



Praqtize 2017

Build an On-Demand Qlik Solution

Introduction

The on-demand app generation approach expands the potential use cases for Business Discovery, enabling business users to conduct associative analysis on larger data sources. It provides a shopping list experience to allow users to first select data they are interested in discovering insights on which interactively an analysis app with full Qlik in-memory capabilities.

This workshop is designed to deliver an introduction into the steps required to build an on-demand solution in Qlik Sense using the built-in navigation options. The source data itself is a cut down version of sales and inventory data stored as qvd files, the process will be the same for a real-world on-demand deployment.

Scenario

Company X has a series of requirements they wish to deliver with Qlik Sense, they want to allow a section of users to access all their data without consuming all the data into a Qlik Sense app. The following metrics are required to help understand the product sales mix and relevant stock level details of the products Company X sells:

- Product sales over time.
- Product analysis with sales amounts and order quantities
- Customer demographic analysis by occupation and gender
- A detailed list of sales orders with associated attributes
- A detailed list of stock movement data by product and product sub categories

Goals and Content

The workshop will cover the script required to create the selection application which will be an aggregated view of the total dataset with associated dimensions as well as the detail applications which the on-demand process will generate user specific slices of.

Within the ODAG_Course.zip file you will find the following assets:

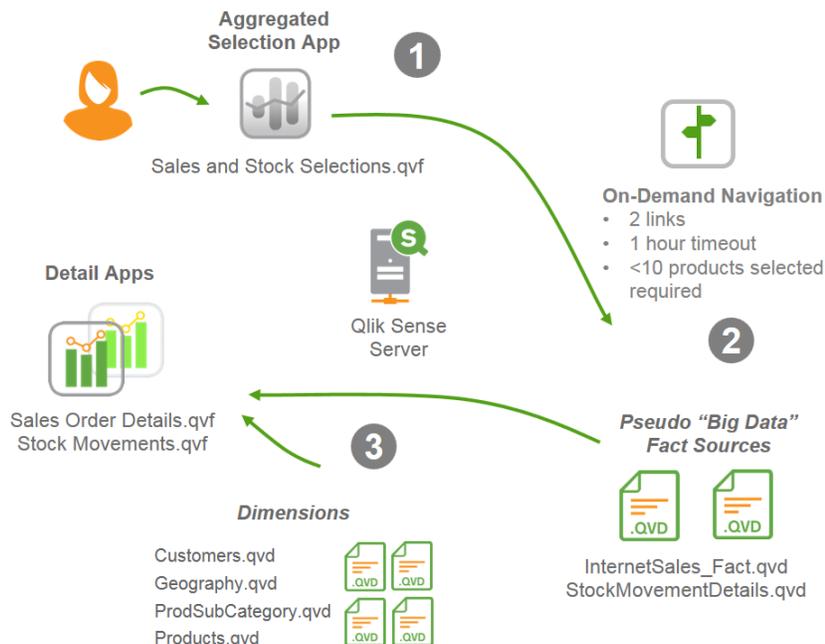
- Source data qvd files
 - InternetSales_Fact.qvd
 - StockMovementDetails.qvd
 - Customers.qvd
 - Geography.qvd
 - ProdSubCategory.qvd
 - Products.qvd
- Source Qlik Sense application which will be used as the template for the detail app on-demand script
 - DetailsTemplate.qvf

Prerequisites

Qlik Sense Server 2017 June release or above (note the on-demand process can be used with older versions using the extension object shipped with Qlik Sense Server 3.0).

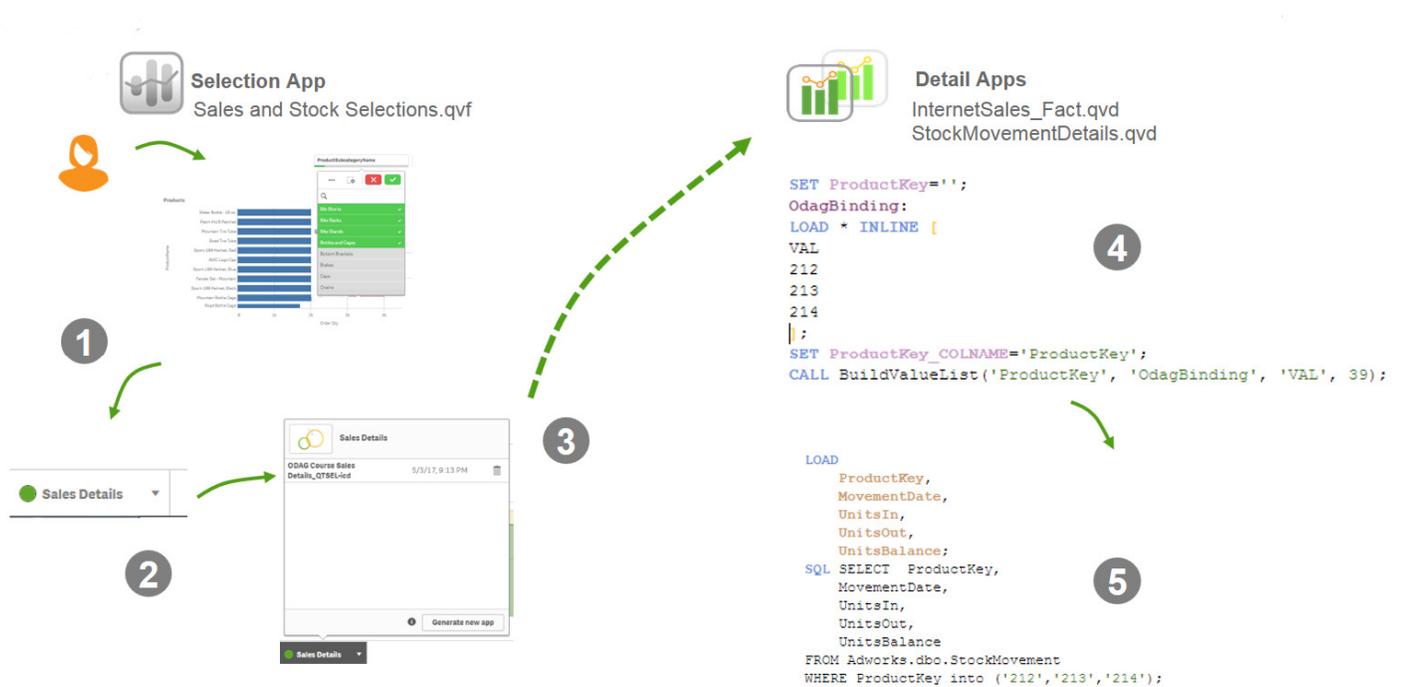
Outcome

The result of the workshop will be three applications, two of which will act as the detail apps which form the basis of the on-demand apps generated from the links created. Below are diagrams which show the process flow:



1. User logs into the Selection App which is populated with dimensional data qvd files and an aggregated view of the InternetSales_Fact qvd. Selections are made to hit the governed limit based on distinct ProductNames.
2. The governed limit is reached and the navigation button on the toolbar becomes active with a green indicator. The user can then choose to generate a new sales detail or Stock movement application by dynamically reloading data from the source qvd files or choose an existing app.
3. The ProductKey associated values are passed into either detail app and form the basis of the WHERE clause populated via an INLINE Load and the app published to a stream.

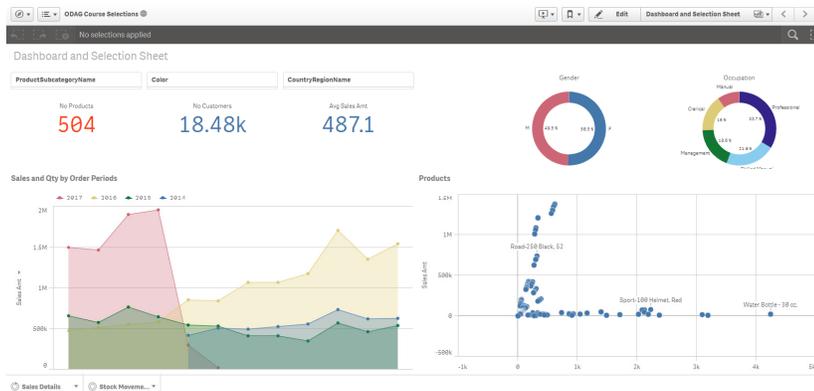
A detail of the script process below:



1. Data items are selected from charts/filter boxes in the aggregated selection app in memory which is populated with dimensional data qvd files and an aggregated view of the InternetSales_Fact qvd.
2. Once a governed limit is reached, based on a data limit - count(distinct ProductName) the on-demand button becomes active.
3. The selections have generated a filtered list of associated ProductKey's and these are dynamically passed into one of the detail apps which contain charts and the reload is initiated.
4. The detail app contains SET statements which indicate which fields are used for filtering (ProductKey) and creates an INLINE LOAD statement dynamically with these values.
5. The detail app contains script to convert these into a WHERE clause variable and the app is reloaded with the where clause and published to a stream with an optional timeout.

Sample chart outcomes are shown below:

Sales and Inventory Selections:



Sales Details:

SalesOrderNumber	SalesOrderLineNumber	OrderDate	ProductName	Sales Amt	Order Qty	Tax Amt
Totals				€29,358,672.22	60398	€2,348,694.23
S051179	4	04/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051488	5	23/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051529	3	25/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051598	4	27/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051882	3	04/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051983	3	04/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S051970	4	06/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052610	5	06/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052831	2	07/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052895	2	08/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052800	4	12/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052357	2	13/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S053372	2	14/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052380	3	14/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72
S052417	1	15/09/2016 00:00:00	All-Purpose Bike Stand	€159.00	1	€12.72

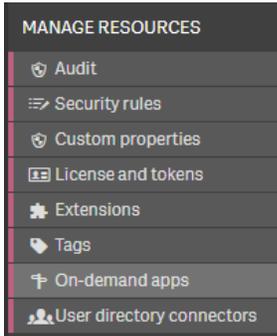
Stock Movement Details:

MovementDate	ProductKey	ProductName	Balances	Outward Movements	Inward Movements
Totals			333,869,486	274,276	276,969
13/02/2014	1	Adjustable Race	875	0	0
13/02/2014	2	Bearing Ret	475	0	0
13/02/2014	3	BB Ball Bearing	700	0	0
13/02/2014	4	Headset Ball Bearings	700	0	0
13/02/2014	5	Blade	700	0	0
13/02/2014	6	LL Crankarm	438	0	0
13/02/2014	7	HL Crankarm	238	0	0
13/02/2014	8	HL Crankarm	438	0	0
13/02/2014	9	Chaining Bolts	875	0	0
13/02/2014	10	Chaining Nut	875	0	0
13/02/2014	11	Chaining	875	0	0
13/02/2014	12	Crown Race	875	0	0
13/02/2014	13	Chain Stays	875	0	0
13/02/2014	14	Decal 1	875	0	0

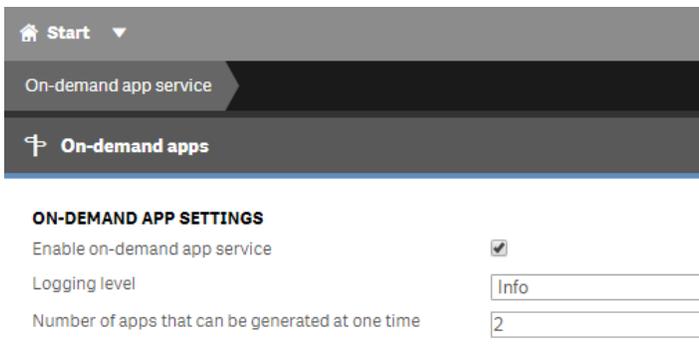
Exercise 1

In this first short exercise, we will build enable the on-demand functionality by activating it in the QMC.

1. Open the Qlik Sense Management console.
2. Navigate to the On-demand apps section under MANAGE RESOURCES



3. Enable the on-demand app service



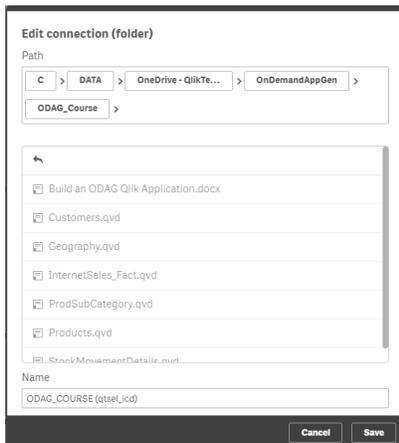
We have now completed the first short exercise.

Exercise 2

In this exercise, we will build a simple aggregated Qlik Sense application based on the dimensional qvd files and an aggregated view of the **InternetSalesFact** qvd file and produce a few simple visuals. This will also act as the initial selection application in the on-demand process.

Create a New Qlik Sense Application

1. Unzip the **ODAG_Course.zip** file to a location we will use to create a folder connection later.
2. Open the Qlik Sense Hub and click *Create new app* and provide a name such as Sales and Inventory Selections.
3. Go into the *Data Load Editor* and create a new folder connection to the location where we unzipped the file from step 1, for example:



4. Create the load script for the dimensional qvd files (**StockMovementDetails.qvd**, **Customers.qvd**, **Geography.qvd**, **ProdSubCategory.qvd**, **Products.qvd**).

Create the summary load script

1. To create a summary view of the fact data we will use the **InternetSalesFact.qvd** file and only use the following fields:
 - ProductKey,
 - CustomerKey,
 - OrderQuantity,
 - DiscountAmount,
 - SalesAmount,
 - TaxAmt,
 - Freight,

- OrderDate
- ShipDate

The measure columns will be aggregated using a sum() function and the date fields will be transformed to only go down to Year, Month granularity.

2. Create the aggregated load script which should look something like this:

```
1 InternetSales_Fact_Summary:
2 LOAD
3     ProductKey,
4     CustomerKey,
5     sum(OrderQuantity) as OrderQuantity,
6     sum(DiscountAmount) as DiscountAmount,
7     sum(SalesAmount) as SalesAmount,
8     sum(TaxAmt) as TaxAmt,
9     sum(Freight) as Freight,
10    Year(OrderDate) as OrderYear,
11    Month(OrderDate) as OrderMonth,
12    Year(ShipDate) as ShipYear,
13    Month(ShipDate) as ShipMonth
14 FROM [lib://ODAG_COURSE (qtse1_icd)/InternetSales_Fact.qvd] (qvd)
15 GROUP By
16     ProductKey,
17     CustomerKey,
18     Year(OrderDate),
19     Month(OrderDate),
20     Year(ShipDate),
21     Month(ShipDate);
```

3. Load the data into the application.

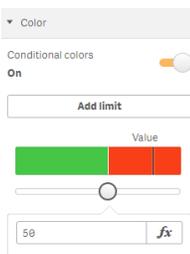
We have now completed the second exercise.

Exercise 3

In this exercise, we will build some charts on the aggregated data in the app we created in exercise 2, this will help answer a few of the business questions for Customer X and provide the selection app for the on-demand process.

1. Create a new sheet and add three filter boxes for ProductSubCategoryName, Color and CountryRegionName
2. Create three measures in the Master Items
 - # Products - count(distinct ProductName)
 - # Customers - count(distinct EmailAddress)
 - Avg Sales Amount - avg(SalesAmount)

As the # Products measure will be used in the on-demand expression later add conditional colors to further indicate to the user that they have reached the required number of products to allow drill to details:



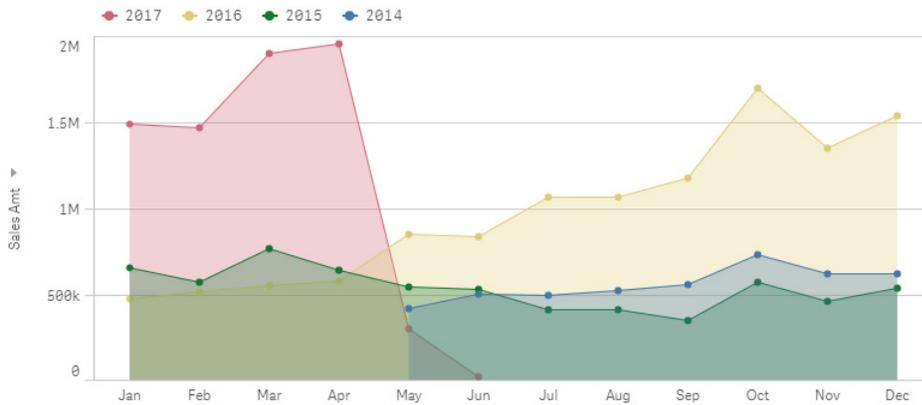
The KPI's should look as follows:



3. Create two new measures:
 - Sales Amt - sum(SalesAmount)
 - Order Qty - sum(OrderQuantity)

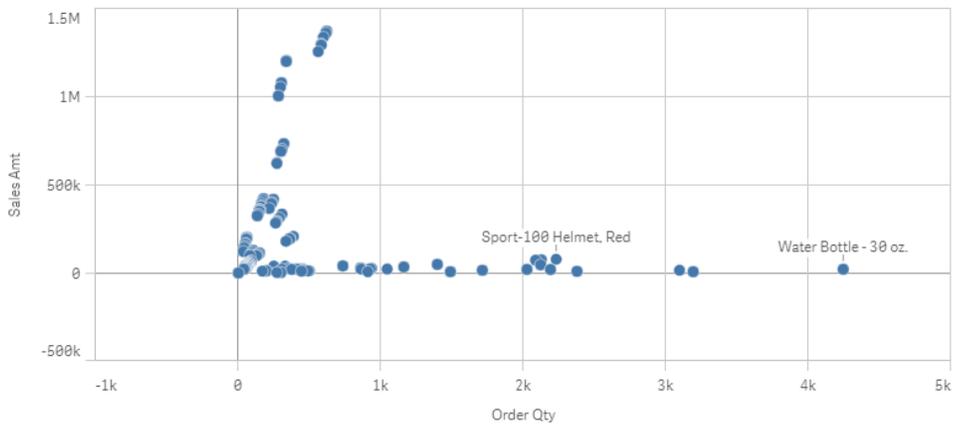
Create an area line chart using Order Year, Order Month as dimensions and Sales Amt as measure with Order Qty as the alternative measure:

Sales and Qty by Order Periods

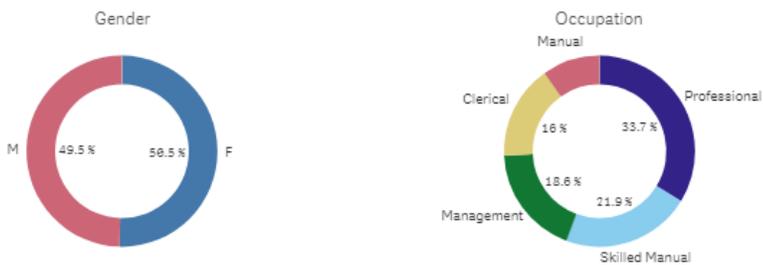


4. Create a scatter plot using ProductName as the dimension and Sales Amt/Order Qty as measures:

Products



5. Create two pie charts one with Gender/Sales Amt the other with Occupation/Sales Amt:



We have now completed the third exercise.

Exercise 4

In this exercise, we will build the on-demand details app to allow users at Customer X to access the lowest level of detail in their dataset namely the sales orders and lines. We will use the template app provided and copy the script to a new app and modify some variables/build some sample charts, finally we will create the on-demand link in the selection app.

1. Import the template app (**DetailsTemplate.qvf**) into your Work stream via the QMC.
2. Create a new application called **OrderDetails.qvf**.
3. Open the *Data Load Editor* and add a new section called Dynamic Data in the newly created **OrderDetails.qvf**.
4. Open the **DetailsTemplate** app navigate to the Dynamic Data section and copy then entire script in that section (Lines 1 – 114).
5. Paste previously copied script into the newly created **OrderDetails.qvf** Dynamic Data section.



The detail script contains several different sections to generate the where clause, in the example we have three subroutines which will not require modifying for the purposes of this exercise. For reference, they are explained below:

The first part of the process specifies the dimensional values which are to be passed dynamically from the selection app into the detail app via an INLINE LOAD:

```
64 SET YourField = '';
65 OdagBinding:
66 LOAD * INLINE [
67 VAL
68 $(odo_YourField){"quote": "'", "delimiter": ","}
69 ];
70 SET YourField_COLNAME='YourField';
71 CALL BuildValueList('YourField', 'OdagBinding', 'VAL', 39); // 39 is for single quote wrapping values
```

Options exist to specify how the data in the fields will be selected by prefixing the column name as follows:

- ods = Selected values
- odo = Associated values
- odso = Selected/associated values

Quoting and delimiters can also be specified (default being single quotes with comma delimited).

This will then call the INLINE Load subroutine.

The next subroutine dynamically constructs an `INLINE[]` Load statement which will contain all of the data items selected/associated from the selection app passed into the detail app:

```

29 // DO NOT ALTER THIS SUBROUTINE
30 SUB BuildValueList(VarName, TableName, ColName, QuoteChrNum)
31 IF $(QuoteChrNum) = 0 THEN
32 LET LOADEXPR = 'Concat($(ColName),' & chr(39) & ',' & chr(39) & ') AS CombinedData';
33 ELSE
34 LET CHREXPR = ' chr(' & '$(QuoteChrNum)' & ') ';
35 LET LOADEXPR = 'Concat( $(CHREXPR) & $(ColName) & $(CHREXPR)' & ',' & chr(39) & ',' & chr(39) & ') AS CombinedData';
36 ENDIF
37 _TempTable:
38 LOAD $(LOADEXPR) Resident $(TableName);
39 Let vNoOfRows = NoOfRows('_TempTable');
40 IF $(vNoOfRows) > 0 THEN
41 LET $(VarName) = Peek('CombinedData',0,'_TempTable');
42 ENDIF
43 drop table _TempTable;
44 drop table '$(TableName)';
45 END SUB;

```

The final subroutine in the process constructs the `$(WHERE_PART)` variable which populates the where clause executed on the fact table select statement. It is specifically designed to cater for multiple values in a QVD file where clause (which does not support IN Clauses) so the `mixmatch()` function is used:

```

1 // DO NOT ALTER THIS SUBROUTINE
2 SUB ExtendQVDWhere(Name, ValVarName)
3 LET T = Name & '_COLNAME';
4 LET ColName = $(T);
5 LET Values = $(ValVarName);
6 IF len(Values) > 0 THEN
7 IF len(WHERE_PART) > 0 THEN
8 LET WHERE_PART = '$(WHERE_PART) AND mixmatch([$ (ColName)],$(Values) )';
9 ELSE
10 LET WHERE_PART = ' WHERE mixmatch([$ (ColName)],$(Values) )';
11 ENDIF
12 ENDIF
13 END SUB;

```

An alternative subroutine (not called in this exercise) is present in the example to act as a reference showing the changes required if the data sources are SQL compliant databases:

```

15 // DO NOT ALTER THIS SUBROUTINE
16 SUB ExtendSQLWhere(Name, ValVarName)
17 LET T = Name & '_COLNAME';
18 LET ColName = $(T);
19 LET Values = $(ValVarName);
20 IF len(Values) > 0 THEN
21 IF len(WHERE_PART) > 0 THEN
22 LET WHERE_PART = '$(WHERE_PART) AND $(ColName) IN ( $(Values) )';
23 ELSE
24 LET WHERE_PART = ' WHERE $(ColName) IN ( $(Values) )';
25 ENDIF
26 ENDIF
27 END SUB;

```

The `$(WHERE_PART)` variable now constructs a where clause using the **IN** function.

Modifications to the detail app load script

1. Replace the “YourField” references (lines 64-71) with the first field we will use in the where clause generation: ProductKey and enable the values to be selected by association (odo):

```
64 SET ProductKey='';
65 OdagBinding:
66 LOAD * INLINE [
67 VAL
68 $(odo_ProductKey){"quote": "", "delimiter": ""}
69 ];
70 SET ProductKey_COLNAME='ProductKey';
71 CALL BuildValueList('ProductKey', 'OdagBinding', 'VAL', 39); // 39 is for single quote wrapping values
```

2. Create a new field reference which will also be used in the WHERE clause generation: CustomerKey and enable the values to be selected by association (odo):

```
64 SET ProductKey='';
65 OdagBinding:
66 LOAD * INLINE [
67 VAL
68 $(odo_ProductKey){"quote": "", "delimiter": ""}
69 ];
70 SET ProductKey_COLNAME='ProductKey';
71 CALL BuildValueList('ProductKey', 'OdagBinding', 'VAL', 39); // 39 is for single quote wrapping values
72
73 SET CustomerKey='';
74 OdagBinding:
75 LOAD * INLINE [
76 VAL
77 $(odo_CustomerKey){"quote": "", "delimiter": ""}
78 ];
79 SET CustomerKey_COLNAME='CustomerKey';
80 CALL BuildValueList('CustomerKey', 'OdagBinding', 'VAL', 39); // 39 is for single quote wrapping values
```

Next, we need to add the list of field names to the script section which invokes the WHERE clause creation subroutines:

```
FOR EACH fldname IN 'YourField'
  LET vallist = $(fldname);
  WHEN (IsNull(vallist)) LET vallist = '';
  IF len(vallist) > 0 THEN
    CALL ExtendQVDWhere('$(fldname)', 'vallist');
  ENDIF
NEXT fldname
```

3. Change the 'YourField' reference to ProductKey and add CustomerKey:

```
FOR EACH fldname IN 'ProductKey', 'CustomerKey'
  LET vallist = $(fldname);
  WHEN (IsNull(vallist)) LET vallist = '';
  IF len(vallist) > 0 THEN
    CALL ExtendQVDWhere('$(fldname)', 'vallist');
  ENDIF
NEXT fldname
```

Next, we need to add the folder connection details by populating the \$(FOLDER) and \$(FACT_QVD) variables:

```
LET FOLDER='';
LET FACT_QVD='[$(FOLDER) / ';
```

4. Add the folder connection details and the qvd file: **InternetSales_Fact.qvd**.

```
LET FOLDER='lib://ODAG_COURSE (qtset_icd)';
LET FACT_QVD='[$(FOLDER)/InternetSales_Fact.qvd] (qvd)';
```

5. Add the load script for the **InternetSales_Fact.qvd** file and change the FROM clause to use the \$(FACT_QVD) variable and add the \$(WHERE_PART) variable to the end of the script.

NOTE In a real-world deployment the fact table could contain billions of rows so for the purposes of construction of the detail app you will need to limit the amount of data in the detail app.

```
LOAD
  ProductKey,
  CustomerKey,
  SalesOrderNumber,
  SalesOrderLineNumber,
  OrderQuantity,
  DiscountAmount,
  SalesAmount,
  TaxAmt,
  Freight,
  OrderDate,
  ShipDate
FROM $(FACT_QVD)
$(WHERE PART);
```

6. The final stage of the process is to add the dimensional load statements for the following:

- Customers.qvd
- Products.qvd
- ProdSubCategory.qvd
- Geography.qvd

We also need to restrict these loads to those items which have corresponding keys in the fact table, an example of this for the customers below:

Customers:

LOAD

```
CustomerKey,  
GeographyKey,  
Title,  
FirstName,  
LastName,  
BirthDate,  
MaritalStatus,  
Suffix,  
Gender,  
EmailAddress,  
YearlyIncome,  
TotalChildren,  
Occupation
```

```
FROM [lib://ODAG_COURSE (qtsel_icd)/Customers.qvd]  
(qvd)  
Where exists(CustomerKey);
```

Create the detail app objects

The sales detail app now contains the relevant script, from here we need to build the detail report and filters:

1. Create a new sheet and add three filter boxes for ProductSubCategoryName, Color and CountryRegionName
2. Create five measures in the Master Items
 - # Products - count(distinct ProductName)
 - # Customers - count(distinct EmailAddress)
 - Sales Amount - sum(SalesAmount)
 - Order Qty - sum(OrderQuantity)
 - Tax Amt - sum(TaxAmt)
3. Add two KPI objects for
 - # Products - count(distinct ProductName)
 - # Customers - count(distinct EmailAddress)
4. Add a table chart containing SalesOrderNumber, SalesOrderLineNumber, OrderDate, ProductName, Sales Amt, Order Qty and Tax Amt

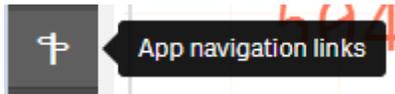
The final sheet should look as follows:

SalesOrderNumber	SalesOrderLineNumber	OrderDate	ProductName	Sales Amt	Order Qty	Tax Amt
Totals				£29,358,877.52	60708	£2,348,684.33
S051179	4	04/05/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051488	5	23/05/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051520	3	29/05/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051558	4	27/05/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051882	3	04/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051983	3	04/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S051970	4	06/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052010	5	06/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052032	2	07/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052065	2	08/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052309	4	12/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052357	2	13/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052372	2	14/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052380	3	14/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72
S052417	1	15/06/2016 00:00:00	All-Purpose Bike Stand	£159.00	1	£12.72

Creating the On-demand links

The final part of the process will create the on-demand link between the selection app and the detail app using the built-in navigation options.

1. Open the selection app created in exercise 2 and navigate to the edit sheet view.
2. Open the *App navigation links* panel



and click *create new*.

3. The link should be setup to point to the detail app created previously. The expression to allow the link to be activated should be based on no more than 50 distinct ProductNames being selected and should open on the first sheet with a timeout of one hour and not published to a steam, for example:

Edit on-demand app navigation link

Name:

Template app:

Expression:
 fx

Maximum row count:

Generated apps

Maximum number:

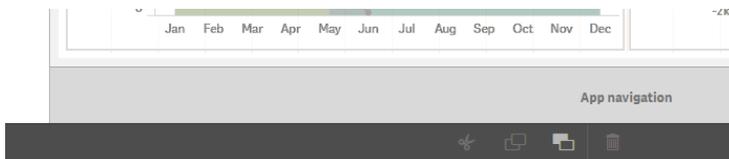
Retention time:

Default view when opened:

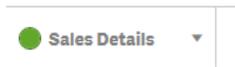
Publish to:

And save the link

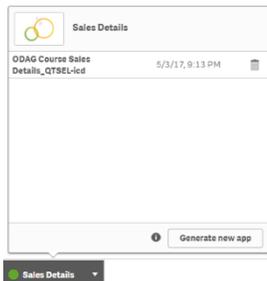
4. Drag and drop the newly created link to the App navigation area at the bottom of the sheet:



5. Click done and test the link by selecting various items on the screen until the on-demand icon turns completely green:



6. Finally open the on-demand dialog and click *Generate new app* and wait for the process to run allowing you to click the newly generated link:



We have now completed the fourth exercise.

Bonus Exercise

Customer X has a further requirement to allow users to drill from the same selection app we have developed into a lower level of detail containing Stock Movements to allow analysis of the following:

- No Products
- Inward Movements
- Outward Movements
- Detail list of MovementDate, ProductKey, ProductName, Balances and inward/outward movements in a table
- Filter by ProductSubCategoryName, ProductName and Color.

To facilitate this **StockMovementDetails.qvd** can be used. The additional requirements are to pass through the ProductKey from the selection to detail app with a limit of no more than ten distinct ProductNames.