working with Microsoft Dynamics CRM using fetch XML, both on 365 and on premise.

It works also with MS SQL Azure and Text files.

The other main purpose with SQM++ is to:

2. Build you own Data warehouse

SQL Query Manager ++ can help you to build your own data warehouse that can collect data from different sources into one single database. Normally companies have different systems using different databases. It could be that you shall make reports combining data from an ERP system with data from your CRM system. The CRM system could be cloud based.

Having a local data warehouse where data is "cleaned" and aggregated, makes it easier for software like Excel or Reporting services to use it as data sources.

SQM ++ is designed to build the data warehouse on MS SQL server.

The queries you have stored in your workspace can be used as sources for creating the Data Warehouse.

Note that instead of creating and updating tables on SQL server, it is possible to create text files instead. Then you can create text files which for example can be imported to other systems.

Your Data Warehouse is kept updated by Scheduled tasks:

- The task has a SQL source (SQL Query and a connection string) to extract the data that you want to insert to your warehouse table
- The task has a target table name and a target connection string
- The task has a Schedule. To define how often and when it runs,

How to

Start SQL Query Manager ++

Logon as admin, leave password as is.

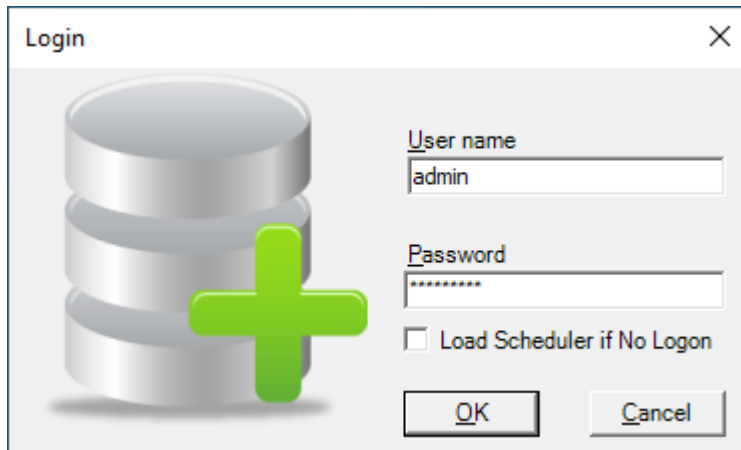


Figure: Logon

First time use on a computer, you get prompted to set the INI directory:

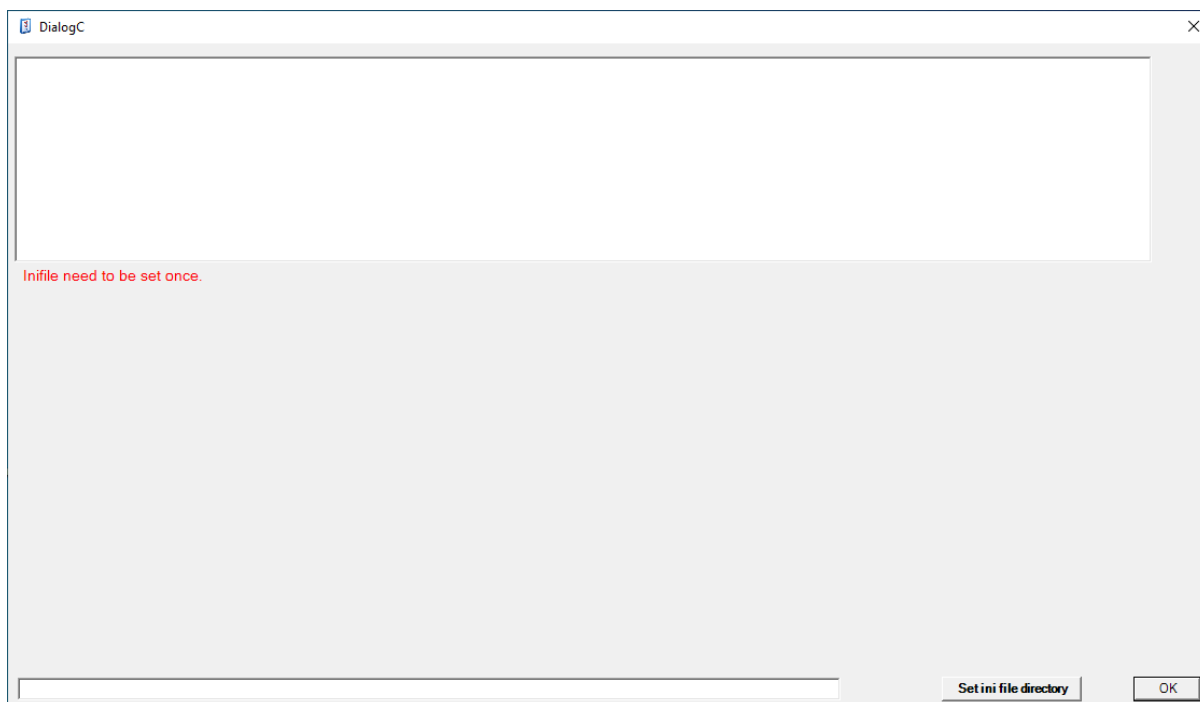


Figure: Set ini file

Click the button "Set ini file directory" (where the basic ini file is stored keeping track of all workspaces)

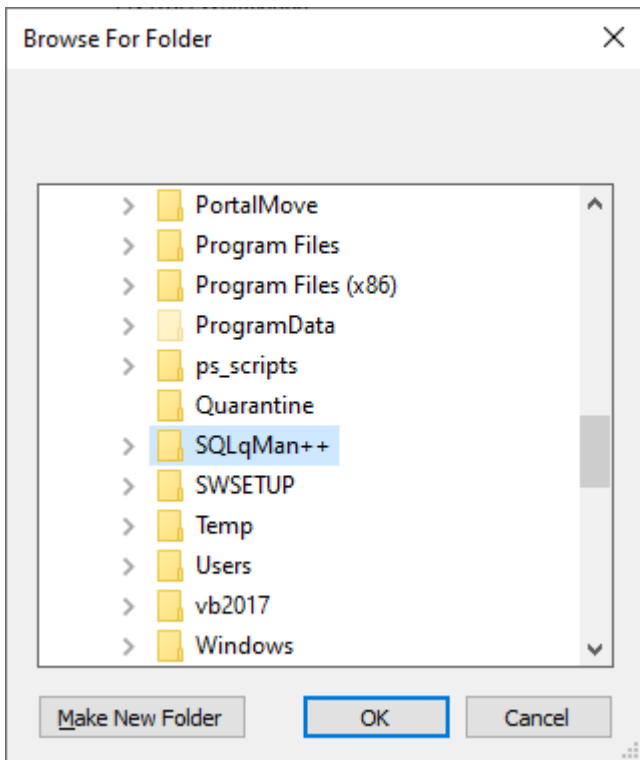


Figure: Browse Dialog, to select ini directory

Choose an existing directory or Create a new to be used. Click OK. And the path will come up in the dialog shown in Figure Browse Dialog. Click OK to set the path.

Then you need to set a Workspace where all SQL shall be stored. This is also a directory on your computer or a network share. A workspace is the same as a file folder, to help to structure your queries. Instead of having hundreds of queries in the same workspace/folder, it makes a better overview to structure the queries for example by source or system they are working with.

For example, you can have a separate workspace for ERP queries and another workspace for CRM queries. Or you can built it more detailed like a “ERP warehouse” workspace or a “ERP Finance” workspace. But don’t worry, queries can easily be copied between workspaces since it is hard to know the need in the start.

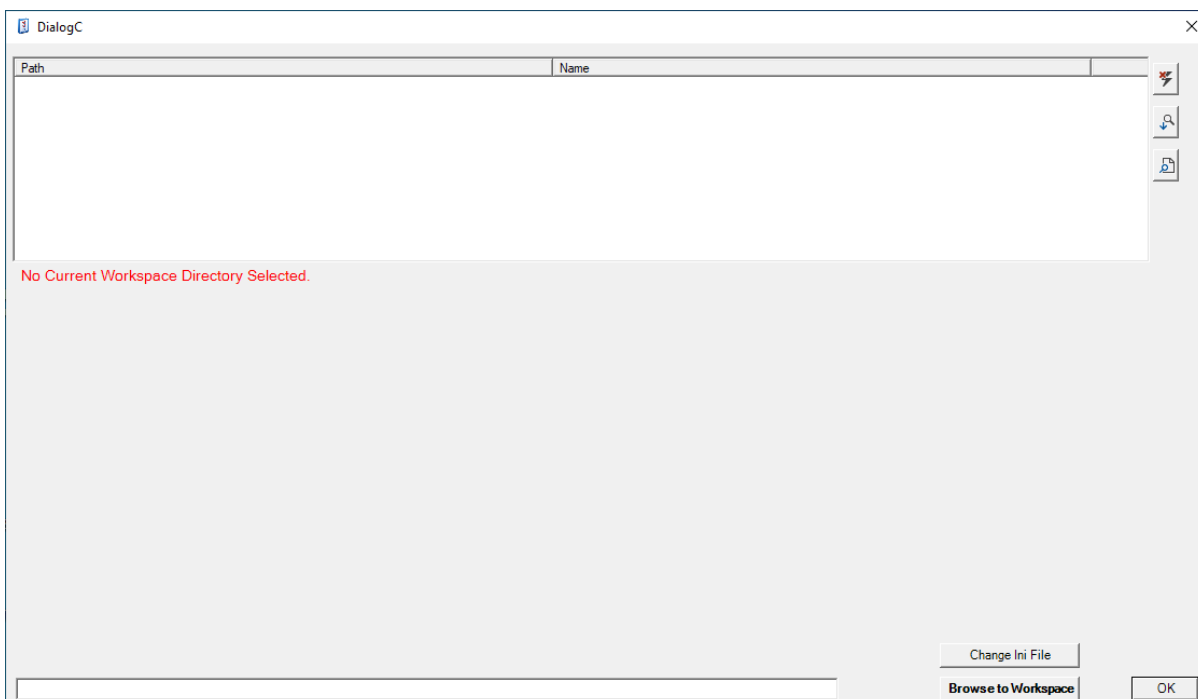


Figure: Set Workspace dialog

Click on the Button “Browse to Workspace”, and browse to a directory you will use as workspace directory:

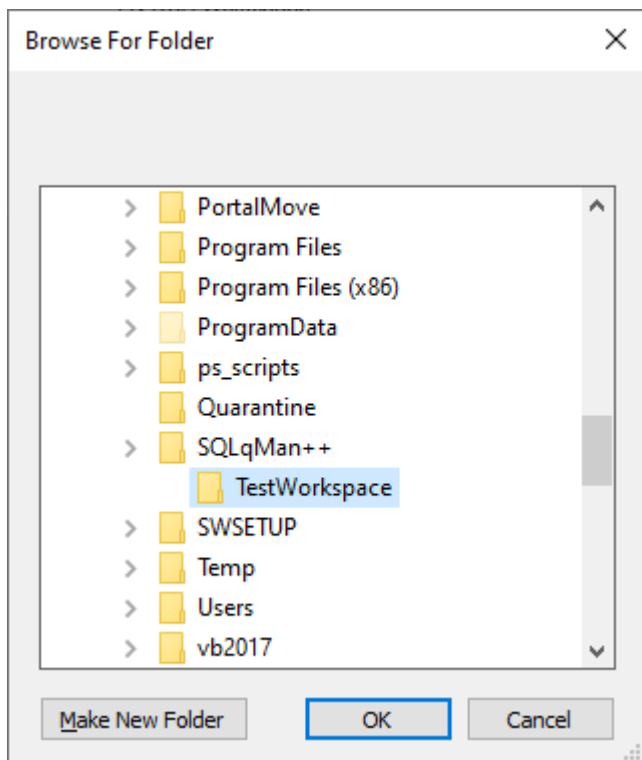


Figure: Browse Dialog

Click Ok,

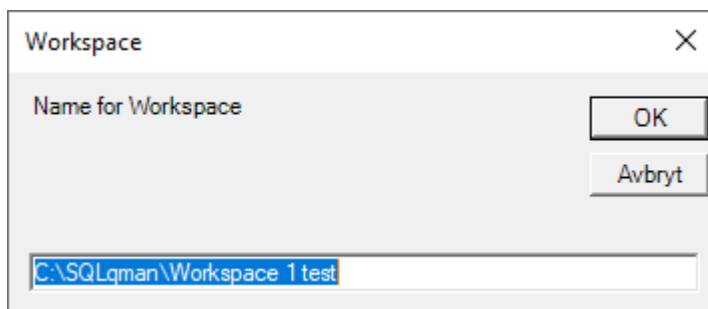


Figure. Name your Workspace

Then give the Workspace a name. The suggested name is the same as Path, but it can be changed.

These steps is only required at first time start up on a computer or for the user profile.

Work with queries:

SQL Query Manager ++ is not replacing graphical tools like MS SQL Management studio. Where you can see relations between tables in a graphical interface. When needing such interface, it is good to do them in other tools, and paste in the built SQL's to SQL Query Manager ++, where you have a structured place to save your queries.

Add a Connection to access your database

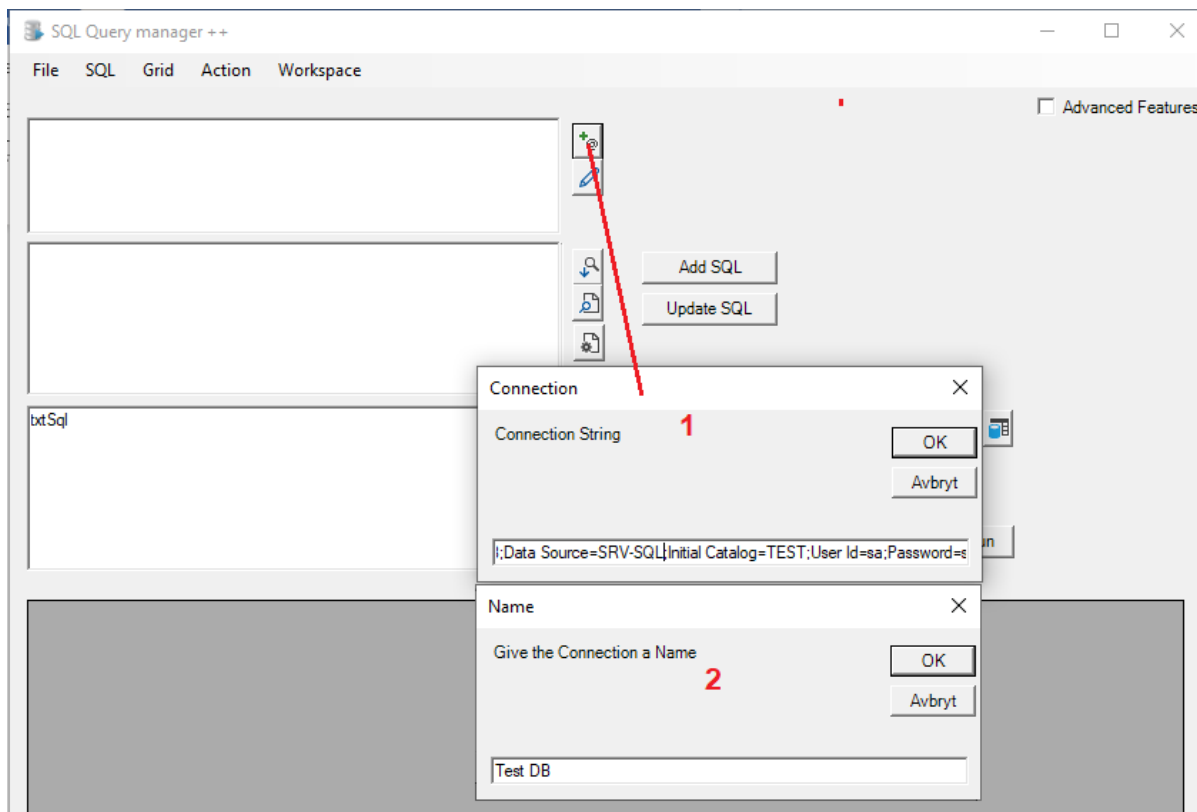


Figure: Add Connection string

Add a SQL Query

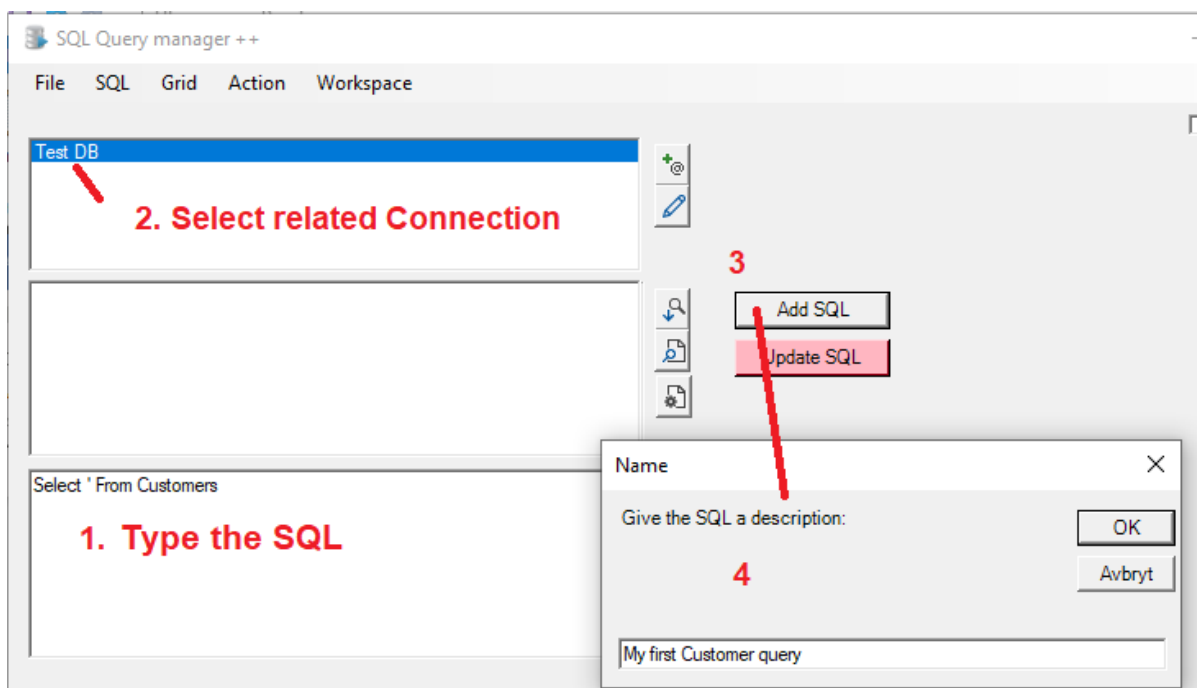


Figure: Add SQL query

The SQL is then visible in the SQL list:

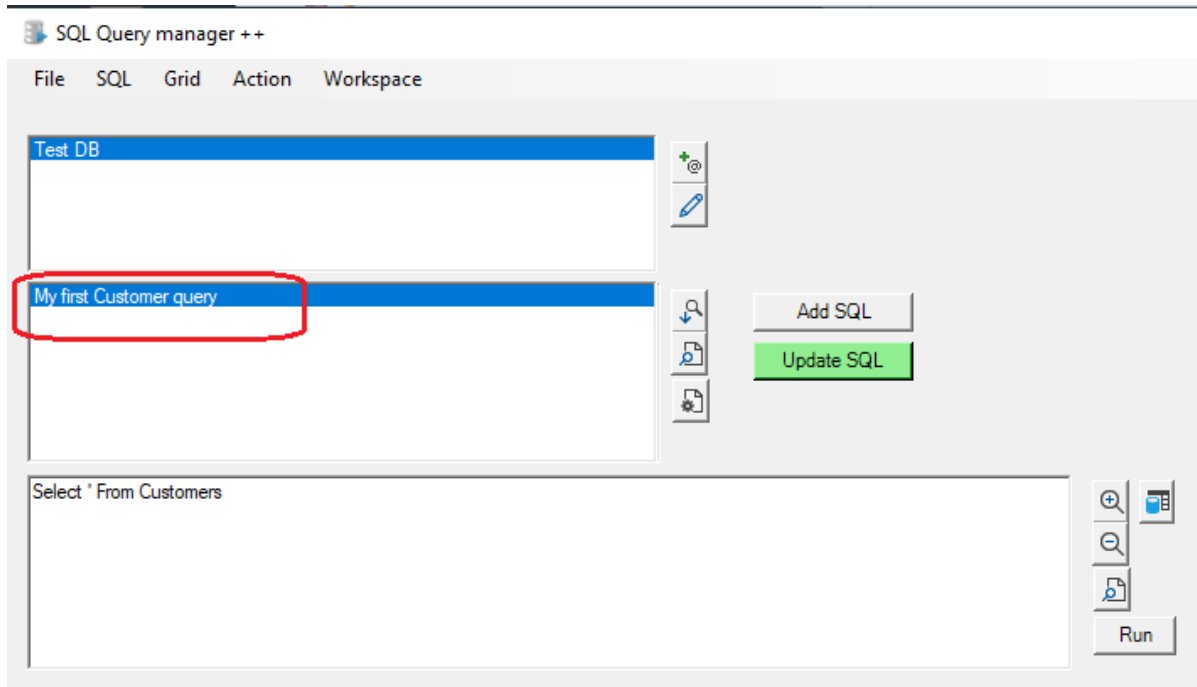


Figure: Your first query is saved

When changing a saved SQL, the Update Button becomes red. Click the Update button to Update the SQL and the button turns green.

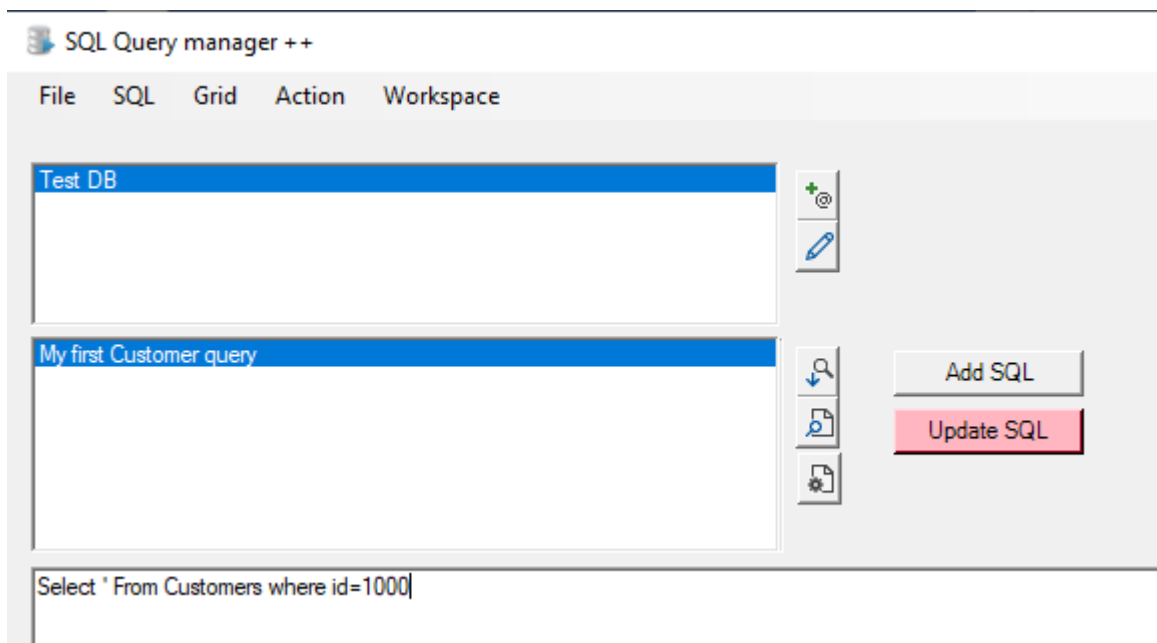


Figure: Change your query

Next sample is done by adding a connection to access Dynamics CRM in 365, then use a Fetch XML (the fetch xml syntax can be exported from Advanced Find in CRM). Then clicking the Run button will bring up the result in the Grid below.

SQL Query manager ++

File SQL Grid Action Workspace

CRM 365 cloud
Test DB

CRM Quotes
My first Customer query

Add SQL
Update SQL

```
<?xml version="1.0" output-format="xml-platform" mapping="logical" distinct="false">
  <entity name="quote">
    <attribute name="quoteid"/>
    <attribute name="quotenumber"/>
    <attribute name="revisionnumber"/>
    <attribute name="statecode"/>
    <attribute name="statuscode"/>
    <attribute name="totalamount"/>
    <attribute name="totalamount_base"/>
  </entity>
</?xml>
```

Run

quotenumber	totalamount_base	statuscode	revisionnumber	oppor_opportunityid	modifiedon	totalamount	createdon
QUO-14793-C3Y...	277643.2660	Canceled	1	629a6679-c340...	21.12.2020 01:31	340000.0000	21.12.2020 01:31
QUO-46055-R0F...	103665.5930	In Progress	0	9d507ee4-34f8-e...	21.12.2020 03:09	829380.0000	21.12.2020 03:09
QUO-46053-K8J...	290000.0000	Lost	1	2d3dc4bf-b93b-e...	11.02.2021 16:58	290000.0000	21.12.2020 07:00
QUO-46057-K3H...	22897.0000	Won	0	957db770-6a43-...	24.12.2020 11:03	22897.0000	21.12.2020 08:50
QUO-46058-X7G...	143000.0000	In Progress	0	fca20a56-9f46-e...	21.12.2020 11:13	143000.0000	21.12.2020 11:13
QUO-46061-S3C...	95607.0000	In Progress	0	f4f69628-8143-e...	21.12.2020 12:31	95607.0000	21.12.2020 11:50
QUO-46062-J0H...	500000.0000	Canceled	0	99cae37-0512-...	17.02.2021 07:08	500000.0000	21.12.2020 12:11
QUO-46064-K3P...	35395.6910	Won	0	844718b2-2a44-...	28.12.2020 09:47	281570.0000	22.12.2020 08:50
QUO-46066-Y5Z...	39126.0000	In Progress	0	577f8b1a-4644-e...	22.12.2020 13:33	39126.0000	22.12.2020 11:11
QUO-46068-G4N...	1180000.0000	In Progress	0	8b76dc0c-4241-...	23.12.2020 14:57	1180000.0000	23.12.2020 14:57
QUO-46070-R8S...	0.0000	Canceled	0	47c0b9f3-a645-e...	18.02.2021 10:42	0.0000	24.12.2020 05:30

Fetches: 11292 23.02.2021 11:29

Figure: Execute your query

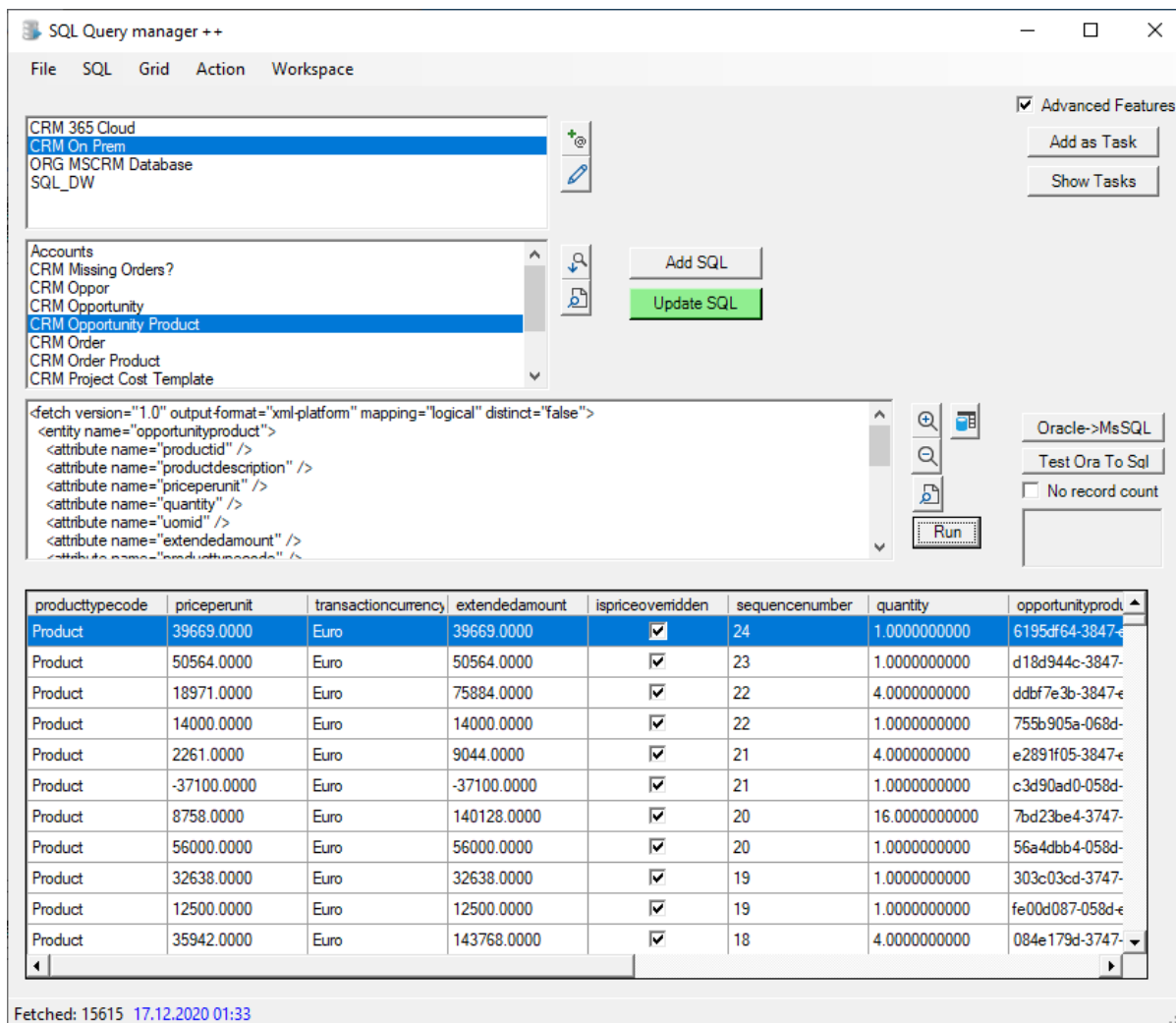


Figure: Another sample

Over time working with SQL Query Manager ++, you might get a lot of queries and it will be nice to have the possibility so search to find a query. There are two options; find the SQL by name or find it by content of the SQL statement.

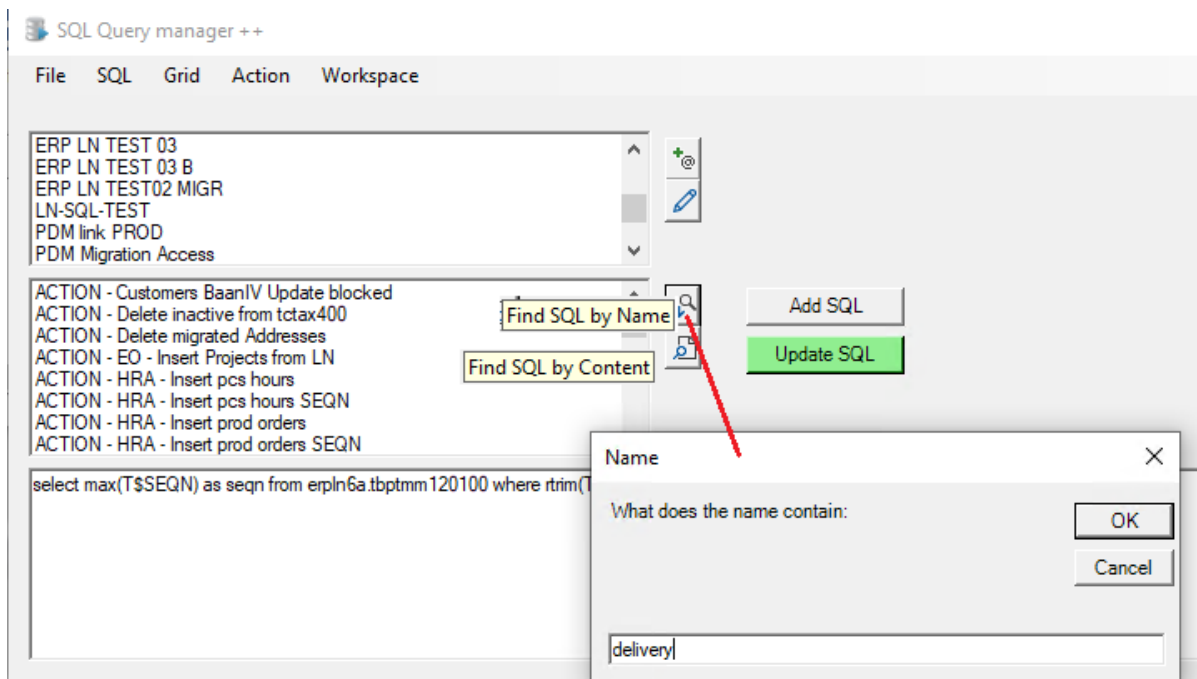


Figure: Find SQL by name or content

Build a SQL by testign and adding field.

In the next sample we will look on how to create the SQL and add fields from the grid. We start with select all columns from the table and if a large table, limit returned rows by using "TOP 10" to return only 10 rows for the test.

The screenshot shows the SQL Query manager ++ application. The top menu bar includes File, SQL, Grid, Action, and Workspace. On the left, a tree view shows 'My_Sql_DataWarehouse' and 'Cost' (with sub-items: Hours Accounting, prcMarket, Tables in Warehouse). The main query area contains the text: 'Select top 10 * from In_Cost_PCS'. To the right of the query area are buttons for 'Add SQL', 'Update SQL', 'Run', and 'Oracle->MsSQL'. Below the query area is a data grid with 8 columns: Description, Cost_Type, Budgeted_Amount, Estimated_Amount, Actual_Amount, Budgeted_Qty, Estimated_Qty, and ActualUnits. The grid displays 10 rows of data. At the bottom, a status bar shows the file path 'C:\Temp\SQLqMan-MyTestWorkspace\', 'Records: 10', 'Seconds used: 0', and the timestamp '10.03.2021 11:30'.

Description	Cost_Type	Budgeted_Amount	Estimated_Amount	Actual_Amount	Budgeted_Qty	Estimated_Qty	ActualUnits
Material Cost	2	0	176059.899	17830.18	0	678	11
Operation Cost	1	0	69969.8499	20895.2	0	85	62.333323
Material Cost	2	0	238102.7155	128.07	0	3839	8
Operation Cost	1	0	80410.2667	344.24	0	238	0.75
Material Cost	2	0	192588.2432	760	0	3590	1
Operation Cost	1	0	98914.8867	17869.22	0	290	45.283328
Material Cost	2	0	50358.2482	445.62	0	430.14	1

Figure: Select all columns

You can right click in the grid to add the Column name to the sql where the cursor is.

This screenshot shows the same SQL Query manager ++ interface, but with a context menu open over the data grid. The query area now contains the text: 'Select "Description", from In_Cost_PCS'. The context menu has the option 'Add Field To SQL' highlighted. The data grid shows the same 10 rows of data as the previous screenshot.

Description	Cost_Type	Budgeted_Amount	Estimated_Amount	Actual_Amount	Budgeted_Qty	Estimated_Qty
Material Cost	2	0	176059.899	17830.18	0	678
Operation Cost	1	0	69969.8499	20895.2	0	85
Material Cost	2	0	238102.7155	128.07	0	3839
Operation Cost	1	0	80410.2667	344.24	0	238

Figure: Add the columns you need

Another way to generate a base SQL for a table is to use the Tools Form, available from the File menu / Open Tools Form.

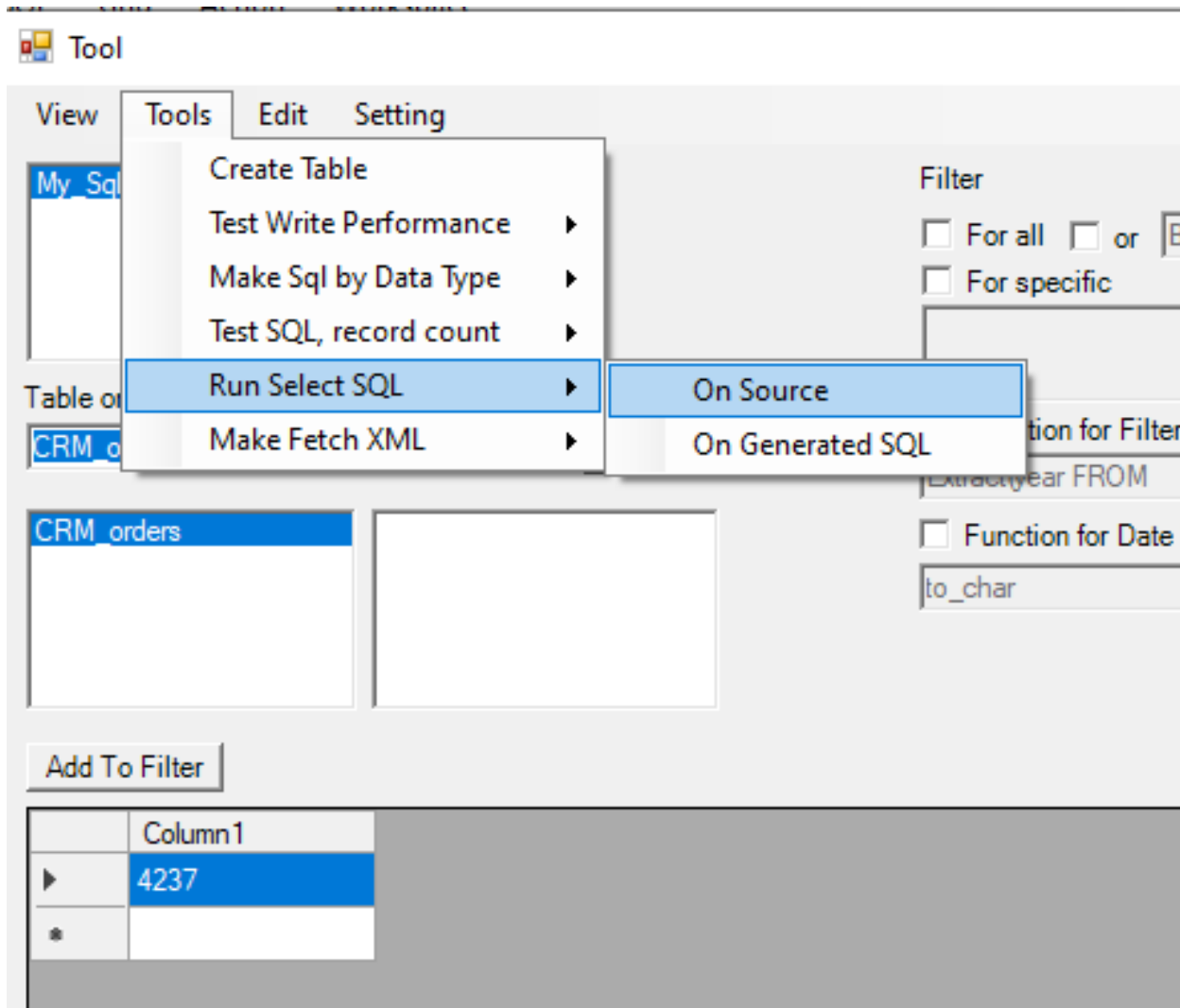
Type the table name, select the connection and press test to check if the table name and connection is set up properly.

The screenshot shows a software window titled "Tool" with a menu bar (View, Tools, Edit, Setting). The interface is divided into several sections:

- Connection List:** A list box at the top left contains "My Sql DataWarehouse", which is highlighted with a red "2" next to it.
- Table or Sql:** A text input field contains "CRM_orders", highlighted with a red "1" next to it. To its right is an unchecked checkbox labeled "Uppercase".
- Test Button:** A button labeled "Test" is highlighted with a red "3" next to it.
- Filter Section:** Located on the right, it includes:
 - Radio buttons for "For all" (unchecked) and "For specific" (unchecked), with an "or" label between them. A text field next to "For all" contains "Between 1 and 1900".
 - A text input field containing "Extract(year FROM".
 - Radio buttons for "Function for Filter Fields" (unchecked) and "Function for Date Fields" (unchecked).
 - A text input field containing "to_char".
- Add To Filter:** A button located below the "Table or Sql" section.
- Data Grid:** At the bottom, a table with two columns is shown:

	Column1
▶	4237
*	

You can run a SQL to select all columns from the table. Record count to be returned and displayed in the grid is default 100 and can be changed by the prompt.



Grid will be filled with the result:

Tool

View Tools Edit Setting

My_Sql_DataWarehouse

Table or Sql ☐ Uppercase

CRM_orders Test

Add To Filter

Filter

☐ For all ☐ or Between 1 and 1900

☐ For specific

☐ Function for Filter Fields

Extract(year FROM

☐ Function for Date Fields

to_char

Generated Sql

Add Sql Execute Log Result Update Sql Delete Sql

account__accountid	statecode	statuscode	sy	totalamount_base	oppor__opportunityid	ordernumber	transactioncurrency	submitdate	cu	ow	e	salesorderid	_margin
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-036...	Euro	8/17/2016 7:39:...	A...	D...	0	a2c06846b264-...	0
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-036...	Euro	8/17/2016 7:39:...	A...	D...	0	9dd5457b264e-...	0
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-036...	Euro	8/17/2016 7:40:...	A...	D...	0	3bf31279b264e-...	0
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-036...	Euro	8/17/2016 7:42:...	A...	D...	0	c1c3caa9b264-...	0
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-036...	Euro	8/17/2016 7:42:...	A...	D...	0	e0786ec3b264-...	0
060f2db3f1e6-e...	Active	New	E...	50460	76b5a5ad4317e...	ORD-027...	Euro	10/2/2014 12:06:...	A...	D...	0	4b34b29a-2c4a-...	0
935c3470-6684-...	Active	New	E...	45265.243	73b847b5-6684-...	ORD-046...	Chinese Yuan Re...	3/29/2018 1:48:...	B...	C...	4	391c91374332e-...	10
3b008b95f57e-...	Active	New	E...	106000	d452f613-0158e...	ORD-035...	Euro	4/22/2016 12:08:...	B...	D...	0	b51f6bf2-8208e-...	0
70216563fbc4e-...	Active	New	E...	67000	f806e803fcc4e...	ORD-023...	Euro	12/17/2013 2:16:...	B...	A...	0	3e58e5d7-2567-...	0
a59607cf-59ef-...	Active	New	E...	0	3c8f3af7-136e-...	ORD-029...	Indian Rupee	2/5/2015 8:32:2...	B...	C...	0	dd1ead83-11ad-...	0
dc0e526e-a7c1-...	Active	New	E...	73120	805a8cd0-9330-...	ORD-013...	Euro	12/27/2011 2:09:...	B...	M...	0	8fbbad68-9430e-...	0
dc0e526e-a7c1-...	Active	New	E...	73120	805a8cd0-9330-...	ORD-016...	Euro	8/13/2012 12:26:...	B...	M...	0	c39e321c-42e5-...	0
dc0e526e-a7c1-...	Active	New	E...	141850	4cb41bab-d2c3-...	ORD-026...	Euro	6/6/2014 3:01:2...	B...	M...	0	1e9d93d5-26ed-...	0
d5f79e20-9685e-...	Active	New	E...	0	98e64d74-9685-...	ORD-029...	Chinese Yuan Re...	12/30/2014 7:10:...	C...	C...	0	5f13c4e8f28f-e4...	0
a78af48a-4da3-...	Active	New	E...	130798.103	8b7f2f4a-9a3f-...	ORD-051...	Chinese Yuan Re...	12/30/2014 7:10:...	C...	C...	0	7257a774-0005e-...	0

Records: 100. Seconds used: 0

You can generate a SQL statement based on the table name you have typed or selected from saved table names.

You can add a filter from the grid where you have the test result.

Tool

View Tools Edit Setting

My_Sql_DataWarehouse

Table or Sql ☐ Uppercase

CRM_orders Test

Add To Filter

Filter

☐ For all ☐ or Between 1 and 1900

☒ For specific

[statuscode]='New'

☐ Function for Filter Fields

Extract(year FROM

☐ Function for Date Fields

to_char

account__accountid	statecode	statuscode	sy	totalamount_base	oppor__opportunityid	ordernumber
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-031
060f2db3f1e6-e...	Active	New	E...	93860	76a98e6c-7443-...	ORD-031

Figure: Add filter based on grid sample

Then you can generate the SQL statement that will include the filter above:

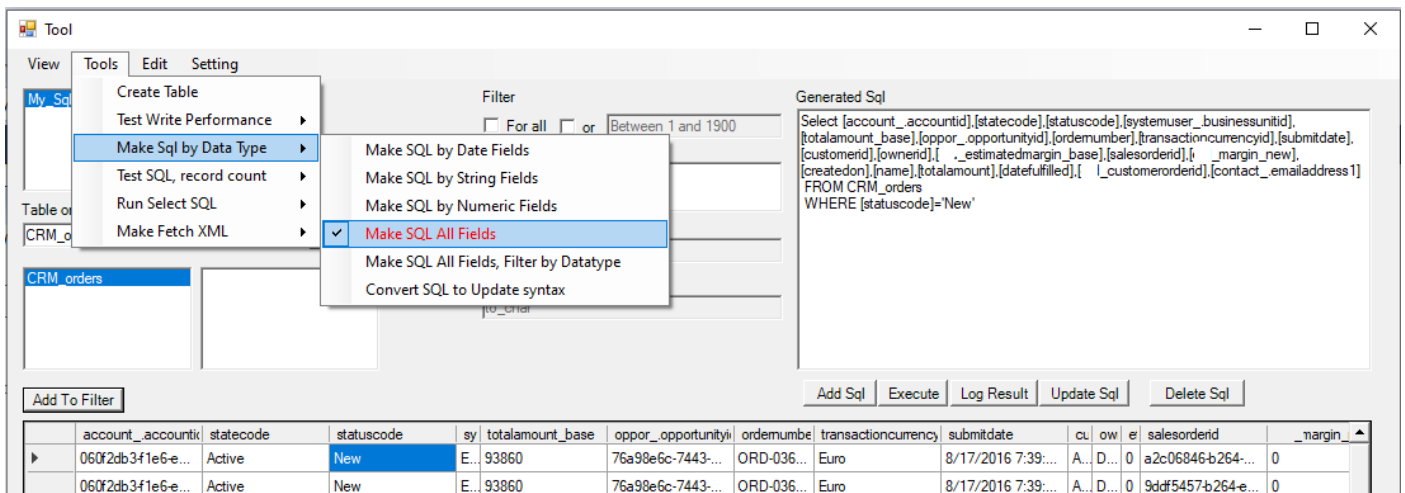


Figure: Generate a SQL

Other Basic functions

When working with queries towards different sources the number of queries can be high. So then it could be smart to arrange your queries into different Workspaces or physical folders on a network share. From the Workspace menu you can select Switch Workspace.

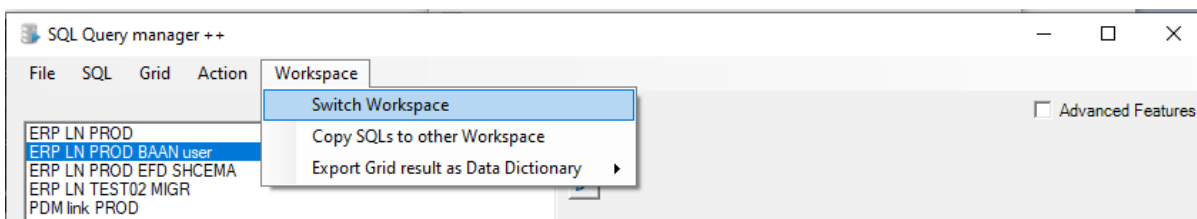


Figure: Switch Workspace

Then you will get up a Dialog where you can Switch to another Workspace or Create a New Workspace. If Creating new workspace you will get a browse folder dialog, and then be prompted for a Workspace name (like figure Browse Dialog).

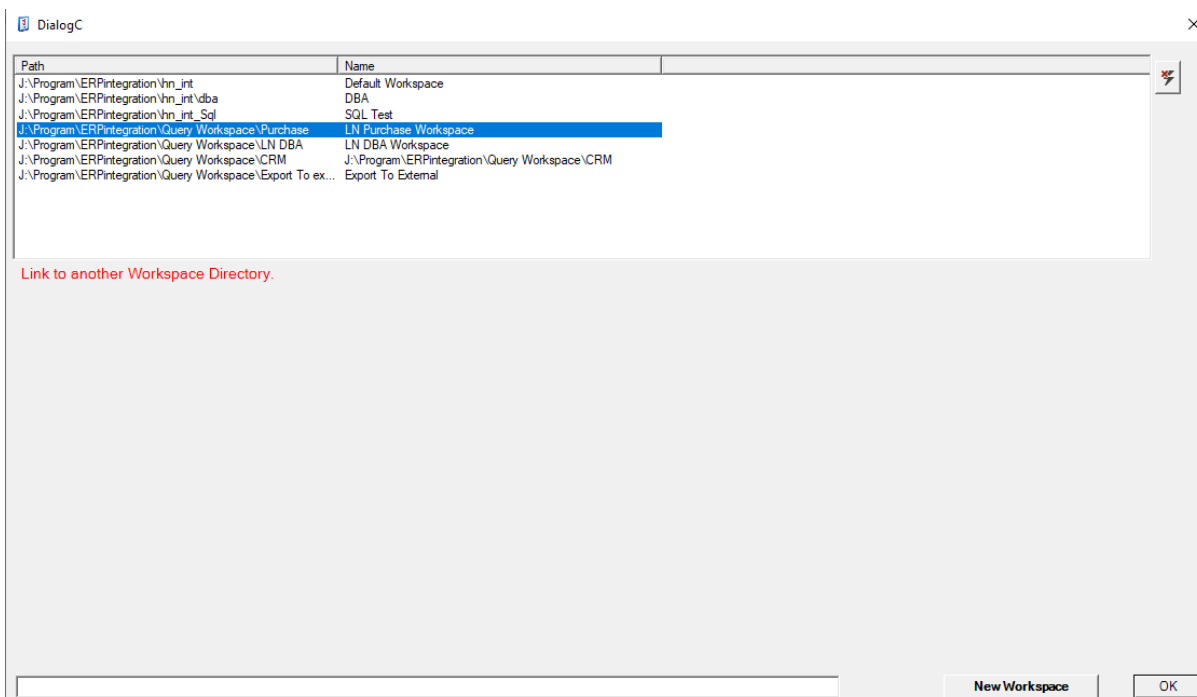


Figure: Select the Workspace

Functions related to the SQL edit box

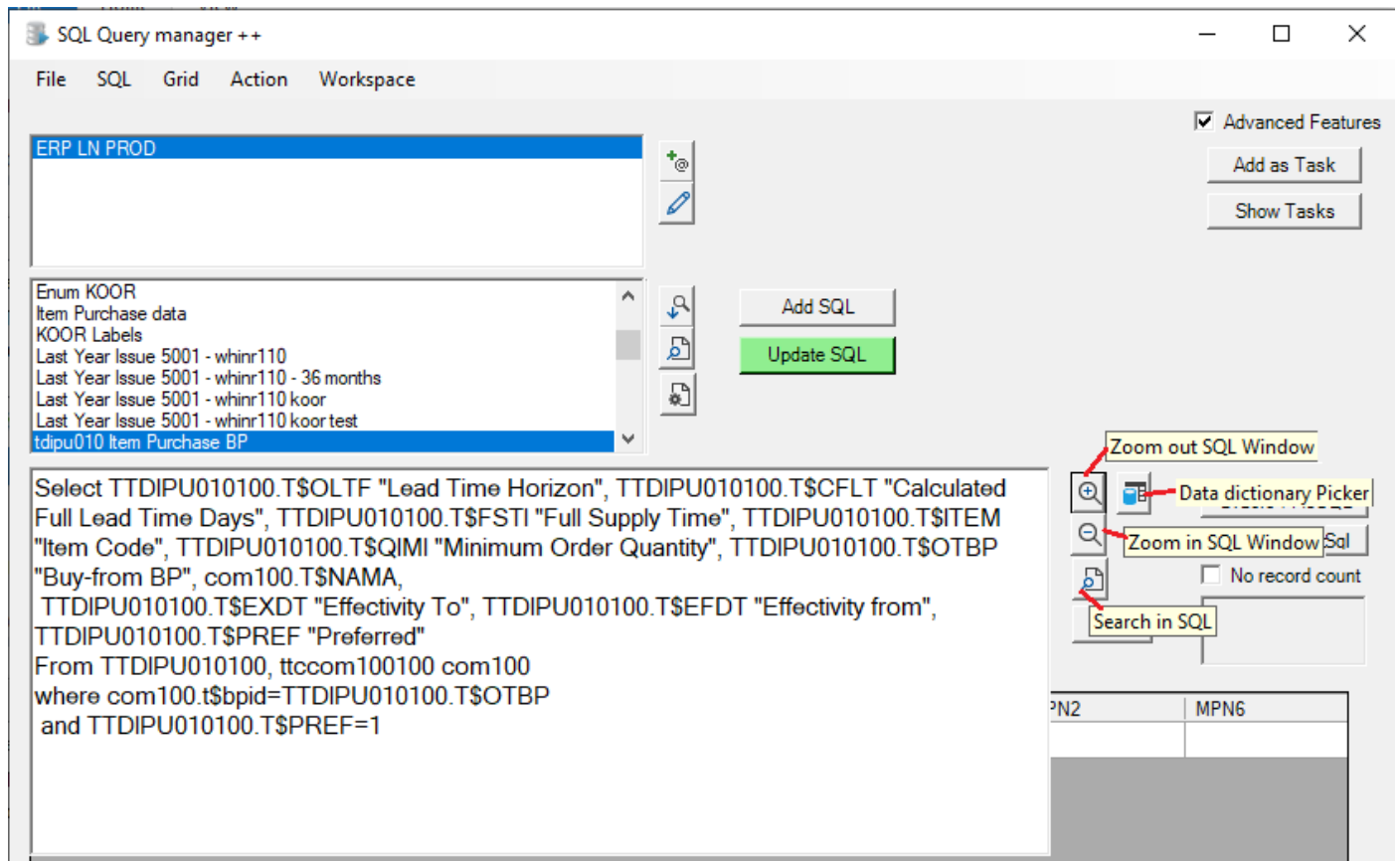


Figure: SQL editor functions

Zoom out to get a bigger edit box with bigger text or Zoom in, to get it smaller. Search in the SQL to highlight the word you are searching for.

Data dictionary picker

The Data dictionary picker is a dialog where you can find tables, select the table and fields to generate a SQL statement instead of writing it or copy it from another Tool.

To use the data dictionary picker, you have to create 2 ini files, one for all your table fields and one for all tables.

SQL Query Manager ++ can create these files if you can make a query to extract the data dictionary information:

SQL Query manager ++

File SQL Grid Action Workspace

ERP LN PROD

Switch Workspace
Copy SQLs to other Workspace
Export Grid result as Data Dictionary

Table descriptions
Field descriptions
Remove Dictionary
Copy Dictionary to other Workspace

Update SQL

Advanced Features
Add as Task
Show Tasks

LN DBA - Labels
LN DBA - Tables
LN DBA - ttadv422 Fields
LN DBA - ttadv422 Fields 0
LN DBA - User defaults test
Workspace Field Dictionary
Workspace Table dictionary
Workspace Table dictionary test

Select "Table", "T\$" || Upper("Field Name") "Field Name", Case When "Label" <> '' then "Label"
When "Field Label" <> '' then "Field Label" When "Domain Label" <> '' then "Domain Label" END
"Field Description",
"Ref Field"
FROM
(Select 'T' || upper(adv422.T\$CPAC || adv422.T\$CMOD || adv422.T\$FLNO) || '100' "Table", (adv422.T\$CPAC || adv422.T
\$CMOD || adv422.T\$FLNO || '.' || adv422.T\$FDNM) "Related Domain",
adv422.T\$FDNO, adv422.T\$FDNM "Field Name",
case when adv422.T\$REFD <> '' then

Run

Oracle->MsSQL
Test Ora To Sql
No record count

Table	Field Name	Field Description	Ref Field
TTCCOM100100	T\$INET	Website	
TTCCOM100100	T\$INRL	Affiliated Company	
TTCCOM100100	T\$ISCN	Affiliated Compan...	
TTCCOM100100	T\$LGID	Legal Identification	
TTCCOM100100	T\$LMDDT	Last Modification ...	
TTCCOM100100	T\$LMUS	Last Modified by	
TTCCOM100100	T\$LVDT	Obsolete	

J:\Program\ERPIntegration\Query Workspace\LN DBA\ Records: 61. Seconds used: 17 10.03.2021 11:30

FieldDictionary.ini - Notisblokk

File Rediger Format Vis Hjelp

```
[TTDPUR401100]
T$PMDE=Price Matrix
T$CFRW=Carrier/LSP
T$BPTC=BP Tax Country
T$PMNI=Payment (Internally)
T$PMND=Payment (Direct Delivery)
T$ACTI=Status Purchase Ord. Line
T$PONO=Position Number
T$CSGP=Purchase Statistics Group
T$SFBP=Ship-from BP
T$PMSE=Price Matrix Sequence
T$SBIM=Self-Billing
T$CPAY=Pmt Terms
T$ADIN=Additional Information
T$BTX2=Backorder Text
T$CMBI=Country/Tax
T$ORNO=Purchase Order
T$MPSN=Manufacturer Part Number Set
T$DMSE=Discount Matrix Sequence
T$CPCP=Cost Component
T$CENO=Exempt Certificate
T$RCOD=Exempt Reason
T$CRRF=Item Cross Reference
T$CRIT=Cross Reference Item
T$PAFT=Invoice after
T$PMNT=Payment
T$ISSP=Invoice by Stage Payments
T$CMBF=MPN and Manufacturer
T$MPND=...
```

Ln 1, Kol 1 100% Windows (CRLF) ANSI

TableDictionary.ini - Notisblokk			
Fil Rediger Format Vis Hjelp			
[Tables]			
TBPMDM000100=People Parameters			
TBPMDM080100=Employee Exams			
TBPTMM120100=Employee Work Time Production			
TBPTMM130100=Employee Work Time Service Order			
TBPTMM270100=History PCS Project Data			
TBRINH999100=Cross Dock Order Types			
TBRKAN001100=Kanban Item Information			
TBRKAN052100=Kanban Requests			
TBRKAN063100=Temporary Kanban Requests - Replenishment			
TBRMCS999100=Warehouse Mobility - General Parameters			
TBRMCS999100=Warehouse Mobility - Hours Accounting Parameters			
TBRMCS999100=Employees by Team - Jobs			
TCISLI000100=Invoicing Parameters			
TCISLI001100=Invoicing Parameters by Source Type			
TCISLI107100=Billing Requests by Billing Request Set			
TCISLI155100=Invoice Layout - Details			
TCISLI159100=Invoice Layouts by Business Partner			
TCISLI204100=Obsolete			
TCISLI210100=Interest Invoice Details			
TCISLI227100=Tax Only Data (Multiple Tax)			
TCISLI240100=Sales and Warehouse Order Invoice Headers			
TCISLI260100=Service Contract Headers			
TCISLI295100=Rebate Invoice Lines			
TCISLI302100=Interim Business Partner Balances			
Ln 1, Kol 1		100%	Windows (CRLF) UTF-8

When these files are created, you can use the data dictionary picker to create SQLs.

Table Name	Table Description	Field Name	Data Type	Field Description	Reference Table and Field
TTDSLS419100	Sales Order Line Confirmations	T\$ORNO	System.String	Sales Order	TTDSLS400.T\$ORNO
TTDSLS453100	Sales Order Line Components History	T\$PONO	System.Decimal	Position Number	(none)
TTDSLS466100	Sales Order Line Component Actual D...	T\$SQNB	System.Decimal	Sequence	(none)
TTDSLS467100	Sales Order Line Component Surplus	T\$DSQN	System.Decimal	Delivery	(none)
TTSMSC120100	Maintenance Sales Order Coverage L...	T\$INVL	System.Decimal	Invoice Line	(none)
TTSMSC810100	History - Maintenance Sales Order It...	T\$STAT	System.Decimal	Invoice line status	(none)
TTDSLS401100	Sales Order Lines	T\$OSEQ	System.Decimal	Position Number	(none)
TTDSLS406100	Sales Order Actual Delivery Lines	T\$DLDT	System.DateTime	Delivery Date	(none)
TTDSLS411100	Sales Order Line Item Data	T\$DLDT	System.DateTime	Delivery Log Date	(none)
TTDSLS457100	Sales Order Line Component Actual D...	T\$QIDL	System.Decimal	Delivered Quantity	(none)
TTDSLS469100	Sales Order Line Component Actual D...	T\$SHPM	System.String	Shipment ID	(none)
TTDSLS801100	Sales Orders Integrations	T\$SHLN	System.Decimal	Shipment Line	(none)
TTDSLS404100	Transfer Orders by Sales Order Line	T\$RCID	System.String	Receipt ID	(none)
TTDSLS415100	Sales Order Estimated COGS	T\$RCID	System.Decimal	Receipt Line	(none)
TTDSLS452100	Sales Order Installment History	T\$CLCT	System.String	Obsolete	(none)
TTDSLS454100	Sales Order Main Items History	T\$SERL	System.String	Obsolete	(none)
TTSMSC110100	Maintenance Sales Order Item Lines	T\$SSSN	System.String	Lot and Serial Set	TTICBD410.T\$SSSN
TTSMSC820100	History - Maintenance Sales Order Co...	T\$DELN	System.String	Delivery Note	(none)
TTDSLS416100	Sales Order Material Supply Lines	T\$GRWT	System.Decimal	Gross Weight	(none)
TTDSLS454100	Sales Order Line History	T\$CWUN	System.String	Weight Unit	TTCMCS001.T\$CWUN
TTDSLS464100	Sales Order Line Component Potential...	T\$LNCL	System.Decimal	Linked to Contract	(none)
TTDSLS465100	Sales Order Line Component Estimate...	T\$CNIG	System.Decimal	Contract Ignored	(none)
TTDSLS471100	External Reference Table Sales Order...	T\$PRIC	System.Decimal	Price	(none)
TTDSLS491100	Sales Order Header Copy Exceptions	T\$CUPS	System.String	Unit	TTCMCS001.T\$CUNI
TTSMSC111100	Maintenance Sales Order Activity Lines	T\$CVPS	System.Decimal	Conversion Factor	(none)

Select the Fields. Press ctrl while clicking for multiple selection

Add To Sql

Select TTDSLS406100.T\$ORNO "Sales Order", TTDSLS406100.T\$PONO "Position Number", TTDSLS406100.T\$SQNB "Sequence", TTDSLS406100.T\$DLDT "Delivery Date", TTDSLS406100.T\$QIDL "Delivered Quantity", TTDSLS406100.T\$PRIC "Price", TTDSLS406100.T\$CUPS "Unit" From TTDSLS406100

Return Query

Build you own Data warehouse

SQL Query Manager ++ can help you to build your own data warehouse that can collect data from different sources into on single database. Normally companies have different systems using different databases. It could be that you shall make reports combining data from an ERP system with data from your CRM system. The CRM system could be cloud based.

Having a local data warehouse where data is “cleaned” and aggregated, makes it easier for software like Excel or Reporting services to use it as data sources.

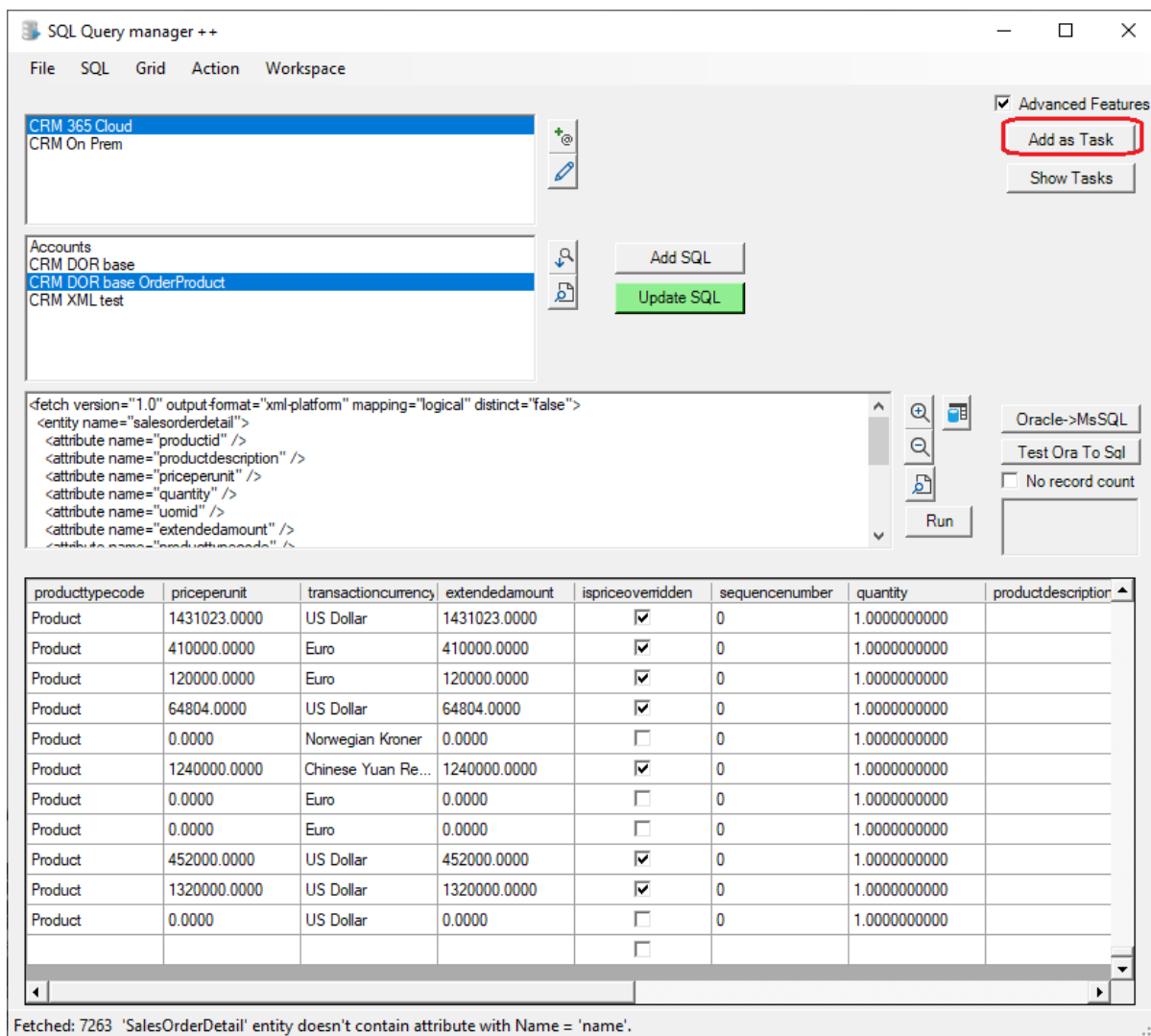
SQM ++ is designed to build the data warehouse on MS SQL server.

The queries you have stored in your workspace can be used as sources for creating the Data Warehouse.

Note that instead of creating and updating tables on SQL server, it is possible to create text files instead. Then you can create text files which for example can be imported to other systems.

How to set up a Task

This is done by selecting the saved SQL you want to use and press the “Add as Task” button.



SQL Query manager ++

File SQL Grid Action Workspace

CRM 365 Cloud
CRM On Prem

Accounts
CRM DOR base
CRM DOR base OrderProduct
CRM XML test

Add SQL
Update SQL

Advanced Features
Add as Task
Show Tasks

Oracle->MsSQL
Test Ora To Sql
No record count
Run

```
<fetch version="1.0" output-format="xml-platform" mapping="logical" distinct="false">
  <entity name="salesorderdetail">
    <attribute name="productid" />
    <attribute name="productdescription" />
    <attribute name="priceperunit" />
    <attribute name="quantity" />
    <attribute name="uomid" />
    <attribute name="extendedamount" />
    <attribute name="ispriceoverridden" />
    <attribute name="sequencenumber" />
  </entity>
</fetch>
```

producttypecode	priceperunit	transactioncurrency	extendedamount	ispriceoverridden	sequencenumber	quantity	productdescription
Product	1431023.0000	US Dollar	1431023.0000	✓	0	1.0000000000	
Product	410000.0000	Euro	410000.0000	✓	0	1.0000000000	
Product	120000.0000	Euro	120000.0000	✓	0	1.0000000000	
Product	64804.0000	US Dollar	64804.0000	✓	0	1.0000000000	
Product	0.0000	Norwegian Kroner	0.0000	✗	0	1.0000000000	
Product	1240000.0000	Chinese Yuan Re...	1240000.0000	✓	0	1.0000000000	
Product	0.0000	Euro	0.0000	✗	0	1.0000000000	
Product	0.0000	Euro	0.0000	✗	0	1.0000000000	
Product	452000.0000	US Dollar	452000.0000	✓	0	1.0000000000	
Product	1320000.0000	US Dollar	1320000.0000	✓	0	1.0000000000	
Product	0.0000	US Dollar	0.0000	✗	0	1.0000000000	

Fetched: 7263 'SalesOrderDetail' entity doesn't contain attribute with Name = 'name'.

Figure: Add SQL as Task. The SQL will be used as the source for the new task.

The first time you add a task, you need to specify the Task directory. The task directory is a separate directory where the source SQL is copied to. To not be affected by eventually changing the SQLs in the workspace later.

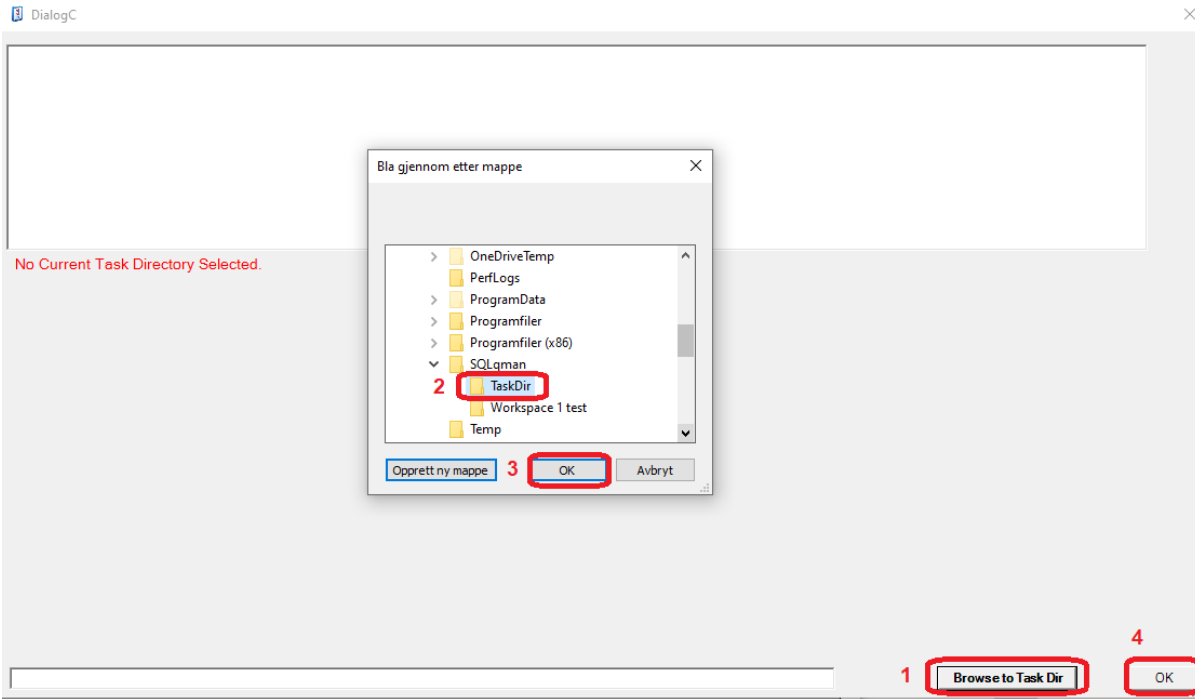


Figure: Set Task directory.

After clicking OK, you are back to the main window (figure 14) and need to click “Add as Task” once more. Note that this is a onetime initial setting, so next time you add a task you don’t need to set the task directory,

Then the Task Dialog comes up where you define more settings for the task, like source connection string (which should be the same as for the Query used as source). The target table name and a target connection string should also be defined. This is the Database on your SQL server that you want to use as your data warehouse. The task needs a Schedule defining when and how often it runs, to update the target table in the data warehouse.

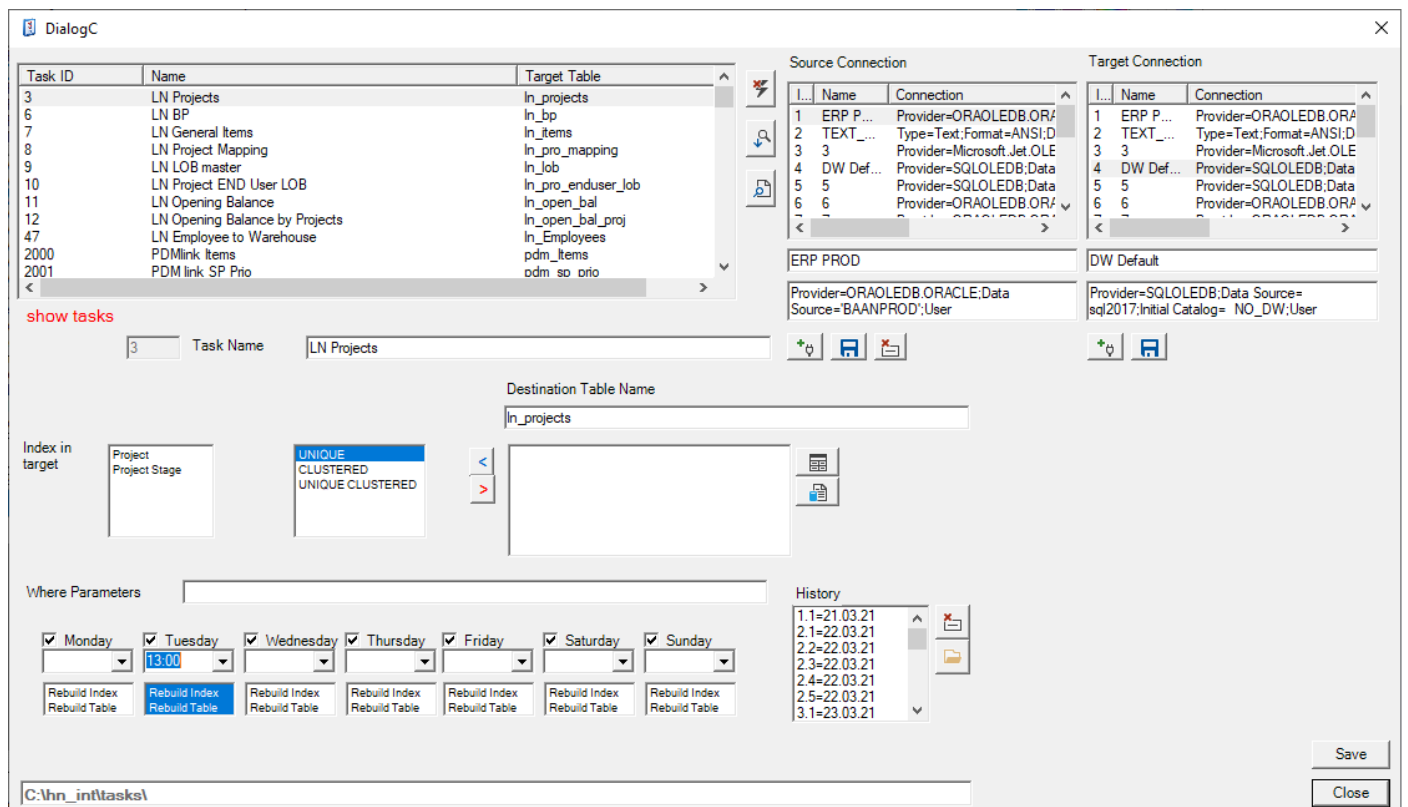


Figure: Task setup.

Select the Source Connection and Target Connection. Initially you need to add Target connection which is the connection string to the SQL server database to use as Warehouse database. Type in a logical connection name and the connection string with provider type, name of SQL server, database name, user id and password having necessary write access to the database. And then click the “Add Target Connection” button to Save it.

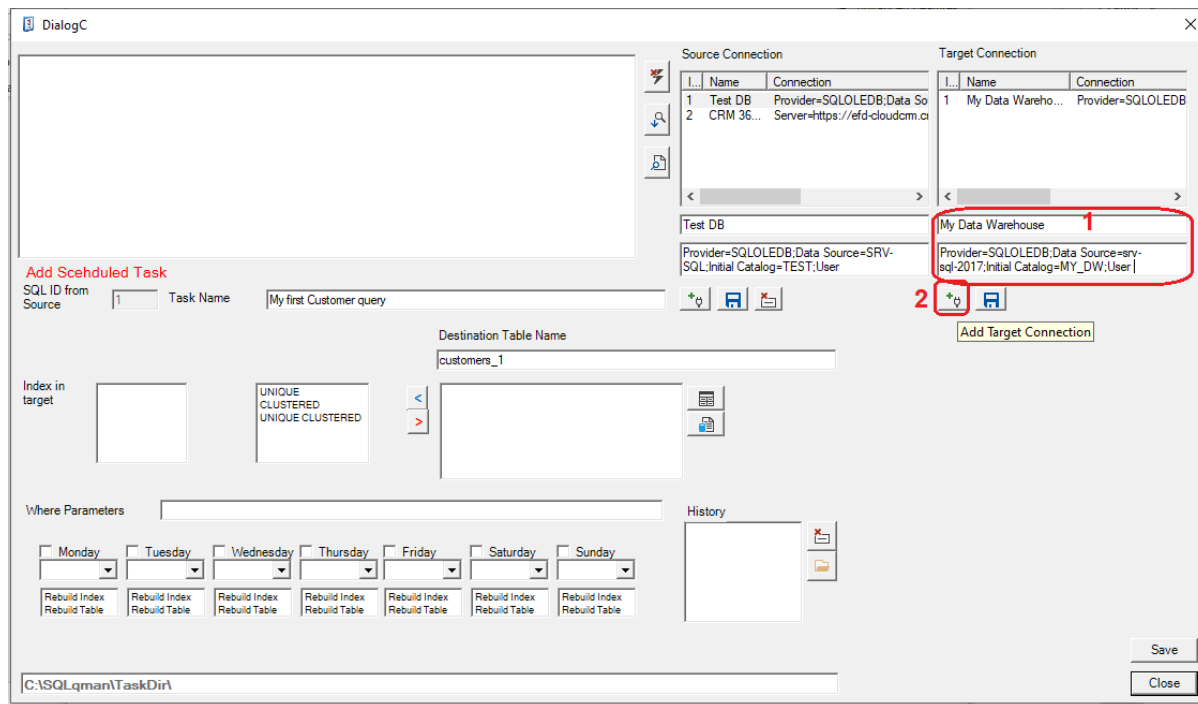


Figure: Add Target Connection

At the bottom of the Task setup dialog you have to tick the Days you want the task to be executed. 06:00 is default time. You can delete this and add other times. Below the time dropdown you can select to Rebuild Table and/or Rebuild Index.

Rebuild Table needs to be done if you change the SQL source, because then a new column set need to be re-created in the warehouse database, by the SQM++ Scheduler. Rebuild Index can be done to improve performance because indexes can be defragmented.

Use of Indexes optimize the performance for your data warehouse reporting.

To add indexes for the Target table click “Show Fields from Source” (see figure below). Then the fields appear in the list and you can click the blue < button to add as Index. You can define it as Unique, Clustered or both.

Remember to Save the Task (right lower corner of the dialog).

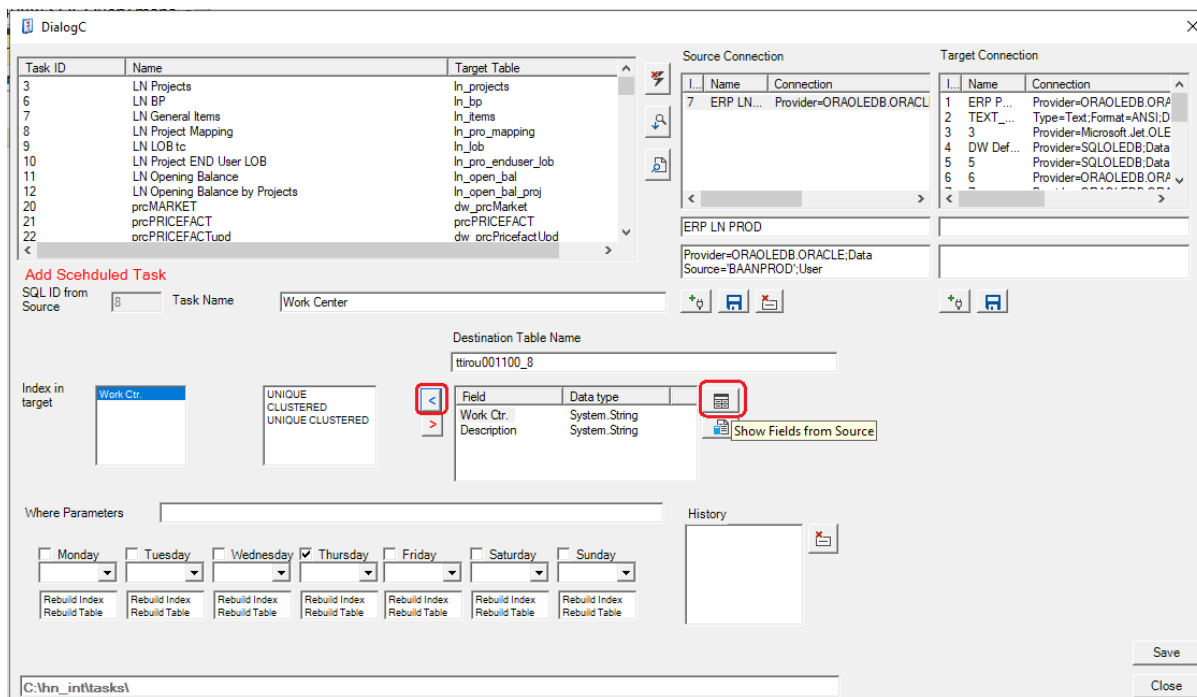


Figure: Show Fields From Source

Change the SQL for your task.

If you need to change the SQL for your task, find the task and click the “Show SQL Statement” button.

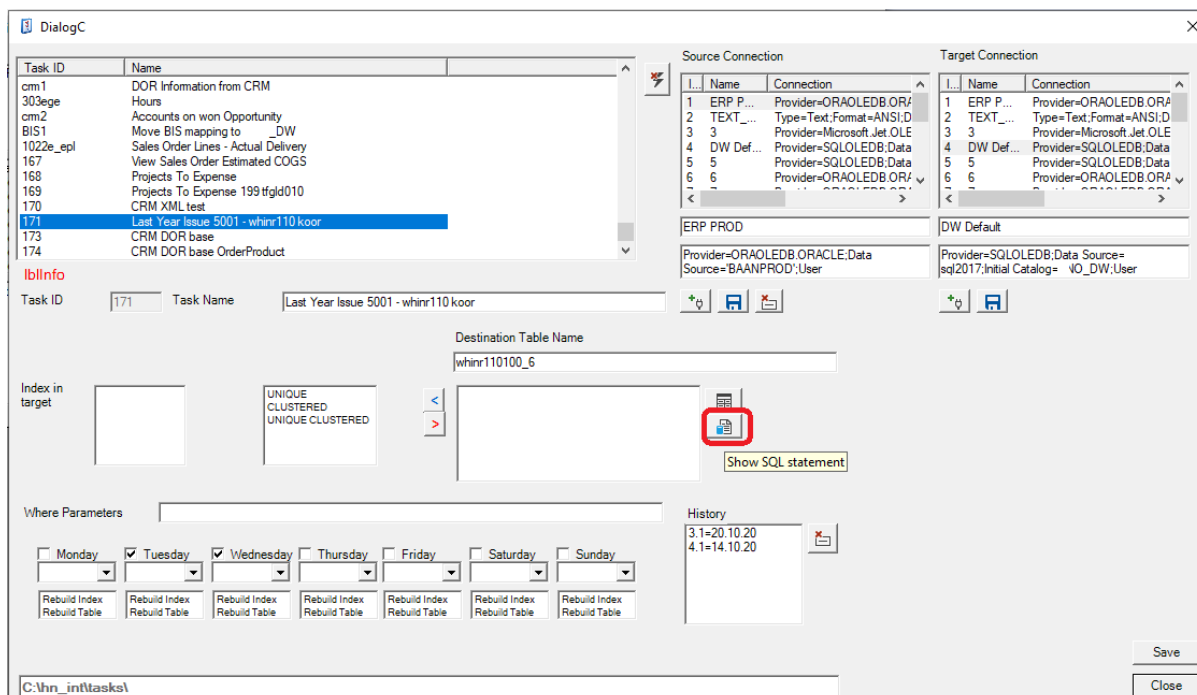


Figure: Show the SQL behind the task

Then an instant of the main form appears with the SQL in the SQL editor. But it is not linked to the original SQL statement it was created from.

There is a separate button to Update the SQL for the task. The other Update button (red in the figure below), is used for updating SQLs in the list. This makes it flexible. So you can both decide to update the Task SQL by typing, or by selecting a saved SQL in the list and Update the Task SQL from this.

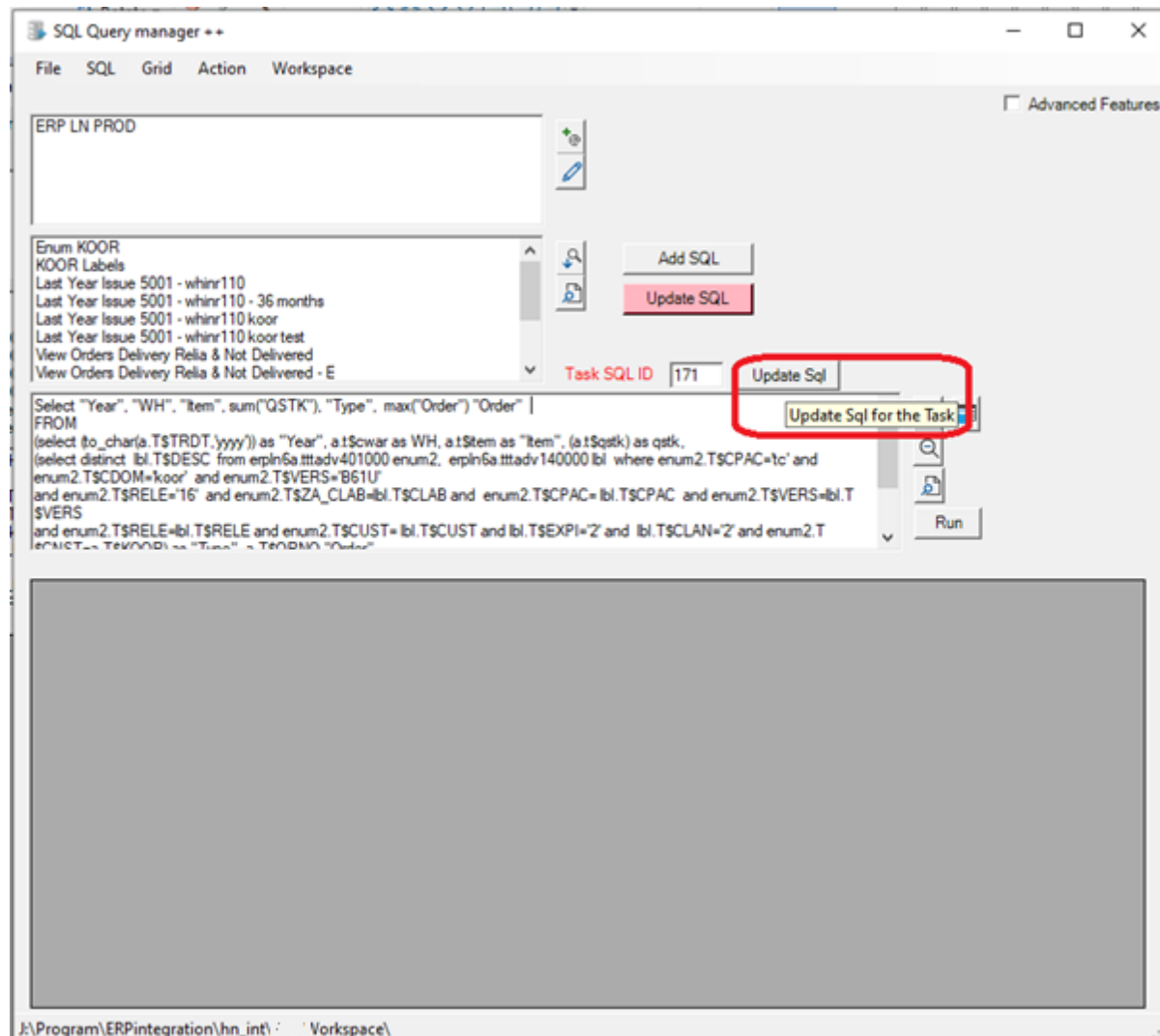
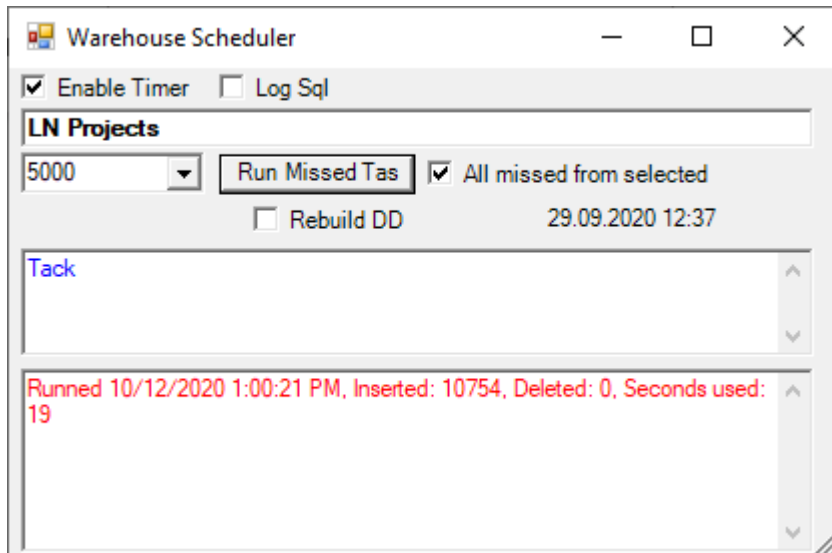


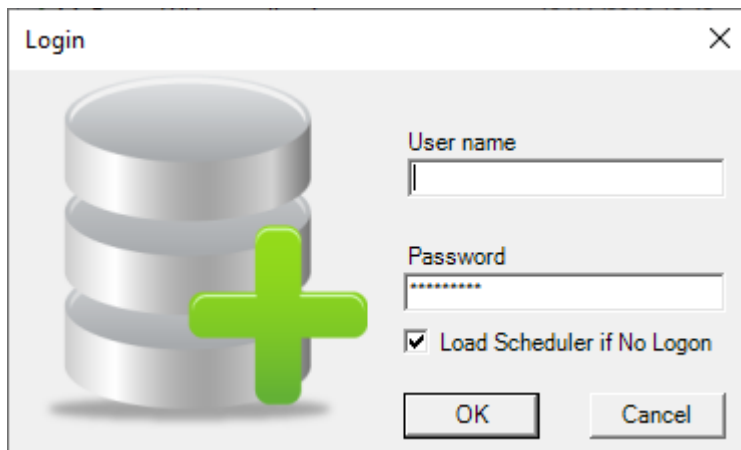
Figure: Update Task SQL statement

Warehouse Scheduler

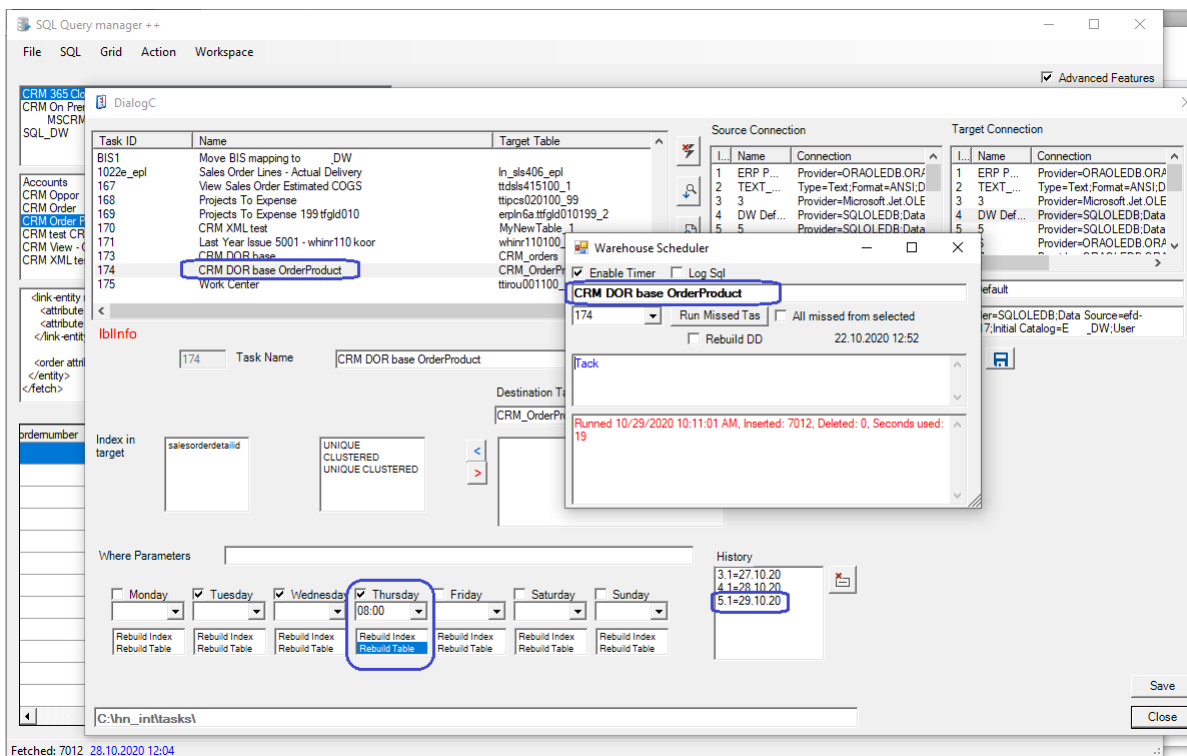
This is the tool that actually are executing the tasks and updates the data warehouse:



To start this tool, just tick the box “Load Scheduler if No Logon”



Wait 30 seconds and it will start. It remember the last used check value, so you can put the SQL Query Manager ++ shortcut in your Startup folder, to make it launch when the computer (being the Scheduler computer) starts up.



SQM ++ also works with SQL server in Azure:

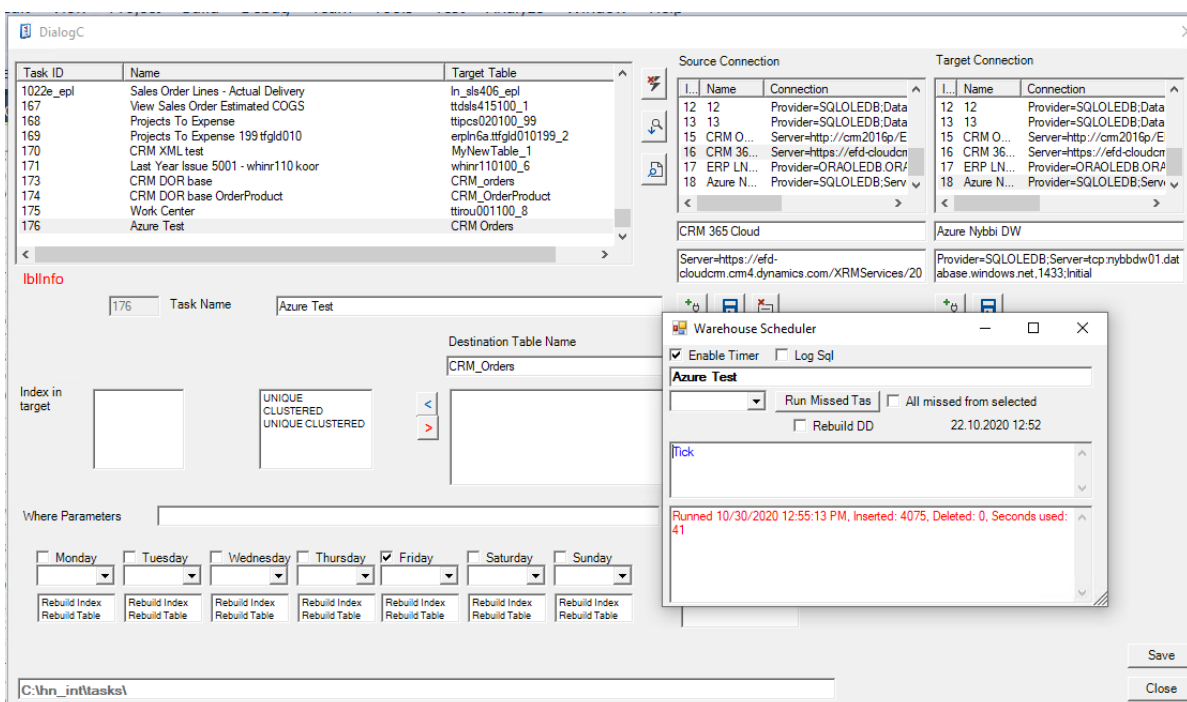


Figure: Azure sample

Logging the result

You can check the result of the Tasks by clicking the “Open Log” button:

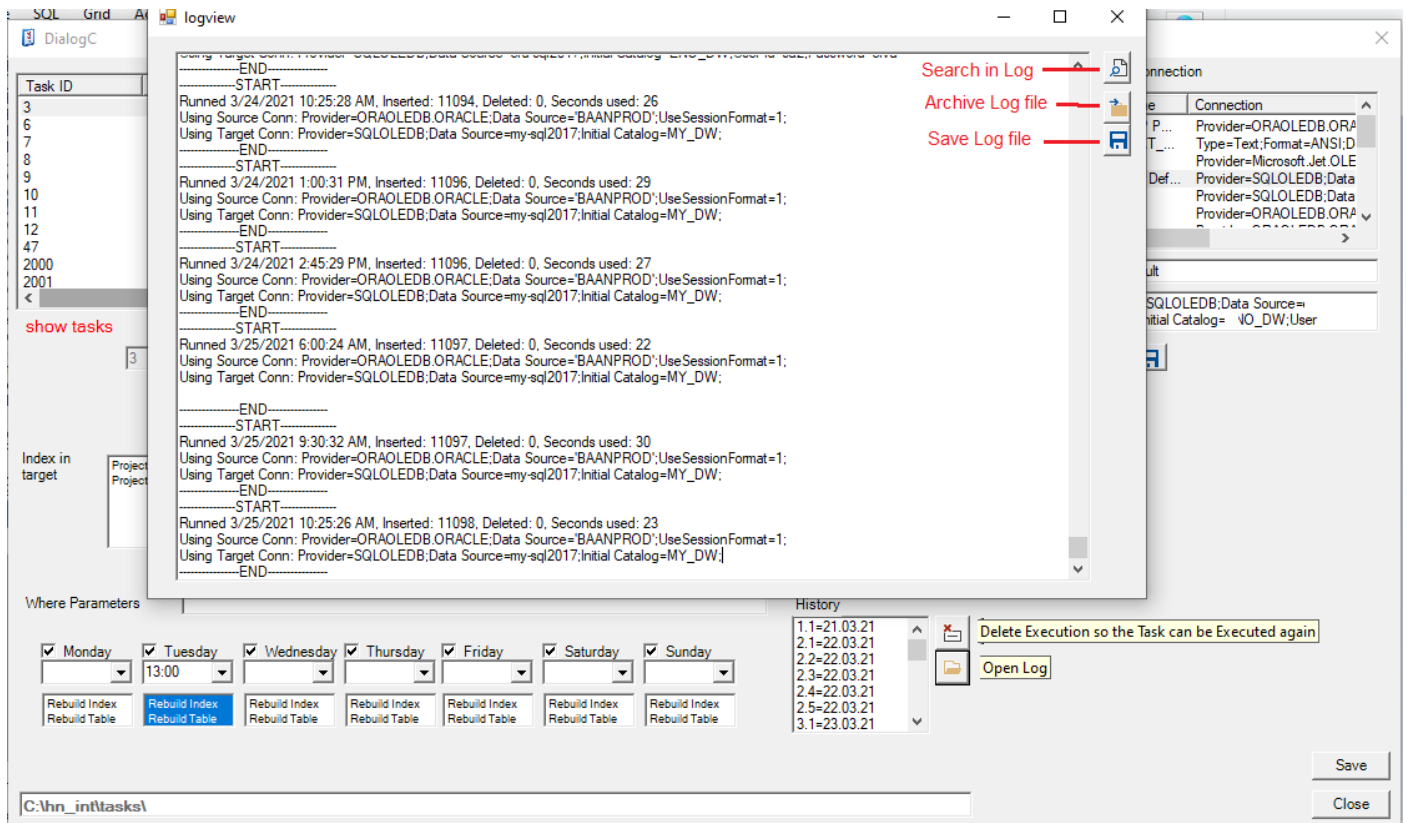


Figure: Task Log

If you want to re-run a task at current day, you can remove the task sequence for today's date, from the History list.