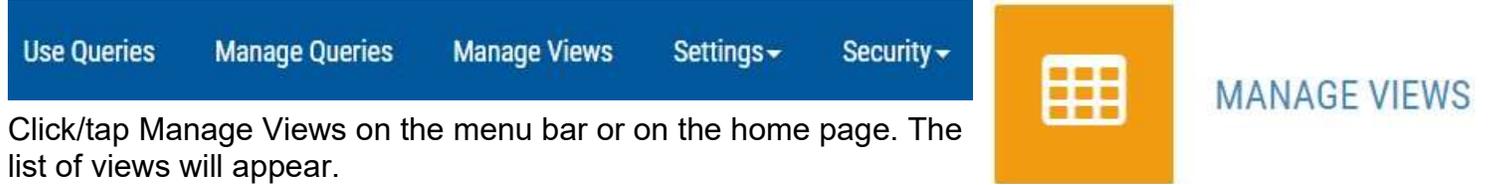


- List Views 1
- Create View – Basic Information 2
- Table List..... 3
- Column List 5
- Condition List..... 8
- Deploy the View 9

List Views



Click/tap Manage Views on the menu bar or on the home page. The list of views will appear.

You can: Edit, Clone, Change Status

Create or Import new view

Status left to right: In process, All saved, Generated, Active, Inactive, Void. Click off; click again on.

	View Name ^	View Type ^	Status	Description ^
	About Ewarenow Quick Query Demo	Imported View	Active	Gives a brief description of the Tourist Trap Di...
	Demo All Items	Imported View	Active	Lists all items (products) in the database, incl...
	Demo All Items Purchase Order History	Imported View	Active	Lists all items with purchase orders and provi...
	Demo All Items Sales Order History	Imported View	Active	Lists all items with sales orders and provides ...
	Demo All Organizations And Their Contacts	Imported View	Active	Lists all organizations and their contacts: sup...
	Demo All Purchase Orders	Imported View	Active	Lists all purchase orders we issued to supplier...
	Demo All Sales Orders	Imported View	Active	Lists all items with purchase orders and provi...
	Demo All Stock History	Imported View	Active	List of all stock (inventory) related transaction...
	Demo View for User Guidance	Created View	In process	View that can be used for demonstrating EQQ ...

You may want to click/tap to show more rows at a time
Or click/tap through the numbered blocks to find the view you want.
And the best choice may be to use dynamic search like this

	View Name ^	View Type ^	Status	Description ^
	Demo View for User Guidance	Created View	In process	View that can be used for demonstrating EQQ ...

Create View – Basic Information

A Structured Query Language (SQL) database consists of a set of tables with rows and columns of data and links relating the tables to one another. (and a lot of other details).

We look at a business document, a purchase order for example, as a single, complete, self-contained thing. But the document's content may be stored in several tables in an SQL database.

An SQL View is a combination 'on electronic paper' of two or more tables. It behaves like a very extensive table with hierarchical data flattened out. Its content only exists when it is being executed; and does not actually exist on a data storage device.

When setting out to create a View, it's advisable to think carefully about its purpose. That will help guide how you construct it.

This user wants to tie everything together from a customer to its sales orders, the items ordered from Tourist Trap Distribution and onward to the suppliers from which they were ordered and received.

Enter a View Name, View Description, and Optional View Tip. Then click/tap Save to be able to proceed to the other steps in the progress bar.

The screenshot shows the 'Edit View' interface with a progress bar at the top. The progress bar has five steps: Basic Information, Table List, Column List, Condition List, and Test View. The 'Basic Information' step is currently active, indicated by a red dot and a red line. Below the progress bar, the 'Basic Information' section contains several input fields and buttons. The 'View Name' field contains 'Demo View for User Guidance'. The 'View Description' field contains 'Customers to sales orders to items to suppliers'. The 'View Tip' field contains 'View that can be used for demonstrating EQQ and SQL features'. The 'Technical Name' field contains 'vQQ_Demo_View_for_User_Guidance_EQQDemo'. The 'View ID' field contains '33'. The 'View Status' field contains 'All saved'. A 'Save' button is located on the right side of the form. Green arrows point from the text above to the 'View Name', 'View Description', 'View Tip', and 'Save' fields.

Field	Value
View Name	Demo View for User Guidance
View Description	Customers to sales orders to items to suppliers
View Tip	View that can be used for demonstrating EQQ and SQL features
Technical Name	vQQ_Demo_View_for_User_Guidance_EQQDemo
View ID	33
View Status	All saved

Table List

The EQQDemo database contains 15 data tables, tEntity through tTypeTransaction listed below. They support the hypothetical Tourist Trap Distribution company's business. Each table has a Primary Key (PK). Many of the tables have Foreign Keys (FKs) linking them to other tables' PKs.

First click/tap C to choose the Controlling table. Then click/tap D to choose all Dependent tables. They will each be linked to one above it by an FK → PK relationship. Then click/tap Hide.

Edit View

Undo Recent
Close Page
Save All
Generate View
Test View
Activate View
Inactivate View
Void View
Clone View

<
Basic Information
Table List
Column List
Condition List
Test'
>

Table List

View Status: In process Save

Source Tables
Hide
Controlling Table

	Table ⇅	Type	
C	D	tAbout	Table
C	D	tDateMgmt	Table
C	D	tEntity	Table
C	D	tItem	Table
C	D	tItemTransaction	Table
C	D	tPurchaseOrderHeader	Table
C	D	tPurchaseOrderItem	Table
C	D	tRelationship	Table
C	D	tSalesOrderHeader	Table
C	D	tSalesOrderItem	Table
C	D	tStatusEntity	Table
C	D	tStatusItem	Table
C	D	tStatusItemTransaction	Table
C	D	tStatusPurchaseOrder	Table
C	D	tStatusSalesOrder	Table
C	D	tTypeItem	Table
C	D	tTypeTransaction	Table

Dependent Tables Drag & Drop

Prefix	Table	Alias	On Clause

Here are the results of selecting tables for this view. The Source Tables are hidden, but can be viewed again if the user clicks/taps the Show Source Tables button

The controlling table is tEntity with C as its alias

Dependent table 1 is tRelationship with the alias CR. It simply links an organization (Customer) to its contact(s): dependent table 2 which is tEntity with the alias CC.

Dependent tables 3 and 4 are the sales orders placed by the customer with aliases SO and SOI

Dependent table 5 is the Tourist Trap Distribution's list of all tItem records with the alias Item.

Dependent tables 6 is a lookup table tTypeItem as TI

Dependent table 7 is tEntity again, this time as S

Dependent tables 8 and 9 are tRelationship as SR and tEntity as SC to obtain suppliers' contacts.

Edit View

Undo Recent
Close Page
Save All
Generate View
Test View
Activate View
Inactivate View
Void View
Clone View

<

Basic Information
Table List
Column List
Condition List
Test View

>

Table List

View Status: In process Show Source Tables Save

Controlling Table

	Prefix	Table	Alias
From		tEntity	C

Dependent Tables Drag & Drop

		Prefix	Table	Alias	On Clause
1		Left Outer Join	tRelationship	CR	[C].[EntityId]=[CR].[OrgId]
2		Left Outer Join	tEntity	CC	[CR].[ContactId]=[CC].[EntityId]
3		Left Outer Join	tSalesOrderHeader	SO	[C].[EntityId]=[SO].[ShipToId]
4		Left Outer Join	tSalesOrderItem	SOI	[SO].[SalesOrderHeaderId]=[SOI].[SalesOrderHeaderId]
5		Left Outer Join	tItem	Item	[SOI].[OurItemId]=[Item].[OurItemId]
6		Left Outer Join	tTypeItem	TI	[Item].[TypeItemId]=[TI].[TypeItemId]
7		Left Outer Join	tEntity	S	[Item].[SupplierId]=[S].[EntityId]
8		Left Outer Join	tRelationship	SR	[S].[EntityId]=[SR].[OrgId]
9		Left Outer Join	tEntity	SC	[SR].[ContactId]=[SC].[EntityId]

10 25 50 100

Back To Manage
Previous Next

Click/tap the link to dependent table 3 to see exactly what ties this pair of tables together for purposes of this view. When the database contains well-defined and unique FK to PK constraints, EQQ can use them as it did here. In other cases, the user creating the view needs to help.

Left Table	Left Column	Left Expression	Right Table	Right Column	Right Expression
tEntity AS C	[EntityId]	[C].[EntityId]	tSalesOrderHe	[ShipToId]	[SO].[ShipToId]

Column List

In this view, the user is using 6 tables, some multiple times. So, there are 10 sets of source columns that can be used in the column list.

Edit View

Undo Recent Close Page Save All Generate View Test View Activate View Inactivate View Void View Clone View

Basic Information Table List **Column List** Condition List Test View

Column List

View Status: All saved New Column Save

Source Columns Hide Columns In View Drag & Drop

	Table	
+	tEntity	as C
+	tRelationship	as CR
+	tEntity	as CC
+	tSalesOrderHeader	as SO
+	tSalesOrderItem	as SOI
+	tItem	as Item
+	tTypeItem	as TI
+	tEntity	as S
+	tRelationship	as SR
+	tEntity	as SC

10 25 50 100

Back To Manage

These are the columns the user chose from the 10 sets of source columns

				Alias	Column	New Name	Settings
A				C	OrgOrLastName	Customer	<input type="checkbox"/>
B				C	Address1	CAddress1	<input type="checkbox"/>
C				C	Address2	Caddress2	<input type="checkbox"/>
D				C	City	CCity	<input type="checkbox"/>
E				C	StateProv	CStateProv	<input type="checkbox"/>
F				C	PostalCode	CPostalCode	<input type="checkbox"/>
G				C	Country	CCountry	<input type="checkbox"/>
H				C	EntityId	CustomerPK	<input type="checkbox"/>
I				C	StatusId	CustomerStatusID	<input type="checkbox"/>
J				C	IsCustomer		<input type="checkbox"/>
K				CR	RoleOrTitle	CustomerContactRole	<input type="checkbox"/>
L				CC	OrgOrLastName	CCLastName	<input type="checkbox"/>
M				CC	FirstName	CCFirstName	<input type="checkbox"/>
N				CC	EntityId	ContactPK	<input type="checkbox"/>
O				CC	StatusId	ContactStatusID	<input type="checkbox"/>
P				CC	IsContact	IsCustomerContact	<input type="checkbox"/>
Q				SO	SalesOrderHeaderId		<input type="checkbox"/>
R				SO	DateOrdered		<input type="checkbox"/>
S				SO	DateDeliveryRequested		<input type="checkbox"/>
T				SO	StatusId	SOStatusID	<input type="checkbox"/>
U				SOI	SalesOrderItemId		<input type="checkbox"/>
V				SOI	StatusId	SOIStatusID	<input type="checkbox"/>
W				SOI	TotalQtyShipped		<input type="checkbox"/>
X				SOI	UnitPrice		<input type="checkbox"/>
Y				SOI	DateLastShipped		<input type="checkbox"/>
Z				Item	OurItemId		<input type="checkbox"/>
AA				TI	TypeItem		<input type="checkbox"/>
AB				Item	OurItemName		<input type="checkbox"/>
AC				Item	ItemDescription		<input type="checkbox"/>
AD				Item	PackQty		<input type="checkbox"/>
AE				S	EntityId	SupplierPK	<input type="checkbox"/>
AF				S	OrgOrLastName	Supplier	<input type="checkbox"/>
AG				S	IsSupplier		<input type="checkbox"/>
AH				S	Address1	SAddress1	<input type="checkbox"/>
AI				S	Address2	Saddress2	<input type="checkbox"/>
AJ				S	City	SCity	<input type="checkbox"/>
AK				S	StateProv	SStateProv	<input type="checkbox"/>

AL				S	PostalCode	SPostalCode	<input checked="" type="checkbox"/>
AM				S	Country	SCountry	<input checked="" type="checkbox"/>
AN				SR	RoleOrTitle	SupplieContactRole	<input checked="" type="checkbox"/>
AO				SC	OrgOrLastName	SCLastName	<input checked="" type="checkbox"/>
AP				SC	FirstName	SCFirstName	<input checked="" type="checkbox"/>
AQ				SC	IsContact	IsSupplierContact	<input checked="" type="checkbox"/>

10 25 50 100

[← Back To Manage](#)

[Previous](#)

[Next](#)

Condition List

Conditions impose limits on which rows of data will be selected.

The user clicked/tapped  several columns to create this list of conditions

Then the user clicked the edit button on each one to specify the exact condition to be met.

Edit View

Undo Recent Close Page Save All Generate View Test View Activate View Inactivate View Void View Clone View

Basic Information Table List Column List Condition List Test View

Condition List

View Status: All saved Save

			Column	Type	From Value	To Value
1			[C].[IsCustomer]	Is equal to	'True'	
2			[CC].[IsContact]	Is equal to	'True'	
3			[S].[IsSupplier]	Is equal to	'True'	
4			[SC].[IsContact]	Is equal to	'True'	
5			[CR].[RoleOrTitle]	Is equal to	'CFO'	
6			[SR].[RoleOrTitle]	Is equal to	'CFO'	

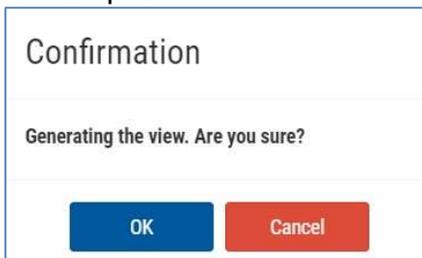
10 25 50 100

Deploy the View

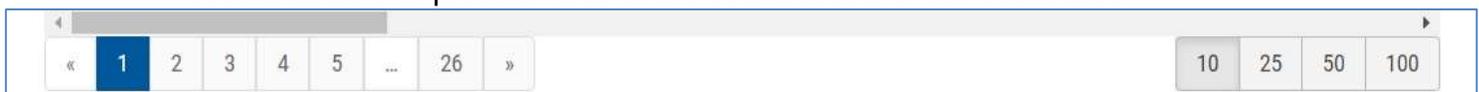
When you have completed your work on Table List, Column List, and Condition List, you can check out your work and, if satisfactory activate the view so it can be used in Manage Queries.



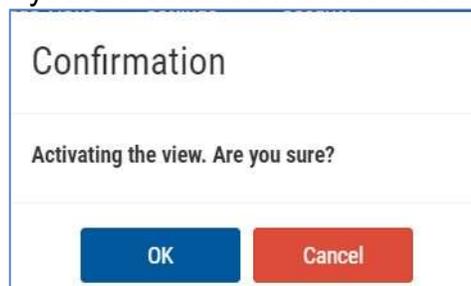
1. Click/tap Save All and receive a green success message: “All data has been saved”
 - If the Save All fails, you need to correct the problems. Common ones are errors in Table List On Clauses and missing Type and Value settings in Condition List.
2. Click/tap Generate View and receive a confirmation message



3. Click/tap OK and receive a green success message: “Generate View succeeded”
 - If the Generate View fails, you need to find and correct the errors and repeat this process.
4. Click/Tap Test View. If no SQL errors are detected, you will see the results displayed on the page
 - The SQL code you have defined with the work above has now been executed. There may be errors. Most likely they will be caused by problems with Table List On Clauses or Condition List Type and Value settings.
5. Review the results to see if the data is what you expected.
6. Scroll to the bottom and look at the number of records generated. Does it seem reasonable?
 - In this example, it's between 250 and 260 records (10 per block with the highest block number being 26). It is a small database with few customers and few items. With data for only about two years. So it's probably OK.
 - In earlier versions of the view it was generating over 3000 records and that was not reasonable. The problem was in the construction of the Table List.



7. If you are satisfied with the results click/tap Activate View and receive the green message:



8. Click/tap OK and receive the green message: Update Status succeeded.
9. Click/Tap Close Page to return to List Views