Qlik Notes:

Joins:

1. Joins: We have 4 Types of Joins:
2. Inner Join, Outer Join, Left Join, Right Join.
3. Inner Joins- It fetches common records between 2 Tables.
4. Outer Joins-It Fetches whole record from both the Tables.
5. Left Joins- if we have 2 Tables. Table A & Table B. IF we do left Join on (Table A) it will take the whole record of Table A and the common record from Table B will be taken.
6. Right Join- if we have 2 Tables. Table A & Table B. IF we do Right Join on (Table A) it will take whole record of Table B and common record from Table A will be taken.
7. Link : https://mindmajix.com/qlikview-intervalmatch

Keeps:

A. Keep: We have 3 Types of Keep:

1. Inner Keep- It Fetches common record between 2 tables and keep both tables separately.
2. Left Keep: - if we have 2 Tables. Table A & Table B. IF we do left Join on (Table A) it will take the whole record of Table A and the common record from Table B will be taken and both the tables will be shown separately.
3. Right Keep: if we have 2 Tables. Table A & Table B. IF we do Right Keep on (Table A) it will take whole record of Table B and common record from Table A will be taken and both the tables will be shown separately.
4. https://mindmajix.com/qlikview-intervalmatch

Note: Frequency increases while doing Joins, and single frequency is seen while doing keep.

Resident Load:

1. If we want to Perform some Calculations on an already loaded table, then we create a new table and do calculations and we take Resident of that Table.
2. Aggregation will not work on Base Table; in this scenario we use Resident Load.

Preceding Load:

1. In Preceding load, we can do multiple calculation & Transformation within a Load script.
2. If we want to do calculation or any aggregation without creating a new table that time, we use Preceding load.
3. Using a preceding load, you don’t need to perform the same calculation several times.
4. Cross Table, Interval match, Join not supported in Preceding Load.

Note: Preceding Load is Faster than Resident Load.

Incremental Load:

1. If we want to update or insert new records at that time, we use Incremental Load.
2. Incremental Load is of 3 types:
   A- Insert Only (Do not validate Duplicate Records)
   B- Insert & Update (we can delete Duplicate Records from here)
   C- Insert, Update & Delete.

Concatenation:

When we have common fields between both the tables that time we do concatenate.

1. There are 3 Types of Concatenation:
   A-Auto Concatenation
   B- Forced Concatenation
   C-No Concatenation.

Inline load :

Inline load is a type of data load that can be used when you do not have a data source. The data can be entered manually or by using the wizard within Qlik View or Qlik Sense.

Mapping:

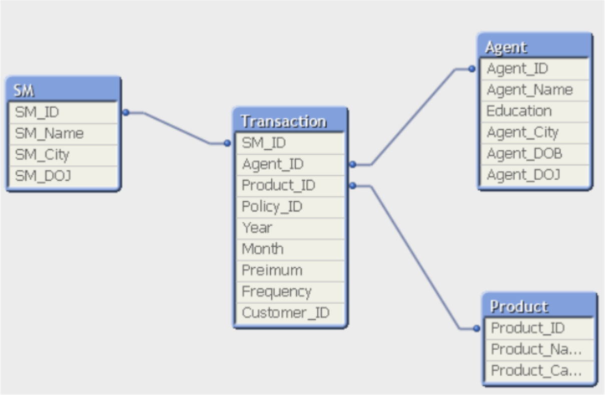
Generally, we do Mapping when we have 2 tables and in 1 Table when we have 2 fields in 1 Table with 1 common key and we have Attribute key that time we do Mapping.

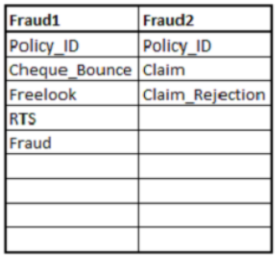
Binary Load:

Binary Load replicates the data model of an existing Qlik View document into another Qlik View document without accessing the original data source**.**

1. While using Binary Load script is hidden.
2. UI & Variables doesn't come when we take Binary load of an application.
3. Only Backend part comes when we take Binary of it, only Data Modelling is shown.
4. Binary Statement is always written as 1st statement.
5. Binary Load includes section access.
6. Time Consumption is less.
7. You can only perform one binary load per script**.**

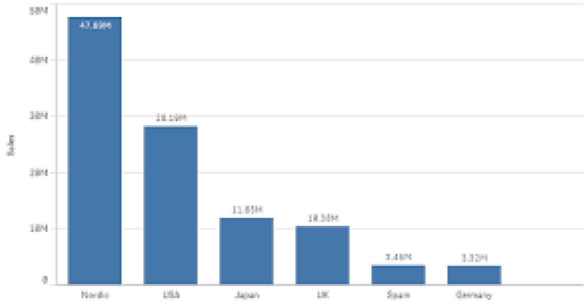
This is an ideal use case for implementing binary load. I already had the data model in Sales dashboard. I can use this existing data model to create a new document and add more tables of fraud metrics for fraud detection.



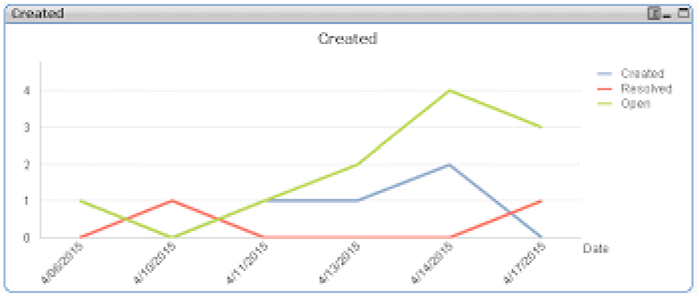


Fraud1	Fraud2
Policy_ID	Policy_ID
Cheque_Bounce	Claim
Freelook	Claim_Rejection
RTS	
Fraud	

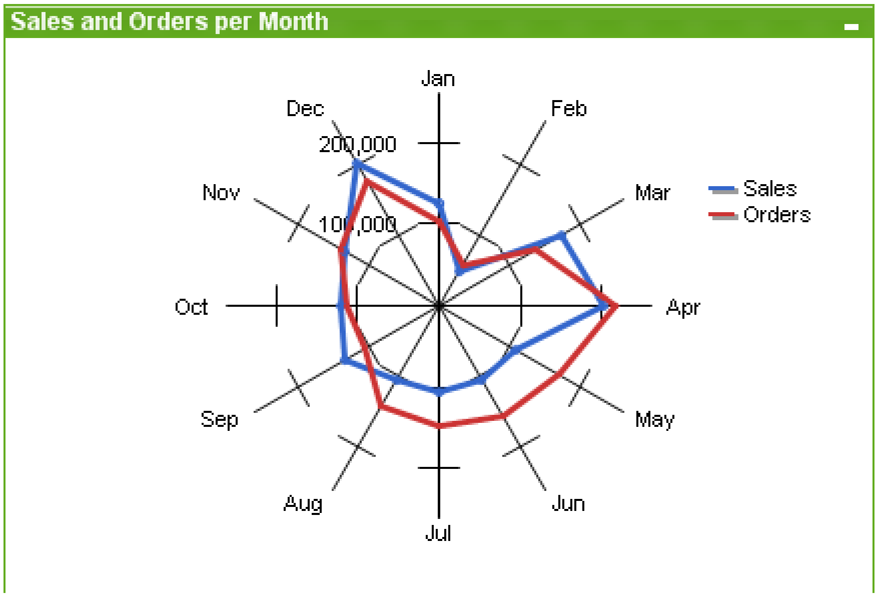
1. **Bar Chart (2 Dimensions and 15 measures):**
   - **If we want to compare information between 2 countries we can use Bar Chart.**
   - **If we want to compare information we can use bar chart.**



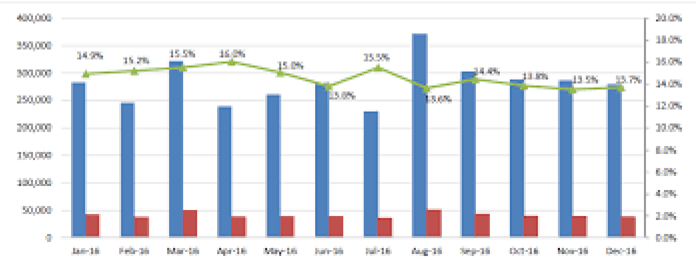
2. **Line Chart (1 Dimension and 1 Measure):**
   - **We show information over period of time.**
   - **To show a trend of a particular country over period of time**
   - **We use Line Chart to display the information.**



3. **Radar Chart: (2 Dimension and 1 Measure)**
   - **They are a great way of representing it in cyclic dimension.**
   - **With the help of Radar Chart, we can visualize the entire year's information in such a way it will help us to know where the sales spiked, which month sales were consistent, which month Sales were low.**

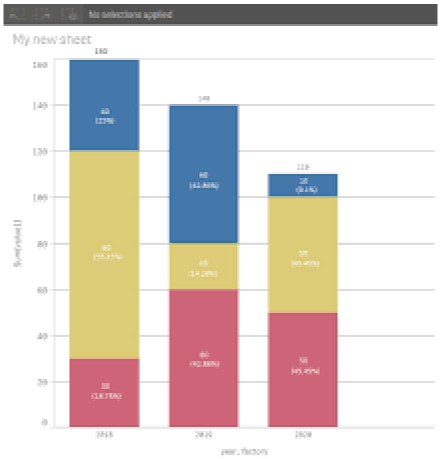


4. **Combo Chart (1 Dimension and 2 measures)**:
   - **It is basically a combination of 2 different charts, Line Chart and Bar Chart.**
   - **The Combo Chart allows you to combine the features of the bar chart with the help of a line chart.**
   - **You can show 1 expression as bars while displaying another expression as Line.**



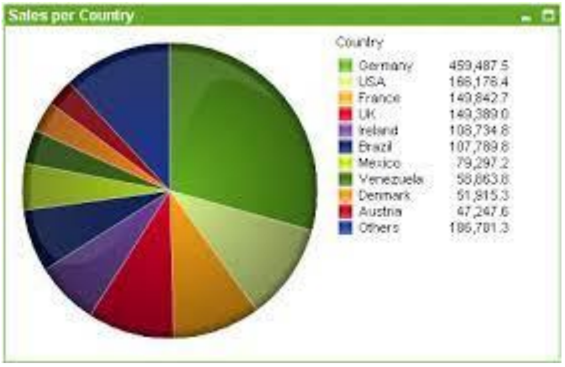
5. **Stacked Bar Chart**:

- **If we want to see a company sale and its projection in just 1 Bar and measure how many projected sales still need to be made in a particular year.**



6. Pie Chart (1 Dimension up to 2 measures):

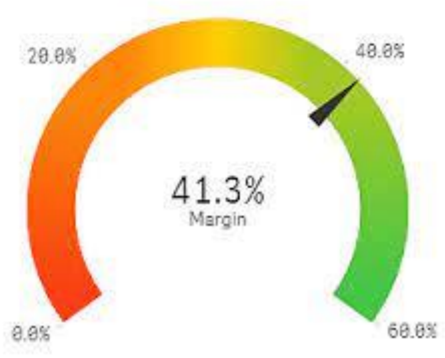
- **We have Sales which we want to see in a circle like how they are distributed by region.**
- **It is used to show the distribution of values over a circle for 1 dimension.**



7. Gauge Chart (Speedometer Chart) (0 Dimension & 1 measures):

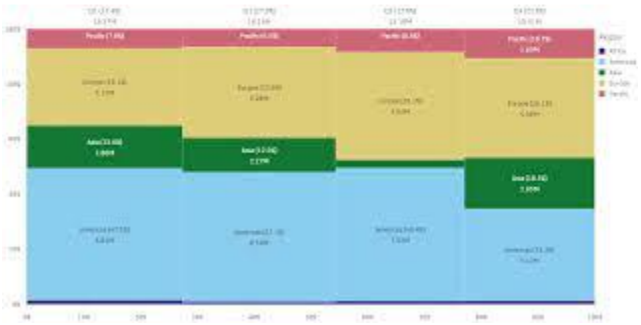
- **Gauge chart is also called as Speedometer chart, that is similar to the component that you usually find in your car and biker dashboard at which speeds I am going or what is the fuel ,extra kind of information on which we can we can make decision if fuel is getting low indication to nearest fuel station and buy some fuel.**

- **How are KPI's or critical business metrics are performing displaying over a gauge chart.**

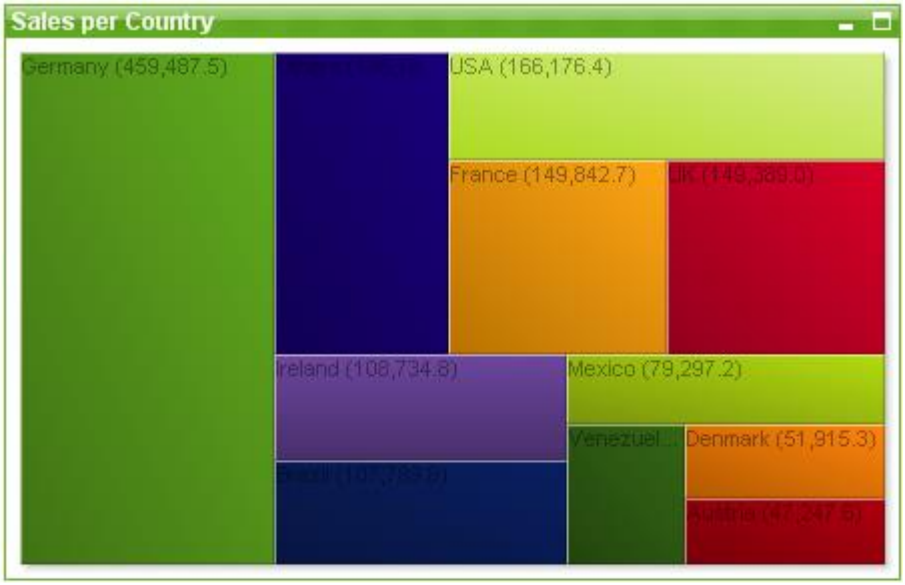


8. **Mekko Chart (2 Dimension and 1 measure):**

    - **Mekko Chart is very similar to Staked Bar Chart, but it has another additional measurement which is the width and that is a great way to display breakdowns in ratios and actual breakdowns.**



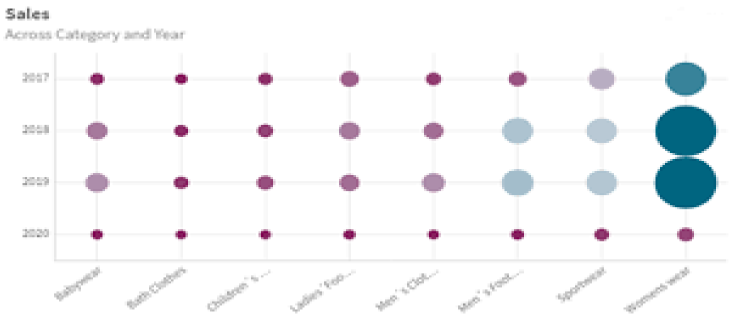
9. **Block Chart (Up to 3 Dimension and 1 Measure):**
    - **It helps us do comparative analysis at multiple levels.**
    - **For example:**
      We have 2 Years where we can do comparative analysis based on the comparison between 2 years.
    - **We will generate respective block that will help us visualize which one is performing better and having a good sales percentage and then making a comparison with another year.**



10. **Scatter Chart: (Up to 3 Dimension and 1 measure)**
    - **Charts are Graphical representation of numerical data.**
    - **It is possible to switch between representatives of an existing Chart.**
    - **The Scatter Chart presents a pair of values from two expressions. This is useful when we want to show data where each instance has 2 numbers.**
    - **Scatter Chart in Qlik view is useful to compare two sets of data visually.**

11.Grid Chart: (2 dimension and 1 measure)

- **The Grid Chart is similar to the scatter chart but plot dimension value on the axes and uses an expression to determine the plot symbol.**
- **These Grid Charts are often used to show the performance or the relationship among similar data points.**
- **It shows the performance over a period of time.**



12.Treemaps:

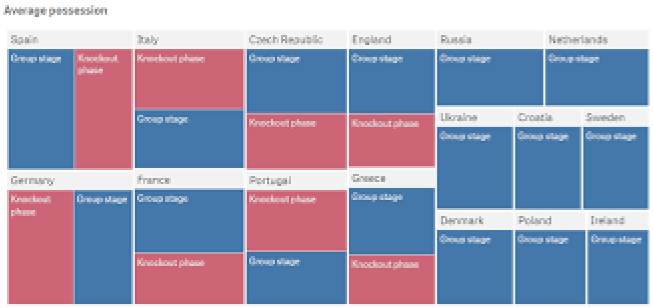
- Use a Tree map when you need a large amount of hierarchical data, but space is constrained.
- Tree maps should be primarily used with values that can be aggregated.

Advantages:

1. The maps are economical in that they can be used within a limited space and yet display a large number of items simultaneously.
2. When there is a correlation between color and size in tree structure you are able to see patterns that would be difficult to spot in other ways.

Disadvantages:

1. Tree maps are less useful when there is a significant big difference in the magnitude of the measured values.
2. Tree maps should not be used when comparing absolute and relative values.
3. Negative values cannot be displayed in Tree maps.



13.Bullet Chart:

- Bullet Chart are gauges with extended options.
- They can be used to visualize and compare performance of a measure to a target value and to qualitative scale.

Section Access:

1. Section Access is actually a way of dynamically reducing data any user in any Qlik Application.

2. Using Section Access we can define which user can see which number of rows or columns or combination of both.

3. There are 2 fields mandatory for any section access table to implement.

- Access

- Use rid

4.*means like whatever its mention in section access table you will see.

5. If the field value is not in upper case, then user will not get access but if your column name or field is not capitalized everyone can see everything, and your section access fails.

6. You can also import section access by taking Binary Load also.

Example:

Year, COUNTRY, Sales

2019, UK, 100

2020, UK, 120

2021, UK, 95

Section Access;

ACCESS	USERID	PASSWORD	COUNTRY
ADMIN	ADMIN	ADMIN	*
USER	SUMEET	1212	INDIA
USER	ASHISH	1413	FRANCE
USER	PAWAN	1111	AUSTRALIA
USER	AMAN	1818	USA
USER	TABREZ	1313	UK

Section Application;

Then Map the data by making a inline load with field COUNTRY.

Then go to Setting-Document Properties-Opening-Initial Data Reduction Based on Section Access.

Then Section Access will happen.

7.**Section Access takes place on sheet level, column level & Row Level.**

Difference between QlikView and Qlik sense?

1. **QlikView is our 1st Generation Analytics Platform.**

2. **QlikView is based on guided Analytics, Application is created by QlikView documents developers who put lot of taught into data model, layout, charts and formulas. They deliver the applications to analysts. Analysts have Total Freedom to explore data, select, drill-down. However, they are limited when it comes to creating new visualizations.**

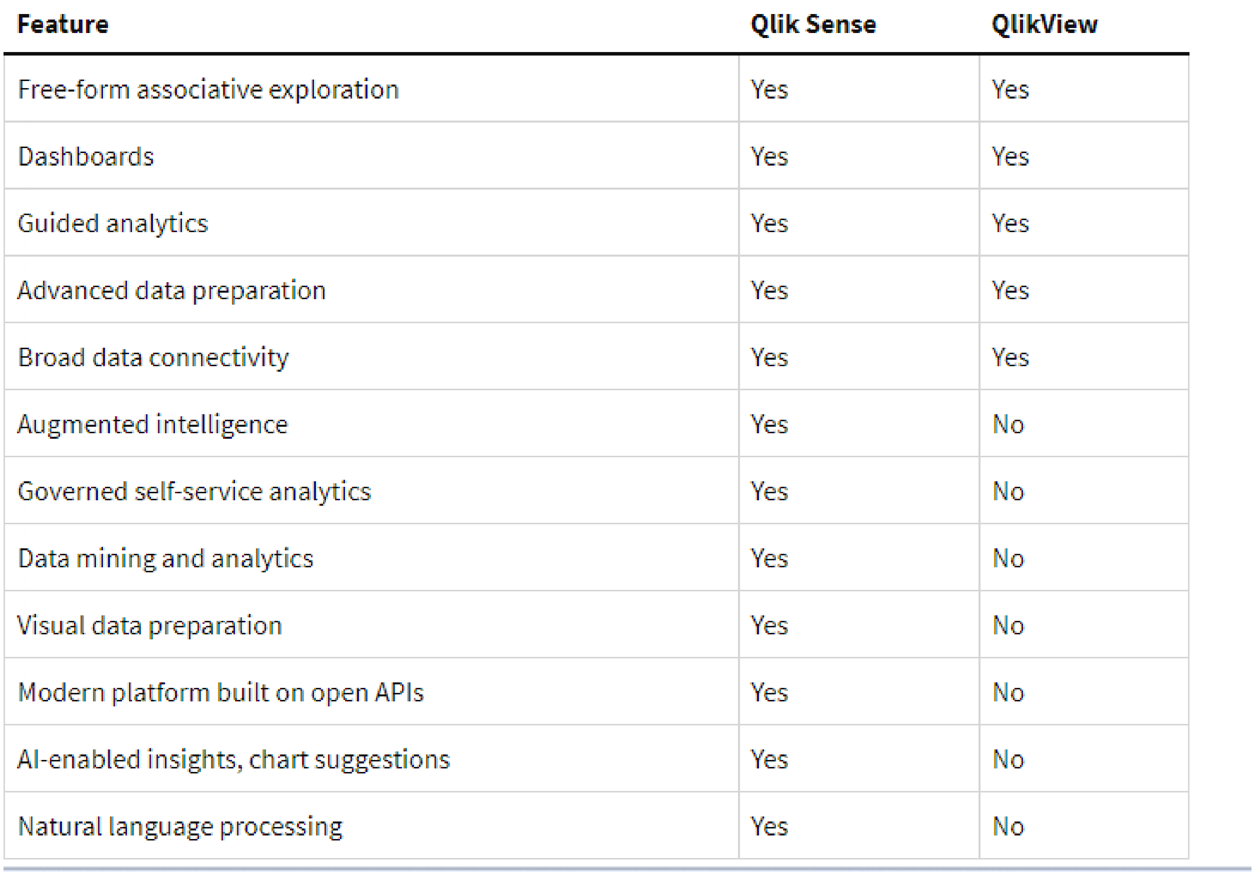
   Qlik sense:

3. **Qlik sense offers modern Analytic solutions.**

4. **Qlik sense offers self-service data discovery. Analysts are free to create new apps, visualization and bookmarks.**

5. **This leads to much more active engaged users. It means there is less data development in the beginning, because Qlik sense app developers do not need to build script that meet every user needs. Qlik sense is much easier to use on touch screen's, and is adaptive to different screen sizes and form factors.**

   Features difference between QlikView & Qlik sense:



Feature	Qlik Sense	QlikView
Free-form associative exploration	Yes	Yes
Dashboards	Yes	Yes
Guided analytics	Yes	Yes
Advanced data preparation	Yes	Yes
Broad data connectivity	Yes	Yes
Augmented intelligence	Yes	No
Governed self-service analytics	Yes	No
Data mining and analytics	Yes	No
Visual data preparation	Yes	No
Modern platform built on open APIs	Yes	No
AI-enabled insights, chart suggestions	Yes	No
Natural language processing	Yes	No

***3 Tier Architecture of Qlik:**

1. **1 Tier Architecture:**
   In Tier 1 Architecture Qlik hits database and pulls data directly and creates Data Model and the visualization is done in single QVF.

2. **2 Tier Architecture:**

First v take data and create data model and in 2nd tier we will take binary load of the QVF and create visualization in the same QVF.

3. 3 Tier Architecture:

1st tier we will fetch data in QlikView application and create QVD's.

In the 2nd tier we will pull QVD's in QlikView application and create Data Model as per user requirement.

In the 3RD Tier we will take Binary Load of the 2nd application where we create Data Model and create the visualization.

Services of Qlik sense:

(1) Qlik Sense Repository Services:

It manages licenses and security of Qlik sense.

(2) Qlik sense Proxy Service:

It is used for authentication, session handling and Load Balance.

(3) Qlik sense Repository Database:

Whatever Data is there it records.

(4) Qlik sense Scheduler Service:

It is basically used for reloading apps.

(5) Qlik Engine Service:

Whatever Data we Load, Logic calculation is run.

(6) Qlik sense Service Dispatcher:

It controls the apps.

*IS Null () Function:

The Is Null () Function checks the values for Null and returns -1(True) if it is Null and 0 (False) if it is not Null.

This function evaluates an expression logically and answers in -1 and 0.

If we have Null value in the table and if if we want to search, then isNull Function is uses.

ID Value Value Null Conv   Is It Null

0 -   - T

1 Null – T

2 - - T

3 Value - F

*Null Function ():

This Function returns a Null value. It is different from IsNull Function as that returns a logical answer in True or False, whereas Null () Function returns '-'in place of Null Value.

Example:

ID	Value	Value NullConv
0		-
1	Null	-
2	-	-
3	29	Value
4	Mark	Value

SET ANALYSIS:

Set Analysis offers a way of defining set(or group) of data values that is different from normal set defined by the current selection.

1. Identifiers:
   Identifiers define the relationship between the set expression and field values or expression being evaluated.

Example: sum({$<Year={2009}>}Sales)

The identifier is the dollar sign . $ and means that the set of records to be evaluated consists of all the records of the current selection.

Identifier	Description
1	Represents the full set of all the records in the application, irrespective of any selections made.
$	Represents the records of the current selection. The set expression **{$}** is thus the equivalent to not stating a set expression.
$1	Represents the previous selection. $2 represents the previous selection-but-one, and so on.
$_1	Represents the next (forward) selection. $_2 represents the next selection-but-one, and so on.
BM01	You can use any bookmark ID or bookmark name.
MyAltState	You can reference the selections made in an alternate state by its state name.

Examples and results

Example	Result
`sum ({1} Sales)`	Returns total sales for the document, disregarding selections but not the dimension.
`sum ({$} Sales)`	Returns the sales for the current selection, that is, the same as sum(Sales).
`sum ({$1} Sales)`	Returns the sales for the previous selection.
`sum ({BM01} Sales)`	Returns the sales for the bookmark named *BM01*.

2. **Operators:**

Operators are used to include, exclude or intersect parts of or whole data sets. All Operators use sets as operands and return a set as a result.

Operator	Description
+	Union. This binary operation returns a set consisting of the records that belong to any of the two set operands.
-	Exclusion. This binary operation returns a set of the records that belong to the first but not the other of the two set operands. Also, when used as a unary operator, it returns the complement set.
*	Intersection. This binary operation returns a set consisting of the records that belong to both of the two set operands.
/	Symmetric difference (XOR). This binary operation returns a set consisting of the records that belong to either, but not both of the two set operands.

Set operators

Operator	Description
+	Union. This binary operation returns a set consisting of the records that belong to any of the two set operands.
-	Exclusion. This binary operation returns a set of the records that belong to the first but not the other of the two set operands. Also, when used as a unary operator, it returns the complement set.
*	Intersection. This binary operation returns a set consisting of the records that belong to both of the two set operands.
/	Symmetric difference (XOR). This binary operation returns a set consisting of the records that belong to either, but not both of the two set operands.

Examples and results

Example	**Result**
`sum( {1-$} Sales )`	Returns sales for everything excluded by the selection.
`sum({$*BM01} Sales)`	Returns sales for the intersection between the selection and bookmark BM01.
`sum({-($+BM01)} Sales)`	Returns sales excluded by the selection and bookmark BM01.
`Sum({$<Year={2009}>+1<Country={'Sweden'}>} Sales)`	Returns sales for the year 2009 associated with the current selections and add the full set of data associated with the country *Sweden* across all years.

3. **Modifier:**

Modifiers are used to make additions or changes to a selection. Such modification can be written in the set expression.

A Modifier consists of one or several fields' names, each followed by one or several selections that can be made in the field.

Modifiers begin and end with <>.

Examples and results

MyField	**Result**
`sum ({$<Order Date = Delivery Date>} Sales)`	Returns the sales for the current selection where Order Date = Delivery Date.
`sum ({1<Region = {US}>} Sales)`	Returns the sales for region US, disregarding the current selection.
`sum ({$<Region = >} Sales)`	Returns the sales for the selection, but with the selection in *Region* removed.
`sum ({<Region = >} Sales)`	Returns the same as the example above. When the set identifier to modify is omitted, the inherited state is assumed.

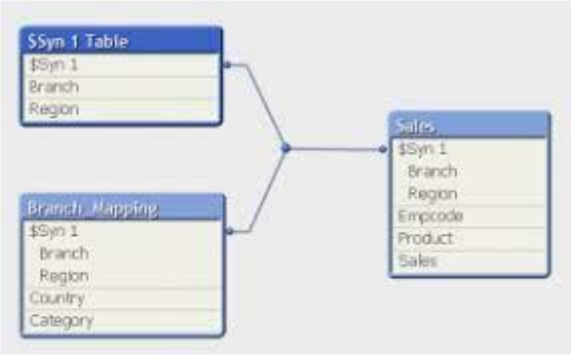
Examples and results

MyField	**Result**
`sum ({$<Year= {2000}, Region={“U*”}>} Sales)`	Returns the sales for the current selection, but with new selections both in *Year* and in *Region*.

***Synthetic Key and Circular Reference:**

1. When 2 tables have more than one common field it will form a synthetic key.

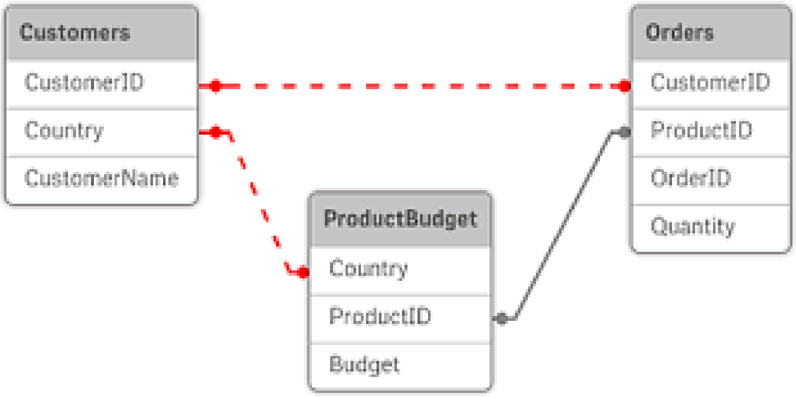
2. It should always be avoided because it will increase load time.



CR:

When 3 or more tables are linked in such a way that it forms a circle.

Like synthetic keys are created, a circular reference can also be the result of unwanted associations in our data modelling.



FUNCTION:

1. **Left:**

   Starting from the left of the string returns the specified number of characters.

   Ex: Left (‘Qlik Sense’,4)

   Ans: Qlik.

2. **Right:**

   Starting from the right of the string, returns the specified number of characters.

   Ex: Right (‘Qlik 2 Express’, 4)

   Ans: Press.

Numeric Function:

1. **Ceil ():**

   It will upgrade the number to topmost value.

   Eg: 9.1 9.5 9.9

   Ans: 10 10 10.

2. **Floor ():**

   It will reduce the number it is opposite of ceil.

Egg: 9.1 9.5 9.9
Ans: 9 9 9

3. Round ():

   Round the number optionally a parameter can be specified to indicate which multiple to round to.

   Eg: 9.1 9.5 9.9

   Ans: 9 10 10

4. Upper: Converts the string to upper case.

   Example: =Upper (‘Qlik 2 Express’)
   ANSWER: QLIK 2 EXPRESS.

5. Lower: Converts the string to lower case.

6. Captalize: It will convert the 1st letter of each word to capital

7. Mid Function:

   Where we want to start the position, number of characters.

   Example: =mid(‘Qlik 2 Express’,8,3)
   Answer: Exp.

8. Index Function:
   Index (column name/string, substring which need to find the position, occurrence (optional)

   Example: =index (‘Qlik 2 Express’), I want to see position of S, by default it will show position of 1st S.

   Answer: 13(it will count space, ‘‘single intverted comma also)

   Example 2=index (‘Qlik 2 Express’, ‘s, 2’) (Which s I want)

Answer: 14.

9. **Replace:**
   It will search the entire string and replace the value.

   Example: = Replace (‘Qlik 2 Express’,’2’, ‘To’)

   Answer: Qlik to Express

   Example: = Replace (‘Qlik 2 Express_2’, ‘2’, ‘To’)

   Answer: Qlik to Express Tô

10: Keep char:

It will keep the value,

11. **Purge char:**
    It will remove the characters.

12. **Subfield:**
    It is used to separate the field.
    Example: (‘ Name/DOJ/Salary/Bonus’, ‘/’,2)

    Answer. DOJ.

13) **Conditional Functions:**
13) alt: alt (date#(date, 'YYYY/MM/DD'),

date#(dat , 'MM/DD/YYYY'),

14) Class (): The class function assigns the first parameter to a class interval.

eg: class (23,10) returns (20-30)

15.) if (): The if function returns a value depending on whether the condition provided with the function evaluates as True or False.

eg:if(Amount>= 0, 'OK', 'Alarm') returns ok otherwise Alarm.

16) Match (): The match function compares the first parameter with all the following ones and returns the number of the expression that matches. The comparison is case sensitive.
match(M,'Jan','Feb','Mar') returns 2 if M=Feb.

17). Mix Match (): The mix match function compares the first parameter with all the following ones and returns the number of the expression that matches. The comparison is case insensitive.

eg: mix match (M, ' Jan','Feb','Mar',’Apr’) returns 1 if M=Jan

18) pick (): It pick function returns the n:th expression in the list.

e.g.: pick (N, 'A','B',4, 6) returns B if N='2'

19) Wild Match (): The wildmatch function compares the first parameter with all the following ones and returns the number of expressions that matches. It permits the use of wildcard characters (* and?) in the comparison strings. The comparison is case insensitive.

eg:wildmatch(M, 'ja*','fee?','mar') returns 1 if M=January or return 2 if M=fex.

20) Peek () & Previous ():
Key Difference:
The key difference between these two functions is where they read records from.
Previous () reads directly from the data source while Peek () reads from data that is already in memory.

- **Both allow you to look back to previously loaded data.**
- **Previous () works on input of load statement.**
- **Peek () works on output of load statement**

Syntax: Peek(‘fieldname’) & Previous(‘fieldname’)

PREVIOUS (): It take cares of data before the filtering
PEEK (): It takes care of data after filtering.

21) Interval Match :The Interval Match prefix is used to create a table matching discrete numeric values to one or more
numeric intervals, and optionally matching the values of one or several additional keys.
Syntax: Interval Match (match field) (load statement | select statement)

22)AutoNumber ():
This script function returns a unique integer value for each distinct evaluated value of expression encountered during the script execution.
Syntax: auto number(expression [, AutoID])

23) Aggr ():
Aggregation the data over the dimensions. Same as group by feature.

- **The Agger function is used for advanced aggregations, in which the Aggr function is enclosed in another aggregation functions.**
- **It is used for defining multiple aggregation function.**

Example: Example: max (agger (sum [Order Total]),
[Customer Name], [Employee ID]).

When should we use AGGR?

- Whenever we do aggregation in two steps like Sum of sales of top selling product by company or Highest average sales by Product and Company, in both statements we are doing two aggregations.
- We want to identify top/ bottom population in segment
- It becomes more powerful when it is used with Set analysis, Rank (), FirstSortedValue().

24) Cross Table (): Cross table is converting rows into columns or vice versa.
Syntax: Cross table (attribute field name, data field name [, n]) (load statement | select statement)

Example: Crosstable (Month, Sales,1) Load * from ex1.xlsx;

25) Range Sum (): Range Sum () returns the sum of a range of values. All non-numeric values are treated as 0, unlike the +
operator.
Syntax: Range Sum (first_expr[, Expression])
Example: 1. Range Sum (1,2,4) Return (7) 2. Range Sum (5,'abc') Return (5)

3. **Range Sum (null ()) Return (0).**

26) Mapping Table & Apply Map ():

- If we want to map the values of a column from one table to another then we need to create a mapping table which should contain only two columns.
- Using Applymap() you can include only one column.
- Use Mapping load then in Mapping table needs two fields at least for mapping.
- It helps us in reducing the number of tables involved in the data model.
- The first column should be the common key column or lookup field

Points to Remember:

- A Mapping table should be loaded before using the Apply Map function.
- Mapping table should have only two columns.
- First Column in the Mapping table should be a key field.
- Mapping Keyword is compulsory before loading any mapping table.
- You cannot use Apply Map function when you need more than one corresponding value from the second table (Fetches single value per matching record)
- In Apply Map () Mapping table name must be enclosed under single quotes.

Syntax: Apply Map (‘Mapping Table Name’, Common Key Column, ‘Default Value’);

Example:

```
Table1:
Mapping
LOAD
Invoice ID as ReceiptInvoiceID,
Amount as Receipt Amount
From {path}
Table2:
LOAD
InvoiceDetailID,
Description,
```

Invoiced,
Apply Map (‘Table1’, Invoiced, 0) as Receipt Amount
FROM {path}
Result:
It will display only one table which contains all fields of applyMap table.

27) Date & Date#:
Date () is called Formatting function. It will convert to number-to-string format.
Syntax: Date (number [, format])
Date# () function is called Interpretation function. It will convert the string -to-number format.
Syntax: Date# (text [, format])
Example:
Sample Date: 2013_24_10
date(date#('2013_24_10','YYYY_DD_MM'),'MM/DD/YYYY')
Returns: 10/24/2013

28) Rank ():
The rank () function is used to rank the values in a dimension contained in rows. Values in each row are
evaluated and compared with values in the other rows within the same dimension or field. So, after the
comparison of values in the rows, the rows or corresponding values are ranked based on how big or small the value is as compared to other values. For example, 100 will be ranked 1st and 95 will be ranked
2nd and so on. This function returns a value in dual format i.e., in both text and numeric representation.
Syntax: Rank([TOTAL] expr [, mode [, fmt]])
=Aggr(Rank (Sum (Sales)), Product) & '. ' & Product =Aggr(Rank (Sum (Sales)),4,1), Product) & '. ' & Product.

29) **Dual ():**
Dual () combines a number and a string into a single record, such that the number representation of the record can be used for sorting and calculation purposes, while the string value can be used for display purposes.
Syntax: Dual (text, number)
Dual video
https://www.youtube.com/watch?v=kl5jquqD5yo

30) FirstSortedValue():
FirstSortedValue() returns the value from the expression specified in value that corresponds to the result of sorting the sort of weight argument, considering rank, if specified. If more than one resulting value shares the same sort of weight for the specified rank, the function returns NULL.
The sorted values are iterated over a few records, as defined by a group by clause, or aggregated across the full data set if no group by clause is defined.
Syntax: FirstSortedValue([distinct] value, sort-weight [, rank])
Example:
Agger Max Sale: =max (agger (sum (Sale), [Company Name], Product))
Agger Min Product: = min (agger (sum (Sale), Product, [Company Name]))

MinProductSales: =FirstSortedValue(Product,Aggr(sum (Sale), Product, [Company Name]))
MaxProductSales: =FirstSortedValue(Product, -Aggr(sum (Sale), Product, [Company Name]).

31) **Link table and concatenate ():**
Link Table:
• Link table used for connection two tables with newly generated table. It is used for removing synthetic keys between tables.
• In Data warehousing - Star schema usually have a single Fact in the center and dimensions surrounding it but sometimes you have a situation where you have more than one Fact table.
• In QlikView, you come across a situation where you must handle multiple Fact tables, which in turn create synthetic keys or Circular references (Loops).
• Link Table is a very useful way to resolve synthetic keys and Circular references (Loops) when multiple fact tables are present in the data model.

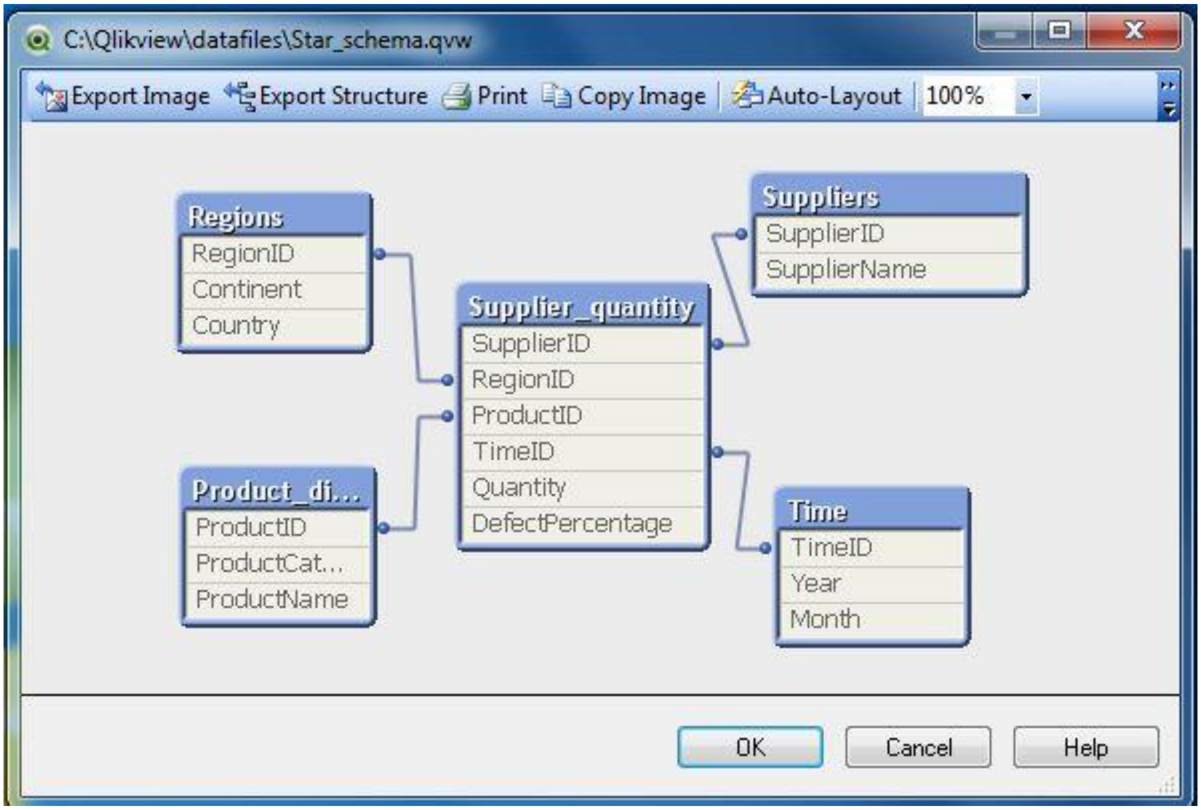
32) **Concatenate:**
• When you have multiple Fact tables in QlikView, it can be handled in 2 ways, by using concatenate or by using Link tables. If the granularity and columns in the fact tables are same then you can use Concatenate which will append the tables into one and resulting table will have the sum of rows of the two tables.
• Use Link table, when the granularity of the facts tables is different and when they are joined to different dimensions.
• Concatenate and Link Tables are ways to resolve synthetic keys which formulate due to the presence of multiple fact tables.

33) AutonumberHash256():
This script function calculates a 256-bit hash of the combined input expression values and returns a unique integer value for each distinct hash value encountered during the script execution. This function
can be used e.g., for creating a compact memory representation of a complex key.
Create Composite key.
Syntax: autonumberhash256(expression {, expression})

* Difference between Star Flake & Snowflake Schema:

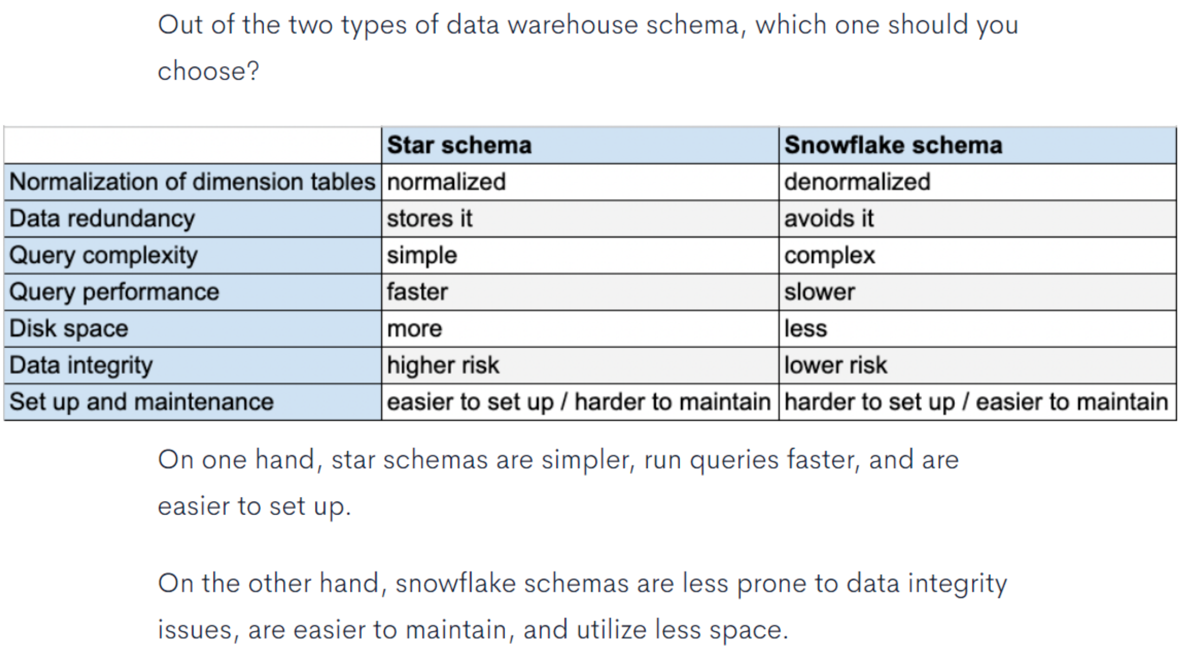
Star:

1. **In a star schema, the fact table will be at the center and is connected to dimensional tables.**
2. **The Tables are completely denormalized in structure.**
3. **SQL Queries is good as there are a smaller number of joins involved.**
4. **Data redundancy is high. Occupies more disk spaces.**



Snowflake:

1. **It's an extension of star schema where dimension tables are further connected to one or more dimensions.**
2. **The Tables are partially denormalized in structure.**
3. **The performance of SQL Queries is little bit less compared to star schema as a more number of**
4. **joins are involved.**
5. **Data redundancy is low.occupies less disk space compared to star schema.**



Out of the two types of data warehouse schema, which one should you choose?

	Star schema	Snowflake schema
Normalization of dimension tables	normalized	denormalized
Data redundancy	stores it	avoids it
Query complexity	simple	complex
Query performance	faster	slower
Disk space	more	less
Data integrity	higher risk	lower risk
Set up and maintenance	easier to set up / harder to maintain	harder to set up / easier to maintain

On one hand, star schemas are simpler, run queries faster, and are easier to set up.

On the other hand, snowflake schemas are less prone to data integrity issues, are easier to maintain, and utilize less space.

Difference between Pivot Table and straight Table:

Pivot Table:

Multi–Dimensional with one or more measures.

Displays sub-totals.

Will show cross tables.

Data is grouped.

Straight Table:

One Dimension with multiple measurements.

Do not display sub-Totals.

Won’t show cross table.

Data is not grouped.

What is Fact Table:

Fact Tables contains the measurable attributes of the data.

Fact tables contain measures on which we can do further calculations to get analysis.

What is Dimension Table:

Dimension Table contains the textual or descriptive attribute of the data.

For example: Customer dimension will contain details about customer's name,address, phone number etc.

What is Link Table:

When 2 or more tables are not connected with each other with any of the primary key then we create connection between each other.

What is N-Printing:

N-Printing enables you to schedule and automatically distribute reports to multiple users via mail, google drive, or folder.

N-Printing reads the Qlikview charts directly from dashboards in qlikview server and generated the reports based on configuration and distribute those in various tables.