

## CLASS # 1

# Modern Challenges for everyone



**Business Intelligence (BI)** refers to technologies, applications and practices for the collection, integration, analysis, and presentation of business information. The purpose of Business Intelligence is to support better business decision making.

Remaining is in pdf file

## CLASS # 1

### Website – Plane Crash Info

For 1974 data:

Steps: Go to Excel – Then Web – Paste URL  
(<https://www.planecrashinfo.com/1974/1974.htm>) – Table – Transform Data – Go to Power Query

Next:

Select 2columns using shift key – Go to replace values – Then Advance options – Replace using special characters(Carriage return) – Replace with(space) – Okay (It will convert all words into a single line)

Can also change data type – Date, text, float no etc

To split the column – Select the column – Go to split column (by mean s of delimiter) – left to first occurrence

To get rid of closing bracket – replace value with nothing – then okay

To give title – check the data and replace name – Faculties, Aboard, Ground

Rename query – Query-1

#### **Convert query into function – to get all years data**

Go to Home Tab – Then to Advance Editor – It shows all the steps we have followed

#### Change in code:

(year)=> (this is function)

Replace value with this - “&Number.ToString(year) &” / “&Number.ToString(year) &” – Done

This will convert query into function

Now, if we write 2001 in function – it will give 2001 data

Now, if we write 1995 in function – it will give 1995 data

#### **How to make your query updated automatically**

Copy url that has all years

Then, go to query pane – New query – Other Sources – Website (paste url)  
<https://www.planecrashinfo.com/database.htm>

Table – Accent Database1 – Okay

To make all the years into a single column

Shift + right arrow – select all the columns

Go to transform data – Unpivot – it converts all column in a single column

Remove column titles – all are now in single column

**Example** – Selecting year using drop down from (1974 to 1800)

Delete other things – now only have function and years data

Rename year's data as Data

### **To activate the function**

Go to Add Column – Invoke Custom function – Query-1 – year(column name) – Okay –  
An electronic table generate – to expand it – Original column – Okay

Change data type

### **Load data into Excel**

Go to Home – Close and Load

## CLASS # 2

# What is Power Query

Power Query is an ETL tool. Its function is to:

- ✓ Extract data from almost every source.
- ✓ Transform data.
- ✓ Load the data.

- Power query is a macro recorder that keeps track of data Extract and Transform steps.
- Define query once and determine where you want to load your data.
- Refresh your query to run it again.



Power Query includes -

- Menu
- Query Pane
- Query Settings – Steps that we use to clean the data
- M-language used for power query

Power BI include –

- Canvas of report
- Tabular form of data
- Model view of data
- Filters pane
- Visualization pane – include Visual tab and Formatting tab
- Data pane
- Mobile layout
- Pages

## Import Excel Data

### Excel Tabular Data

- Excel will kick off the process of creating official Excel Table.
- On Confirmation, it will launch Power Query Interface.
- Table Headers will be imported from the Official Excel Table, so there is no Promoted Headers step.

### Named Range

- Define a Named Range that covers the data.
- Select the Named Range.
- Create a new query: From Table

- Data given in form of table – Tabular Data
- Data in tabular form but formatted as excel table – means it can expand to the right or to the bottom
- Pivot Table – It cannot expand unless you do
- Range Data

## **HOW TO EXTRACT DATA FROM CSV FILE**

### **Jan 2008.csv file**

4 columns + Grand total

1<sup>st</sup> column – Date (but at last it contain text) – Date should be on US System

Go to Power BI – Get data – Text/csv – jan.2008.csv – Transform data – Power query open

Change Data type

To remove error – Go to home – reduce rows – then reduce errors (Or)

Another way – Go to home – reduce rows – then reduce bottom row – Number of rows = 1 – Okay

Close and Apply

## **TO CONNECT MULTIPLE FILES IN ONE FOLDER – CSV FILE**

### **Session-1-Ex-5-Start**

Go to Power BI – Get data – Folder name - Session-1-Ex-5-Start – Transform data – Power query open

Select 1<sup>st</sup> and 2<sup>nd</sup> column – Remove other columns

In 2<sup>nd</sup> column – Split by delimiter – Custom. (dot) – Right most delimiter – remove csv column

To create electronic data of the files – Add column – Custom column

Csv.Document([Content]) = Okay

Remove 1<sup>st</sup> column – as it is not needed

Expand the electronic table

Automatically name will arrange – Go to Home tab – Use first row as a header

Go to specific column – 2<sup>nd</sup> column – Home tab – remove row – remove errors

Change 1<sup>st</sup> column name as “File Name” – Then change all data types

Close and Apply

## **TO CONNECT MULTIPLE FOLDER IN ONE FOLDER – CSV FILE**

Select all the Folder – Then Ctrl+C – Copy all the folder as the subfolder in that main folder

Go to home – then to the query – then Refresh

It will connect all files – and get and transform the data

### **EXAMPLE**

#### **Session-2-Ex-1-Unpivot Data**

Go to Power BI – Get data – Folder name - Session-2-Ex-1-Unpivot Data – Transform data – Power query open

Except Content column – Remove all other columns

To create electronic data of the files – Add column – Custom column

Excel.Workbook([Content]) = Okay

If any file is hidden – then in data it is True (if not it is written False)

Remove 1<sup>st</sup> column and then expand the electronic column

Then select 1<sup>st</sup> and 2<sup>nd</sup> column – and remove other columns

Expand Data column

Go to Home – Remove rows – Remove row from the top – No of rows=1 – Okay

Home – Use first row as a header

Then in Date column – expand and uncheck date and total and click okay – to get rid of text in data

3<sup>rd</sup> column – uncheck (sales as product) – to get rid of text in data

Remove Total column

1<sup>st</sup> column – Rename as Country name

Select 1<sup>st</sup> and 2<sup>nd</sup> Column – Go to Transform – Then unpivot other columns

Rename – 3<sup>rd</sup> column = Product and 4<sup>th</sup> column = Sales

Change data types

Close and Apply

## **EXAMPLE**

### **France.xlsx**

Data is in form of excel table

There is a range in excel table – As France range

Sheet name – France + also contains hidden sheet as Additional Information

Go to Power BI – Get data – Folder name - Session-1-Ex-7-Other Workbooks – Transform data – Power query open

Select 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> column and remove rest of the columns

To lowercase all the things in column – Select the column then right click on title – Go to transform and then to – lowercase

To get only excel files from the data – go to replace value of the extension column.(dot) with space – okay

Then split the column – by number of characters = 3 – Okay

Remove the 4<sup>th</sup> column

Then in extension column unselect everything and only select xls

Then remove 3<sup>rd</sup> column

Then for 2<sup>nd</sup> column – split the column by delimiter

Remove extension column

Rename column as = File Name

To extract data from electronic table

Add column – custom column: Excel.Workbook([Content]) = Okay

Expand the data

Uncheck True sheets

Except Sheet – uncheck everything

Remove last 3 column and name column and 1<sup>st</sup> column

Expand data from data column

Transform – first row as header

Rename first column as – File Name

Red marks show error – Uncheck Products from column

Black marks show empty shells – Uncheck null values from date column

Close and Apply

### **CLASS # 3**

#### **Transposing and Unpivoting Complex Data**

#### **Session-4-Ex-2-Unpivoting Subcategorized Table.xlsx**

To convert flat data into excel table – Go to Home tab – Format as Table

If first row is blank – Uncheck header

But do not do this.

---

Select the data – assign name – My Data – Means create name range (dynamic range)

Go to Power BI – Get data – Folder name - Session-4-Ex-2-Unpivoting Subcategorized Table.xlsx – My Data – Transform data – Power query open

Power Query cannot copy paste – but information can be filled to the bottom or upward but not in a row towards left or right

Flip the data – Transform – Transpose (convert row into column and vice versa), Fill the 2<sup>nd</sup> column – Transform – downwards

Column 2 and 3 are interlinked – so we can merge – select both columns – Transform – merge columns – Separator – Custom (|) – Okay

Then Transpose to get data in original state

Fill the 1st column – Transform – downwards and Home – Remove rows – Remove top row

Home – First row as header

Select column 1 and 2 – Transform – Unpivot other columns

3<sup>rd</sup> column – Split column by delimiter and Rename – Revenue, Product, Month, Category, Sales

Products – uncheck null values

Close and apply

## Grouping and Summarizing Data

### Session-4-Ex-1- Grouping and Summarizing Data.xlsx

## Grouping and Summarizing Data

When you are working with large amount of data, you need to group and summarize records at the sources level. This will help you to avoid importing unnecessary data for analysis and reports.

The screenshot shows a Microsoft Excel table with columns: Date, Country, No of Items, and Total Sales Per Country. The data includes rows for Australia, Canada, England, Japan, Norway, Portugal, Spain, Sweden, and USA. To the right, a 'Applied Steps' pane lists: Source, Changed Type, Changed Type1, and Grouped Rows (highlighted in green). A 'Grouping' section describes summarizing total sales by country. A 'Summarizing' section describes calculating proportions and percentages. A formula bar at the bottom shows: =Table.Max([Details], "Amount")

Date	Country	No of Items	Total Sales Per Country
01/01/2016	Australia	8	1490672
01/01/2016	Canada	8	1252198
01/01/2016	England	8	1498163
01/01/2016	Japan	8	1249455
01/01/2016	Norway	8	1486603
01/01/2016	Portugal	8	1568224
01/01/2016	Spain	8	1113601
01/01/2016	Sweden	8	1612891
01/01/2016	USA	8	1232421

**APPLIED STEPS**

- Source
- Changed Type
- Changed Type1
- Grouped Rows

**Grouping**

Total sales each day by country with number of items that made-up Total sales.

**Summarizing**

The largest Brand's proportion and Percentage in Total Daily Sales.

=Table.Max([Details], "Amount")

Excel table expands to right and to the bottom

Go to Power BI – Get data – Folder name - Session-4-Ex-1- Grouping and Summarizing Data.xlsx – Table 1 – Transform data – Power query open

**To group data** – Home – Group by – Advanced – Date – Add grouping (Country)

(if we do move up then it will group first country then date)

Column name (Total Sales) – Operation (Sum) – Column (Amount)

Add aggregation – Number of items – Count rows – Empty – Okay

Go to grouping gear option – Add aggregation – Details – All rows – Okay

### Largest Bands in proportion and the Percentage in Total Daily Sales

Add column – Custom Column – Name (Highest Band)

Table.Max([Details], "Amount")

Remove 2<sup>nd</sup> last column

Expand Brand and Amount - Change data type

Add column – Custom Column – Name (Sales Percentage)

Amount / Total Column – Close and Apply

## **TEXT FILE**

### **Conditional Logic in Power Query**

#### **Session-4-Ex-4-Conditional Logic in Power Query**

Go to Power BI – Get data – File name - Session-4-Ex-4-Conditional Logic in Power Query– Transform data – Power query open

Home – Remove rows – Remove from top – Number of rows = 4

Home – First row as header

Remove last 3 columns

Add column – Custom Column – Name (Time Out)

try Time.From([Out]) otherwise null – Okay

Add column – Custom Column – Name (Employee Name)

if [TimeOut] = null then [Out] else null – Okay

Select Employee column – Transform – Fill – Downwards

Uncheck Cont. worker and total from 1<sup>st</sup> column

Move last column towards 1<sup>st</sup> column

Remove Time out

Change data types

Remove null value from 1<sup>st</sup> column

Close and Apply

## **CLASS # 4**

V-lookup – used for connecting datasets that has something common between them (trying to match some pattern).

It works only when there is at least one common value between both datasets (join between two datasets)

### **How to connect two datasets**

#### **Session-2-Ex-3A-Merge**

1<sup>st</sup> Dataset columns: Airline name, Country, Continent, Carrier Code, Landing Charges

2<sup>nd</sup> Dataset columns: Date, Carrier Code, Flight Code

Go to Power BI – Get data – File name - Session-2-Ex-3A-Merge in Power Query– Select Table Airline1 and Flight2 – Transform data – Power query open

### **To Merge Datasets**

Go to Home – Merge queries (Merge queries as new) – Select common column between two datasets (here it is Carrier Code) – Select ok

Change Name of query – My Data

Expand Airline column – Uncheck Carrier Code and Expand rest of the columns.

Close and Apply

### **Transposing Stacked Data**

Dataset include: Date, Paid To, Transaction Reference and Amount, and then 2blank rows (these make one record)

Date type – Time Stamp CS Date starts from 1<sup>st</sup> Jan 1900 (days passed written as serial number e.g. 42303)

### **In stack data:**

- First we need to find hidden pattern
- Assign Index number to every information (helps to find starting and end point)
- Need to tell power query what is pattern (using Modulo column)

### Session-4-Ex-3-Transpose Stacked Table

Go to Power BI – Get data – File name - Session-4-Ex-3-Transpose Stacked Table in Power Query – Select My Data – Transform data – Power query open

To add Index Number – Go to add column – Then to Index Column

To add pattern in – Go to add column – Standard – Modulo (value = 6)

Go to add column – Conditional Column:

- Name (Date) – Column Name (Modulo) – Value = 0 – Output (Select from - Transaction) column – Else (null) – Okay
- Name (Paid To) – Column Name (Modulo) – Value = 1 – Output (Select from - Transaction) column – Else (null) – Okay
- Name (Reference) – Column Name (Modulo) – Value = 2 – Output (Select from - Transaction) column – Else (null) – Okay
- Name (Amount) – Column Name (Modulo) – Value = 3 – Output (Select from - Transaction) column – Else (null) – Okay

Select Modulo column – uncheck 4 and 5 (to reduce null values from column)

To fill all the null values of 4 columns (Date, Paid To, Reference, Amount) – Select these four column – Go to Transform – Fill Up

Select Modulo column – uncheck all except 0

Remove all other column except the required four columns and set its data types

Date, Text (Paid to, Reference), Amount (decimal no)

Close and Apply

## Importing Web Data

One of the really interesting use cases for Power Query is when leveraging it to pull data relevant to your business from the web.  
**CSS : Cascading Style Sheets.**

### Pages with Tables

- Use Anonymous authentication.
- [https://en.wikipedia.org/wiki/List\\_of\\_countries\\_and\\_dependencies\\_by\\_population](https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population)

### Web hosted files

- It will take you directly to Power Query environment.
- <https://www.domain.com/robots.txt>

### Pages without Tables

- Use HTML source code
- Not in the scope of this course

### To get data from pdf file

Go to Power BI – Get data – File name – Power Query pdf file in Power Query– Select both tables – Transform data – Power query open

#### Select Page001 dataset first

To remove 1<sup>st</sup> blank row – Go to Home – Remove row from top – Number of row = 1

Go to Home – First row as a header

Remove last total column

From Date column – uncheck total

Select first column – Right click and unpivot other columns

Rename 2<sup>nd</sup> column as Product and 3<sup>rd</sup> column as Sales

#### For Table 001 Data set

Remove last total column

From Date column – uncheck total

Select first column – Right click and un-pivot other columns

Rename 1<sup>st</sup> column as Date, 2<sup>nd</sup> column as Product and 3<sup>rd</sup> column as Sales

Go to Home Tab – Append – Append as new query – Query1 (Page001) and Query2 (Table001) – Okay

Close and Apply

### Website – Aviation Safety Network

**Challenge: How we connect to website + how we connect to years + How to tell every year has how many number of pages**

Go to Power BI – Get data – From URL – <https://aviation-safety.net/database/> in Power Query– Select Table1

But here table is not recognizing – So do the following step:

Check Add table as an example (at the bottom of page) – Column (add 1919, then 1920 – Move) – Rename column (Year) – Okay

Select Table2 (from custom table) – Transform data – Power query open

## Rename query as Years

Now select the certain year (e.g. 1945) then copy the url, in power query create new query – from web – paste the url – okay – Add table using example – Write 1 in column – scroll down the given option (you find <1>234.....15) – Select the option – okay

Right click the column – Replace value < with nothing and again replace value > with nothing

Right click – Split column – by space – Each occurrence of delimiter – okay

Select all the column – Transform – un-pivot all columns

Remove the first column as it has no required information

Select the 2<sup>nd</sup> column – uncheck blank. Rename as “Page Number” and change data type as “Whole no”. Rename query as “Pages” – These are the page number of only year

## To get page number of all years

Go to Home – Advanced editor:

- Declare variable in m-language: (year) =>
- In Source code (in place of 1945 write): “&Number.ToText(year)&”
- Done

It converts query into a function – It tells number of pages of the year we enter.

Let's filter some years – you can do it on all years all well

To filter check only 1924, 1975, 2009, 2022

In year query – Add a column – Invoke a custom column – pages – year – okay – expand the column – change data type as ‘whole no’

Go to the same url of 1945 copy the url - in power query create new query – from web – paste the url – Table 1 – okay

Remove last four columns – as it is not required

Rename query as My Data

## To get information of all years

Go to Home – Advanced editor:

- Declare variable in m-language: (year, pagenumber) =>
- In Source code (in place of 1945 write): “&Number.ToText(year)&/  
“&Number.ToText(pagenumber)&”
- Done

It converts query into a function – It tells information of the year and page number we enter.

In year query – Add a column – Invoke a custom column – My Data – year – 2<sup>nd</sup> (from column) – page number – okay – expand the column – change data type as ‘whole no’

## LOCKBOOK EXERCISE

Get data – text.csv – Select file (Process Lockboxes\_GFS.txt) – Transform data

Transform – Format – Trim (Trim removes ends and beginning spaces of text)

## Filtration

Add new column – conditional column: if Column Name (Column1), Operator (begins with), Value (Batch Name), Output (1) else(null)

Add new column – conditional column: if Column Name (Column1), Operator (begins with), Value (Applied), Output (2) else(null)

**Note:** But instead of filtration in multiple columns – add two if statement in same column.

Add new column – conditional column: if Column Name (Column1), Operator (begins with), Value (Batch Name), Output (1) +

Else if – if Column Name (Column1), Operator (begins with), Value (Applied), Output (2) +

Else if – if Column Name (Column1), Operator (begins with), Value (On Account), Output (3) +

Else if – if Column Name (Column1), Operator (begins with), Value (Cash Claims), Output (4) +

Else if – if Column Name (Column1), Operator (begins with), Value (Unapplied), Output (5) +

Else if – if Column Name (Column1), Operator (begins with), Value (Unidentified), Output (6) + else(null) – okay

In Column – unselect everything – except 1,2,3,4,5,6

### **Separate Batch Name in separate column**

In add column – Custom column: if Text.StartsWith([Column1], "Batch Name ") then Text.End([Column1], 5) else null

Change data type of column to text

Select column1 – transform – fill – fill down

Remove previous column and rename column1 as Batch Number

### **Remove Batch Number from the first column – as batch number is separated now into a new column**

Add new column – conditional column: if Column Name (Column1), Operator (begins with), Value (Batch Name), Output (1) else(null)

Unselect everything and select null value only

Remove this column

Now Select column no 1 – transform – format – trim

### **To remove Applied line item from the 1<sup>st</sup> column – as it is not required**

Add new column – conditional column: if Column Name (Column1), Operator (begins with), Value (Applied Line Item), Output (1) else(null)

Now select only null values

Then remove the column

### **Now slice the trimmed column into 4 different columns**

Note: Spaces are not same between every line – but it is seen that minimum 4 spaces are available in every line

Right click 1<sup>st</sup> column – replace values – value of find ( - 4spaces) – replace with () – ok

Right click 1<sup>st</sup> column – replace values – value of find (||) – replace with () – ok

Right click 1<sup>st</sup> column – split column – by delimiter () – each occurrence – ok

Rename: 1<sup>st</sup> column as Description + 2<sup>nd</sup> column as Count + 3<sup>rd</sup> column as Count Percentage + 4<sup>th</sup> column as Amount + 5<sup>th</sup> column as Amount percentage

Drag Batch Number column as 1<sup>st</sup>

## Cleaning

Select Count and Count Percentage column – transform – format – trim

Right click on Select Count Column – replace value – value to find (--) – replace ( -blank space) – okay

Right click on Select Count Percentage – replace value – value to find (----) – replace ( -blank space) – okay

Change Count Column data type as whole number and Count Percentage data type as decimal number

## CLASS # 5

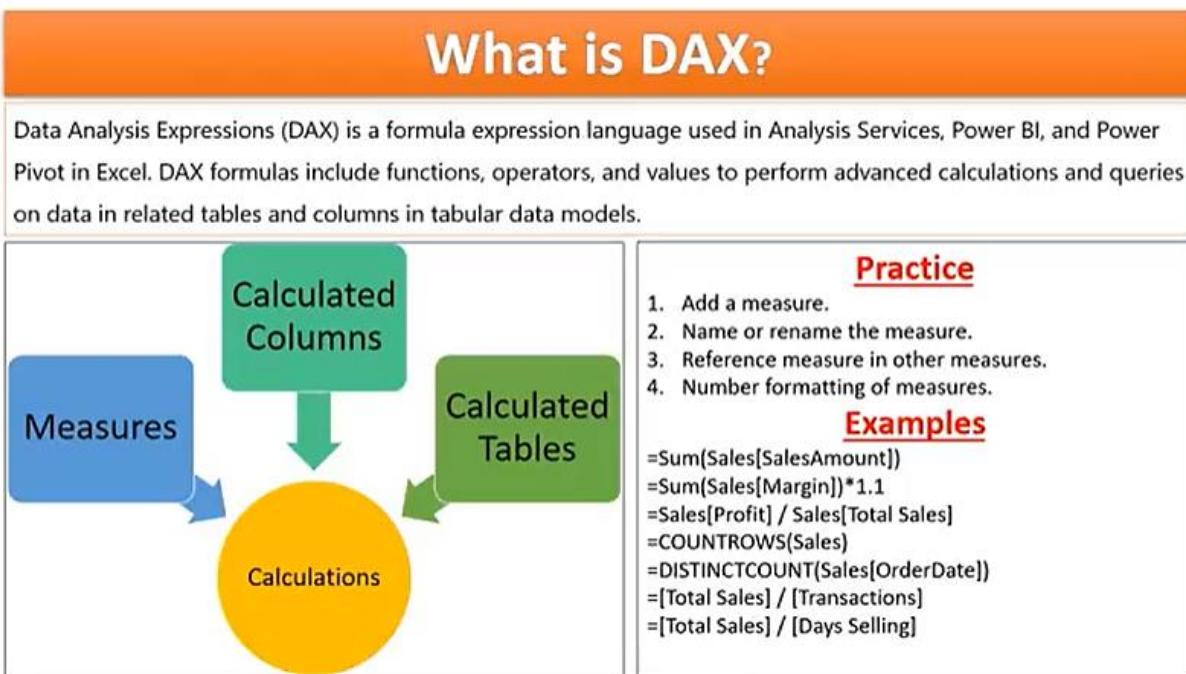
### Module 2- Data Analysis Expression – DAX

It works on two basic principles of aggregating and displaying the data: Filters and Relationships (Interlink between tables)

DAX is a formula language.

DAX functions apply on entire table or sheet. It does not apply on a particular cell.

Calculations in DAX done by three ways: Measures (mostly used), Calculated Columns, Calculated Tables



### WHAT ARE MEASURES?

Measures are dynamic calculation formulas where the results change depending on context.

Measures are used in reporting that support combining and filtering model data by using multiple attributes.

In DAX – **measures** are portable functions / dynamic calculation (filter data on current filter context)

Measures filter data on current filter context.

Note: Avoid adding extra columns in the data.

## QUESTIONS:

1. What was total sales amount for each month.
2. What was total sales margin for each month.
3. What was profit Ratio for each month.
4. How many transactions made up the total sales of each month.
5. How many days made up the total sales of each month.
6. What was average Sales per Transaction for each month.
7. What was average Sales per Day for each month.
8. What was Total profit for Product Key 600 for each month.
9. What was total Sales for Product 214 and 217.
10. What was Total Cost of Product 475 with Black Color.
11. What was Total Profit of Year 2002 for each month.
12. Create a report to show Total Sales per year with break down by Gender. Modify the above report to include Total Sales made to those whose Occupation was "Management".
13. Create a report to show Total Profit earned over the years for each Product Sub Category. Modify the report to show what Profit was earned from customers who were University Graduate. Modify the report to show what Profit was earned from University Graduates who were married.
14. Create a report to show Total Sales for each month with Slicer for Years.

## What are measures?

- Measures are dynamic calculation formulas where the results change depending on context.
- Measures are used in reporting that support combining and filtering model data by using multiple attributes.

1. What was total sales amount for each month.
2. What was total sales margin for each month.
3. What was profit Ratio for each month.
4. How many transactions made up the total sales of each month.
5. How many days made up the total sales of each month.
6. What was average Sales per Transaction for each month.
7. What was average Sales per Day for each month.
8. What was Total profit for Product Key 600 for each month.
9. What was total Sales for Product 214 and 217.
10. What was Total Cost of Product 475 with Black Color.
11. What was Total Profit of Year 2002 for each month.
12. Create a report to show Total Sales per year with break down by Gender. Modify the above report to include Total Sales made to those whose Occupation was "Management".
13. Create a report to show Total Profit earned over the years for each Product Sub Category. Modify the report to show what Profit was earned from customers who were University Graduate. Modify the report to show what Profit was earned from University Graduates who were married.
14. Create a report to show Total Sales for each month with Slicer for Years.

## **Exercise-Multiple Data Tables**

Go to Power BI – Get data – Excel file – File name - Exercise-Multiple Data Tables in Power Query– Select all tables – Transform data – Power query open

In Sales query – change sales data type as ‘date’

Close and Apply

In Power BI – Data includes: Calendar, Customer Name, Exchange Rate, Mini List Price, Products, and Sales

In DAX – Mostly three things use: Simple table, Matrix (for three dimensions of data), Slicer (to filter data)

$\Sigma$  symbol means – numerical number

No symbol means – text

Calendar means – date

Calculator sign - measure

Change data type of year in sales data – from number into text (as it does not require any mathematical calculation)

Click Table – Increase font size – drag and drop “years” column from sales data

Go to visuals:

- Values (change font size to 15)
- Column header (change font size to 15 and bold)

Drag and drop “Sum of sales amount” and “Product Cost” columns from sales data

### **To Create Measure**

Hover on the data – More options – New measure

Or Go to Home – New Measure

## **For Total Sales, Total Cost and Total Profit Margin - According to Year**

**To Create Sales Measure** - Hover on the Sales data – More options – New measure

Total Sales = Sum(Sales[SalesAmount]) – Enter

Drag and drop “Total Sales” measure from sales data

**To Create Cost Measure** - Hover on the Sales data – More options – New measure

Total Cost = Sum(Sales[ProductCost]) – Enter

Drag and drop “Total Cost” measure from sales data

**To Create Profit Measure** - Hover on the Sales data – More options – New measure

Total Profit = Sum(Sales[SalesAmount]) - Sum(Sales[ProductCost]) – Enter => But it is not a best way

Another Way: Total Profit = [Total Sales] – [Total Cost] – Enter

Drag and drop “Total Profit” measure from sales data

Remove “Sum of sales amount” and “Product Cost” columns from table

**To Create Profit Margin Ratio Measure** - Hover on the Sales data – More options – New measure

Profit Margin = Sum(Sales[SalesAmount]) + Sum(Sales[ProductCost]) / Sum(Sales[SalesAmount]) – Enter => But it a longer version (not best)

Another Way: Profit Margin = [Total Profit] / [Total Sales] – Enter

Drag and drop “Profit Margin” measure from sales data

Select Profit Margin – go to measure tools – change data type to “%”

## **For Total Sales, Total Cost and Total Profit Margin - According to Customer Key**

Create another table – drag and drop customer key from sales data

To apply same formatting as previous table – select first table and go to format painter

Drag and drop “Total Sales”, “Total Cost”, “Total Profit”, “Profit Margin” measure in table

**OR** Drag and drop “Year” column above customer key, so it includes all data according to year and customer key as well

Create New Page – Create Matrix – Drag and drop “Year” in columns and “Customer Key” in Rows and “Total Sales”, “Total Cost”, “Total Profit” and “Profit Margin” measure in Values

### **Slicer**

Note: To add slicer make sure no value is selected

On Page1 – Select Slicer – Drag and drop “Years” from sales data in it.

Go to Format setting – Styles – Tiles

Can also change font size

### **How many number of lines added to made up total cost**

**To Create Transaction Measure** - Hover on the Sales data – More options – New measure

Total Transaction = COUNTROWS(Sales) – Enter

Drag and drop “Total Transaction” measure from sales data

### **For how many days customer do a transaction – total days (means count unique values)**

**To Create Days Measure** - Hover on the Sales data – More options – New measure

Number of Days = DISTINCTCOUNT(Sales(OrderDate)) – Enter

Drag and drop “Number of Days” measure from sales data

## CLASS # 6

### Create Measures from Multiple Table

There are two kinds of data sets:

#### Data set that is dependent on another dataset

Ex: Payroll department dependent on dataset. The link between these two departments is employee id.

The purpose of this is to connect two separate datasets and grab information from two separate tables and make the record more minimum.

To reduce data redundancy.

In Hr department – employee id can never duplicate (means it is a **unique value**)

#### Data set that is independent on another dataset

Ex: Hr department is independent on payroll department.

To connect(link) two datasets – first need to find out that what is the **common key** between two datasets. That common key should be **primary key** in one table and **foreign key** in another table.

Primary key – data that has unique value

Foreign key – contains repeated key(data)

### To Create Relationships between Tables

1. Identify those fields that can be used to connect table
2. What is the data type of these fields? Data type ideally should be same.

### Types of Relationships:

One to Many Relationship means – one primary key while multiple foreign keys.

One to Many Relationship

Many to Many Relationship etc

## **Exercise-Multiple Data Tables**

Duplicate Page 1 – by right click on page and select duplicate

Remove 2<sup>nd</sup> table from the page

Table 1 has: Year, Total Sales, Total Cost, Total Profit, and Profit Margin in %

### **To Show Sales of a Specific Product**

Go to relationships area – delete relationship between product and sales and sales and customers

### **To create relationship between Products and Sales table**

Product Key is same in between sales and product table. Drag product key from the product table and drop at product key at the sales table

Hover mouse of relationship line it will highlight the relation part. (One to many relationship)

### **To create relationship between Customer and Sales table**

Customer Key is same in between customer and product table. Drag customer key from the customer table and drop at customer key at the sales table

Hover mouse of relationship line it will highlight the relation part. (One to many relationship)

**Note:** Sales table now can work as a link table to create relation between customer and product table, as there is no common key between them.

## **CALCULATE () - Your new friend**

**CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])**

What was **Total Revenue** of **Product X** sold to **Professors** in **Y-2020**.

<b>Expression</b>	Total Revenue
<b>Filter1</b>	Product X
<b>Filter 2</b>	Professors
<b>Filter 3</b>	Y-2020

**What was the Total profit for Product key 600 for each month.**

**To Create Sales of Product key 600 Measure** - Hover on the Sales data – More options – New measure

Sales P-600 = CALCULATE([Total Sales], Products[Product Key]=600) – Enter

Drag and drop “Sales P-600” measure from sales data into table1

**To Calculate Sales of two different products**

**To Create Sales of Product key 600 and 528 Measure** - Hover on the Sales data – More options – New measure

Sales P-600 P-528 = CALCULATE([Total Sales], Products[Product Key]=600) +  
CALCULATE([Total Sales], Products[Product Key]=528) – Enter

Drag and drop “Sales P-600 P-515” measure from sales data into table1

**Sales of Product 475 in Black Color**

Here 475 = Numerical value and Black = Strong / text value

**To create measure** - Hover on the Sales data – More options – New measure

Sales-P-600-Black = CALCULATE([Total Sales], Products[Product Key]=475,  
Products[Color]= “Black”) – Enter

Drag and drop “Sales P-600 Black” measure from sales data into table1

**Sales of Product 475 in Black Color sold to customers of University graduates  
(Graduate Degree)**

Note: In Customers – English Education column – data of degrees available

**To create measure** - Hover on the Sales data – More options – New measure

Sales-P-475-Black-GD= CALCULATE([Total Sales], Products[Product Key]=475,  
Products[Color]= “Black”, Customer[English Education]=“Graduate Degree”) – Enter

Drag and drop “Sales-P-475-Black-GD” measure from sales data into table1

No answer will come – Remove the measure

## Total Sales of University graduates (Graduate Degree)

To create measure - Hover on the Sales data – More options – New measure

Sales-GD= CALCULATE([Total Sales], Customer[English Education] = "Graduate Degree") – Enter

Drag and drop “Sales- GD” measure from sales data into table1

## Create a table that contains list of the product, total sales and English education

Create a new page – Click to add table

From Product Table – drag and drop the Product Id

From Sales Table – drag and drop the Total Sales

From Sales Table – drag and drop the Year (Here we can replace Year with English education)

## Disconnected Tables

### Disconnected Tables

- A table that has no relationship with any other table in the dataset is called a Disconnected Table.

1. Create a Sales report and shows the following :
  - A. Annual Sales in Dollars by Product Name .
  - B. Annual Sales in Euros by Product Name.
  - C. Note:- Report should be fully dynamic. Values should be changed automatically when a new exchange rate is selected.
2. Create a dynamic data model that should show Total Sales by Product Category for a Threshold List Price selected by the user.

Table that cannot be linked with any other table is known as **Disconnected Table**

Duplicate the previous duplicate page 1

Delete all the data from the page

### **Show sales in different exchange rates**

Select Year and Total Sales from Sales Table – in the main table

Add slicer in the page – drag and drop exchange rates in the slicer

Go to visual – Style – Select Tiles

Hover on the Exchange rate table – More options – New measure:

Selected Rate = SELECTEDVALUE(Exch Rates [USD Per Euro]) – Enter

Drag and drop selected rate from Exchange rate table – into the main table

Hover on the Sales table – More options – New measure:

Total Sales in Euros = [Total Sales] / [Selected Rates] – Enter

Drag and drop Total Sales in Euros from sales table – into the main table

Remove Selected rate from the table (as it is already available in slicer)

### **Answer of Question-02**

#### **Total Sales over Product Category**

Click on Matrix to add on Page

Drag and drop “year” in rows and “total sales in euros” in value from sales table

Add “English Product Category name” in columns from the products table

Add Threshold Price list in the slicer.

## CLASS # 7

### Previous Data File

#### Answer of Question-02

Add a Slicer – Drag and drop the minimum price list file into the slicer

Go to Visual – Settings – Style – Tiles

Go to Visual – Increase font size to 13 and make it bold

Select a table – drag and drop “English Product Category name” in table

Go to Visual – Increase value font size to 14 and Column header font size to 16 make it bold

Hover on the Min Price list table – More options – New measure:

Selected Price = SELECTEDVALUE(MinListPrice [Min List Price]) – Enter

Drag and drop “selected price” from Min Price list table – into the matrix table

Hover on the Min Price list table – More options – New measure:

Total Sales Above Listed Price = CALCULATE([Total Sales], Filter(Products, Products[List Price]>=[Selected Price])) – Enter

Drag and drop “Total Sales Above Listed Price” from Min Price list table – into the main table

### Time Intelligence

## Time Intelligence

Sometimes you need to perform calculations that answer the questions like the following:

- How is business performing relative to same period last year.
- What was our Year to Date (YTD) as of June 1<sup>st</sup>?
- What was our best quarter over the past two years?

- Everything in Time Intelligence requires that you have a separate Calendar Table.
- Contains at least one column of “date” data type.
- Contains exactly one row per day.
- Contains completely consecutive dates, no gap.
- Must be related to all of your data tables.

### Properties of Standard Calendar:

- ✓ Feb. has 28 days (29 in leap year).
- ✓ All other months have 30 or 31 days.
- ✓ Quarter consist of three consecutive months.
- ✓ Year has 365 days (366 in leap year).
- ✓ A give month this year might have more or less Saturdays in it in the same month last year.

### Custom Calendar

- ✓ Using working days of the month instead of calendar days.
- ✓ Non conventional months in quarter [like first quarter starting from July].

**Note:** In Time Intelligence Function – need to summarize data (means drill down data in a single visual). Like Year to Month, Month to Days etc.

For Time Intelligence Function – Calendar is required known as “Calendar Table”. It should cover every single date that is present in dataset.

### **Create Calendar Table from 2001 to 2004**

Go to Home tab – New table: My Calendar = CALENDAR(“01/01/2001”, “31/12/2004”) – Enter

Select the table – go to table tool – Change data type to “Date” only

Go to Home tab – Add a new column: Year = Year(My Calendar[Date]) – Enter

Go to Home tab – Add a new column: Quater = QUATER(My Calendar[Date]) – Enter

Go to Home tab – Add a new column: Month Number = Month(My Calendar[Date]) – Enter

Go to Home tab – Add a new column: Month Name = Format(My Calendar[Date], “mmmm”) – Enter

Go to Home tab – Add a new column: Day Number of the week = WEEKDAY(My Calendar[Date], 1) – Enter {Note: 1 means 1=Sun and so on..}

Go to Home tab – Add a new column: Day Name = Format(My Calendar[Date], “dddd”) – Enter

Delete Old Calendar Table i.e. Calendar New from the data model (as there are two tables now). Click the table – Delete from the data model

### **To mark My Calendar table – as Calendar Table in Power BI**

Select My Calendar table – go into Tree model – Select Mark as date table

In Date column – Select Date – Ok

**Note:** To mark a table as date type there is one requirement – there should be at least one date field that contain continuous and no-duplicate date. (means it should have unique value with no gaps)

It activates time intelligence functions

## **Total Sales Over the year by the months**

Select matrix – Drag and drop Total Sales from sales table in values

From My Calendar table – drag and drop year into the column

From My Calendar table – drag and drop month name into the rows

In Visual pane – Increase font size of text to 13 and font size of column header to 14 and font size of rows to 14

Note: The result came has same values all over the table. This is due to we involve two tables Sales table and My Calendar table and there is no relation between them.

## **To Create Relationship between tables**

Go to 3<sup>rd</sup> icon (model view) – Drag Date from My Calendar table and drop it on Order date column of Sales table to create relation between them

Now the result will come different.

## **To Sort Month / Day Name**

Select Month Name column – go to column tools – Sort column (by Month Number)

Select Day Name column – go to column tools – Sort column (by Day Number of the week)

In the main page table – go to more options – Sort by – Ascending order

## **Module-3-Hierarchy**

To drill down data – Use Hierarchy (steps use to filter down the information)

**Hierarchy of Calendar:** Year – Half – Quarter – Month – Week – Day – Hour – Minute – Second – Millisecond

## **To Create Year Hierarchy**

Select a Slicer – Select Year – From Visual – Change Style – Tiles

Select Year column in My Calendar table – Create hierarchy

Rename hierarchy – Data Drill Down

Select Quarter column in My Calendar table – Add to hierarchy – Select Data Drill Down

Select Month Name column in My Calendar table – Add to hierarchy – Select Data Drill Down

Select Date column in My Calendar table – Add to hierarchy – Select Data Drill Down

Create a New Page – Select Stacked Column Chart

Drag and drop Total Sales from the sales table

Select Data Drill Down hierarchy

Go to Visual – X-axis and Y-axis – Values font size = 13 + On Data labels + Values font size = 14

### **Drill Down Option – for specific year**

Select the drill down option on the visual – if single click on 2003 (it will automatically drill down data according to quarters) + if further click on quarter 4 (it will further drill down data according to months) + if further click on December month (it will further drill down data into date)

To go back to previous steps – Click on drill up

### **Drill Down Option – for all years**

If we click on double arrow (that is next to drill down) – it will drill down the data into quarters but according to all years (not a specific year) + if further click on it (it will drill down the data into month but according to all months)

To go back to previous steps – Click on drill up

## Time Intelligence Functions

<b>DATESYTD()</b>	Calculates Year To Date values for a numerical field <code>Calculate([Total Sales], DATESYTD(Calendar[Date]), "12/31/2004")</code>
<b>DATESQTD()</b>	Calculates Quarter To Date values for a numerical field <code>Calculate([Total Sales], DATESQTD(Calendar[Date]))</code>
<b>DATESMTD()</b>	Calculates Month To Date values for a numerical field <code>Calculate([Total Sales], DATESMTD(Calendar[Date]))</code>
<b>TOTALYTD()</b>	A replacement of DATESYTD() function <code>TOTALYTD([Total Sales], Calendar([Date]))</code>
<b>FIRSTDATE()</b>	Returns the first date in the current context for the specified column of dates. <code>FIRSTDATE('Calendar'[Date])</code>
<b>LASTDATE()</b>	Returns the last date in the current context for the specified column of dates. <code>LASTDATE('Calendar'[Date])</code>
<b>CLOSINGBALANCEMONTH()</b>	Evaluates the <b>expression</b> at the last date of the month in the current context. <code>CLOSINGBALANCEMONTH([Total Sales], Calendar[Date])</code>
<b>DATEADD()</b>	Shifts either forward or backward in time by the specified number of intervals <code>CALCULATE([Total Sales], DATEADD('Calendar'[Date], -1, YEAR))</code>
<b>DIVIDE()</b>	Performs division and returns alternate result or BLANK() on division by 0. <code>=Divide(([Total Sales]-[Total Sales 1 Year Back]), [Total Sales 1 Year Back])</code>
<b>SAMEPERIODLASTYEAR()</b>	Dates shifted one year back in time from the dates in the specified <b>dates</b> column <code>CALCULATE([Total Sales], SAMEPERIODLASTYEAR('Calendar'[Date]))</code>

## Time Intelligence - Practice

- ✓ Create a simple pivot table that shows Total Sales per weekday over the months. Sort months and weekdays in order.
- ✓ Create a dynamic report that shows Total Sales as well as Year-To-Date Sales.
- ✓ Create a dynamic report that shows Total Sales and Total Sales Quarter To Date for a selected year.
- ✓ Create a dynamic report that shows Total Sales and Total Sales Month To Date for each week of the year.
- ✓ Create a dynamic report that shows Total Sales and Total Sales YTD for each day number of the year.
- ✓ Create a dynamic report that shows what was first and last date of each week number of the year.
- ✓ Create a dynamic report that shows Total Sales for the month as well as Total Sales of the last date of the respective month.
- ✓ Create a dynamic report that shows Total Sales for each quarter of Selected Year, Year Before the Selected Year and Year after the Selected Year.
- ✓ Create a dynamic report that shows Per Month Total Sales for the selected Year, Total Sales for Previous year of the selected year and Growth Rate.
- ✓ Create a dynamic report that shows Total Sales Per Week number of the year, Total Sales for the Same Period Last Year and Growth Rate.

### **Sales in terms of month**

On a New Page – Create a matrix – Drag and drop month name in column from My Calendar table

Drag and drop Total Sales from Sales table in column

Note: If want to see Sales according to Days – drag and drop day name in column from My Calendar table

### **Sales in terms of Year to date**

Click on Sales – New measure: Total Sales YTD = TOTALYTD([Total Sales], My Calendar[Date]) – Enter

Drag and drop Total Sales YTD from Sales table in column

### **Sales in terms of Quarter to date**

Click on Sales – New measure: Total Sales QTD = TOTALQTD([Total Sales], My Calendar[Date]) – Enter

Drag and drop Total Sales QTD from Sales table in column

### **Sales of Last Date of the Month**

Click on Sales – New measure: Sales on the Last Date = CLOSINGBALANCEMONTH([Total Sales], My Calendar[Date]) – Enter

Drag and drop Sales on the Last Date from Sales table in column

### **To verify the sales on last date of month above values**

Create a new page – Add year from sales table in column + Add month name from sales table in rows + Add date from sales table in rows + Add total sales from sales table in values

### **Total sales of year before selected year**

Duplicate the previous page – remove all other columns from table except month name and total sales

Click on Sales – New measure: Total Sales Last Year = CALCULATE([Total Sales], Dateadd('My Calendar'[Date], -1, YEAR)) – Enter {Note: means minus one year}

Drag and drop Total Sales Last Year from Sales table in column

### **Total sales of year after selected year**

Click on Sales – New measure: Total Sales Coming Year = CALCULATE([Total Sales], Dateadd('My Calendar'[Date], 1, YEAR)) – Enter {Note: means add one year}

Drag and drop Total Sales Coming Year from Sales table in matrix column

Arrange columns as: Total Sales of Last Year, Total Sales, Total Sales of Current Year

### **Growth rate of year before selected year**

Growth = [Total Sales] – [Total Sales of Last Year] / [Total Sales of Last Year] – Enter

Select growth column – Measure tools – Select %

## CLASS # 8

### Conditional Measures

## Conditional Measures

**SWITCH()** Evaluates an expression against a list of values and returns one of multiple possible result expressions.

=Switch(Customers[EnglishOccupation], "Clerical", "Cash Offer", "Management", "Purchase Discount", "Manual", "Trade In", "Skilled Manual", "Gift Voucher", "Gift Hamper")

**if() And Blank()** Works same as Excel

=If([Total Sales]>=100000,[Total Sales]\*0.02,Blank())

- ✓ Management has decided to offer different incentives to customers based on their English Occupation.

Clerical	Cash Offer
Management	Purchase Discount
Manual	Trade in
Skilled Manual	Gift Voucher
Others	Gift Hamper

Create a calculated column in Customers table, name it as "Incentive" and add this as field to report.

- ✓ To encourage its sales force, Management has decided to give 2% of the sales value as Bonus if weekly sales is more than \$100,000. Prepare a report that shows Bonus amount for each week number over the years. Where Bonus is zero, the cell should be Blank().

There are two parts of conditions: outcome depends on whether the condition is True or False

Nested if – can also use for multiple conditions

To test multiple conditions – use **Switch** function (works on specific column). It has three parameters: Name of the column, Value want to apply as test, what is the outcome if condition is true

### Previous data

### Use of Switch Function

Go to table view – Customer table – English Occupation column

Go to column tools – Select New column: Incentives =  
SWITCH(Customers[EnglishOccupation], "Clerical", "Cash Offer", "Management", "Purchase Discount", "Manual", "Trade in", "Skilled Manual", "Gift Voucher", "Gift Hamper") – Enter {Note: Here Gift hamper is for the else part}

**If Sales is more than 100 thousand dollars give 2% discount + where bonus is 0 cell should be blank**

Add a new page – select a table

Go to table view – calendar table – add a new column: week number = WEEKNUM('My Calendar[Date]', 1)

Now go back to visual pane – select year in the table + select week number in table

From the sales table – drag and drop total sales in the table

Go to visual – setting – increase font size of values and column headers to 13

On Sales – right click – new measure: Bonus = (if [Total Sales] > 100000, [Total Sales]\*0.02, 0) – Enter

Drag and drop Bonus column in the table

### Edit Sales measure – for where bonus is 0 cell should be blank

Bonus = (if [Total Sales] > 100000, [Total Sales]\*0.02, " ") – Enter {Note: the double quotation makes the 0 value blank, but in the form of string due to which when we drag and drop deduction column in the table it will cause an error}

To resolve the problem, instead of write double quotation write blank:

Bonus = (if [Total Sales] > 100000, [Total Sales]\*0.02, Blank())

On Sales – right click – new measure: Deduction = [Bonus]\*0.05 – Enter

Drag and drop deduction in the table

---

## Iterator Functions

### Iterator Functions

These are functions that work through a table, row by row to complete the evaluation in the current filter context.

**SUMX()** SUMX() will iterate through a table specified in the first parameter, one row at a time, and complete a calculation specified in the second parameter. Once it has done this for every row in the specified table in current filter context, it then adds up the total of all the row-by-row calculations to get the total. It is this total that is returned as the result.

=SUMX(table,Expression)  
=[Total Sales] / DISTINCTCOUNT(Sales[OrderDate])  
=SUMX(VALUES(Products[ModelName]),[Average Sales Per Day])

**MAXX()** Evaluates an expression for each row of a table and returns the largest numeric value.

=MAXX(table,Expression)  
=MAXX(Values(Customers[EnglishOccupation]), [Total Sales])

**RANKX()** Returns the ranking of a number in a list of numbers for each row in the *table* argument.

=RANKX(Values(Customers[Full Name]),[Total Sales]) [each customer considered individually]  
=RANKX(ALL(Customers[Full Name]),[Total Sales])

Functions that run through loop – means run row by row (evaluate each row) – known as **Iterative Function**. Mostly these functions end at ‘x’.

## Iterator Functions

### TOPN()

Returns the top N rows of the specified table.

```
=SUMX(table,Expression)  
=DISTINCTCOUNT(Sales[ProductKey])  
=CALCULATE([Total Sales],TOPN(5, Products,[Total Sales]))  
=DIVIDE([Sales of Top 5 Products],[Total Sales])
```

- ✓ Create a report that shows per year figures of:
  - Average Sales per day made to each English occupation.
  - Add a new column in your report to show correct Totals for the above requirement.
- ✓ Create a report that shows ranking of Total Sales per Customer [Full Name].
- ✓ Create a dynamic report that shows the following for a selected year:
  - ✓ Per Month Total Sales.
  - ✓ Total number of products sold for each month's Total Sales.
  - ✓ Sale of Top 5 Products of Per month Total Sales.
  - ✓ Percentage of Top 5 Products Sales compared to Total Sales of each month.

### Average Sales per day

Add a new page – select matrix – add year in rows and add total sales in values

Go to settings – visuals – increase row headers size to 15

Add English occupation from customers table in matrix table rows. Right click on matrix table and expand it.

Click on Sales – New measure: Avg Daily Sales = [Total Sales] / [Number of Days] – Enter

Add Avg Daily Sales in matrix values

Note: In Avg Daily Sales column the first row number shows the average of year 2001, not the total of the filtered context value. To get the average of all the filtered context values use iterated function.

### Iteration on Avg Daily Sales – Use Function SUMX

Go to Sales column – new measure: Avg Add up =  
Sum(Values(Customers[EnglishOccupation]), [Avg Daily Sales]) – Enter

Drag and drop Avg Add up on the values of matrix table

## **Ranking of Total Sales per Customer – Use Function RANKX**

Go to table view – customers table – new column: Full Name = Customers[First Name] & “ “ & Customers[Middle Name] & “ “ & Customers[Last Name] – Enter

Remove year from the previous matrix table – copy paste the year slicer from the previous page in the new page and sync it.

Create a new matrix table – add full name in rows + add total sales in values

Go to Sales column – new measure: Sales Ranking =  
RANKX(All(Customers[FullName]), [Total Sales],,, DESC) – Enter {Note: DESC means order of the ranking is in descending order}

Drag and drop Sales ranking in the table. From more options – Sort it by Sales ranking – by descending order

## **Extraction of Top values from data – use Function TOPN**

### **Total Number of Product sold for each month**

Create a new matrix – add month name from calendar column on rows + add total sales from sales table in values

Go to Sales column – new measure: No of Products =  
DISTINCTCOUNT(Sales[ProductKey]) – Enter {Note: press enter by selecting the year 2003 from column}

Drag and drop No of Products in the table

### **Sales of Top 5 Products per Month Total Sales – Extraction of Top values from data – use Function TOPN**

Go to Sales column – new measure: Sales-Top-5-Products = CALCULATE([Total Sales], TOPN(5, Products, [Total Sales])) – Enter {Note: 5 means top5 values, products – column name}

Drag and drop Sales-Top-5-Products in the table

Note: There is issue in ranking – due to dataset problem - 1:43: 00

## **CLASS # 9**

### **Ex-1-Power\_BI.accdb**

Click on get data – All – Access Database – Connect - Ex-1-Power\_BI.accdb – Select all tables – Transform data

In Calendar table – change data type of date into ‘date’ + convert data type of year into ‘text’

In Manufacturer table – change data type of manufacturer id into ‘text’

In Sales and Sentiments table – change data type of date into ‘date’

Rest every other data type is fine

Close and Apply

### **Sort columns**

Go to table view – calendar table – table tools – mark as date table – Date column (date) – ok

Select month name column – column tools – sort by (month no)

### **Create relationship**

Go to relationship model view – create relationship between calendar and sales table – drag date from calendar table and drop it on date of sales table

### **Create measures**

Right click on sales – new measure: Total Revenue = Sum(Sales[Revenue]) – Enter

In measure tools – select decimal value = 0

Right click on sales – new measure: Total Units = Sum(Sales[Units]) – Enter

### **Create Hierarchy**

In products table – Right click on category – click on more options – create hierarchy

Right click on segment column – add to hierarchy – category hierarchy

Right click on product id – add to hierarchy – category hierarchy

**Note:** To drill down a specific element (click on that element), while to drill down the whole dataset (click on the double arrows)

## Summary of data

### Create stack chart

Click on stacked column chart – add total units from sales table in visual (y-axis) + add hierarchy from products table in visual (x-axis)

Go to visual – setting – increase size of x and y-axis to 13 and bold + turn on data labels + increase its value size to 12 and bold

Turn on drill down option by clicking on it

Click on urban category bar (it will expand to next level) + Click on extreme segment (it will further drill down into products) + More option – sort by (descending order) + Click on drill up to go back to first level.

Click on double arrow (it will drill down segment according to all category)

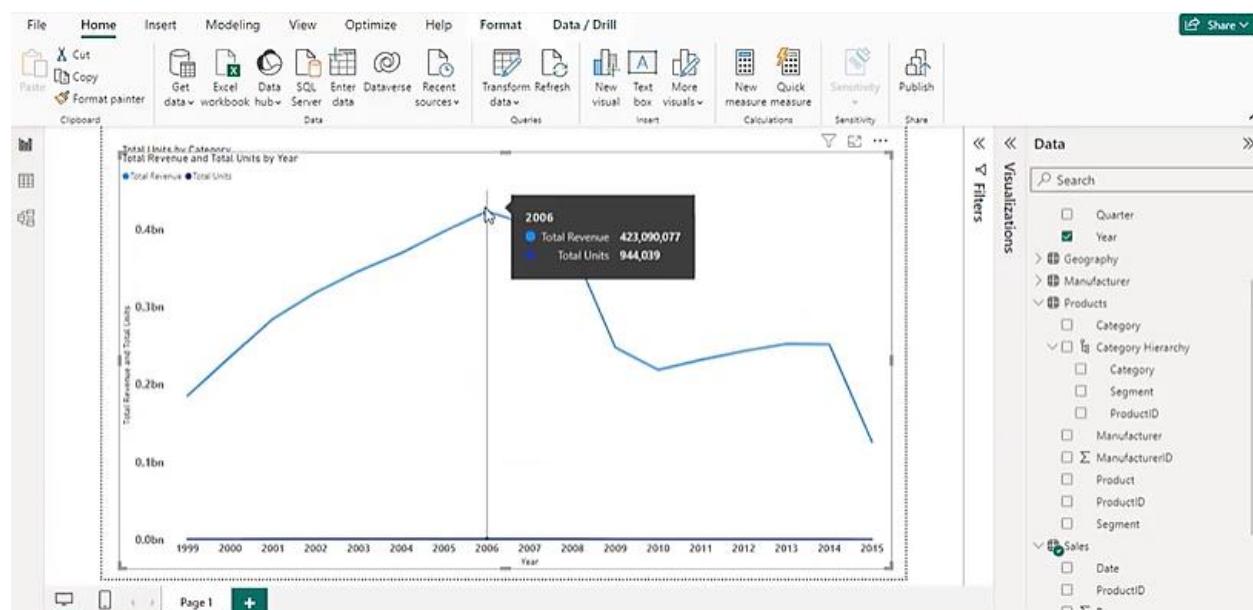
### Create line chart

Click on line chart (increase its size) – add year from the calendar table (x-axis) + add total revenue and total units from the sales table (y-axis)

Go to visual – settings – increase size of x-axis to 12 + increase size of y-axis to 12 and bold

Click on more options – sort axis – year + Click on more options – sort axis – sort by ascending order

**Note:** In below figure we can see that total revenue scale and total units scale has different scales. That's why total units show a straight line. So for different scale data do not use line chart – instead use **line and stacked column chart**.



## Create line and stacked column chart.

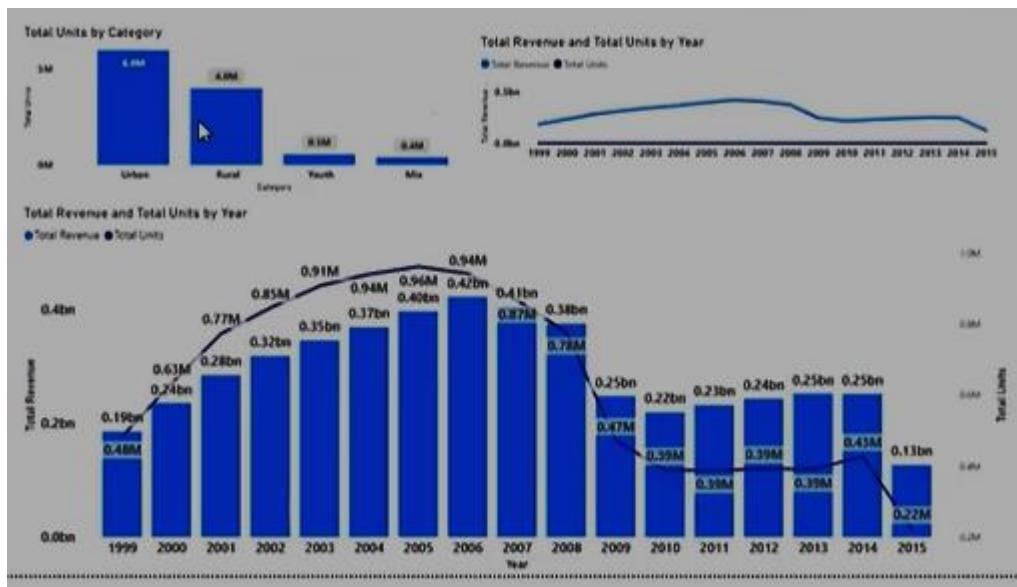
Click on line chart (increase its size) – add year from the calendar table (x-axis) + add total revenue from the sales table into y-axis columns + add total units from the sales table into y-axis line

**Note:** There is billions scale on left side for total revenue and millions scale on right side for total units.

Click on more options – sort axis – year + Click on more options – sort axis – sort by ascending order

Go to visual – settings – increase size of x-axis to 12 + increase size of y-axis to 12 and bold + turn on data labels + increase size of value to 12 and bold

**Note:** Drag and drop category into column legends it will divide columns in terms of category which is total revenue. But it cannot divide unit line into categories which is total units. So, it is not appropriate to add that. But it depends on the user. Best practice is to break down visual into two visuals.



**Same category visuals:** Gauge, Single line card, Multi-row card, KPI – used to show progress for a certain product.

## Single line card, Multi-row card

Click on card – drag and drop total revenue in it.

Add another card on page – drag and drop total units in it.

Note: To show both information in one image, use multi-row visual – drag and drop total revenue and total units in it (resize it)

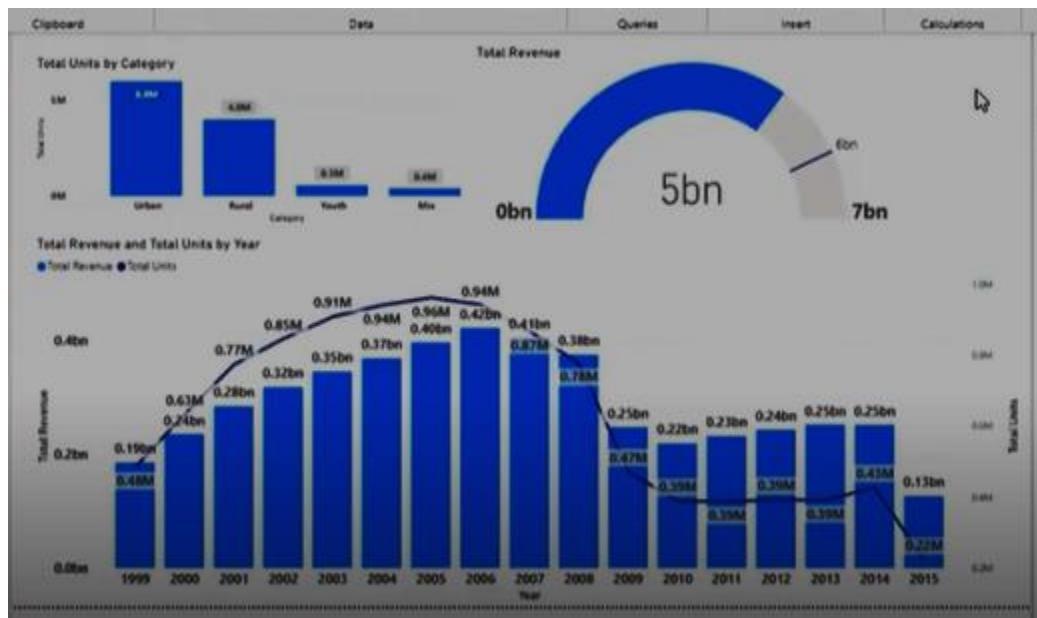
## Gauge

Click on gauge – drag and drop total revenue in it.

Go to visual – setting – data labels – increase value size to 19

**Note:** In gauge visual – by default it double the result for total. Like if revenue is 5bn then total will be 10bn

Go to visual – setting – gauge axis: Target = 6000000000 (estimated value), Max = 7000000000(budget number)



## **CLASS # 10**

### **Waterfall Chart**

To show for running total

#### **Example**

Total number of responses this year compare to last year

Total sales of current year compare to total sales of last year + Growth in terms of dollar value and in terms of percentage – Use table or matrix

To show growth in the table in visual form – waterfall chart

Select a table – Add Month Name from Calendar data

For visual properties – Go to values and text – increase its size = 17 and bold

Select Slicer (new) – Drag and drop Year from Calendar data

For visual properties (Slicer Settings: Shape – Snipped)

Select 2006 tile

Drag and drop – Total Revenue from Sales data in the table

#### **Sale of Last Year**

Right click on Sales table – new measure: Last Year Revenue = CALCULATE[Total Revenue], SAMEPERIODLASTYEAR('CALENDAR'[Date]) – Enter

Drag and drop – Last Year Revenue from Sales data in the table

#### **Growth in Terms of Dollar**

Right click on Sales table – new measure: Growth = [Total Revenue] – [Last Year Revenue] – Enter

Drag and drop – Growth from Sales data in the table

#### **Growth in Percentage**

Right click on Sales table – new measure: Growth % = [Growth] – [Last Year Revenue] – Enter

Drag and drop – Growth % from Sales data in the table

Go to measure tool – change it into %

**Note:** If we select 1999 in slicer so there is no previous year data in this. Means we cannot calculate growth in this as base number is not there. How to solve this?

**Ans:** Use Divide Function

### Growth in Percentage

Right click on Sales table – new measure: Growth % = DIVIDE([Growth], [Last Year Revenue]) – Enter

Drag and drop – Growth % from Sales data in the table

Go to measure tool – change it into %

Select waterfall chart

Drag and drop Growth from sales data in y-axis

Drag and drop Month Name from calendar data in y-axis

Go to visual settings – turn on data labels and increase size of value to 12 and bold and change its color to red

Also increase size of x and y-axis to 12 and bold

### KPI Visual

It shows current progress compare to the target

Target displayed in the background and actual value display above it

Target = Revenue earned last year

### What is performance in current year compare to last year

Add new page – copy paste slicer visual from previous page – Sync

Select KPI Visual – drag and drop Last year revenue in target

Drag and drop total revenue in values

Drag and drop Month Name in Trend axis

**Note:** Red means target does not achieved (as shown in year 2006) while Green means target achieved (as shown in year 2000)

## **Artificial Intelligence Visuals**

Go to insert – there is a portion known as “AI Visuals”

**Decomposition Tree** – Breakdown a specific number into components of the user choice

Add a new page

Select Decomposition Tree – drag and drop Total Revenue from sales data into Analyze

Go to visual settings – Increase font size of value to 12 and bold along with increase font size of category labels to 13 and bold

Drag and drop Month Name from calendar data in Explain by

Right click on total revenue – by month name

---

Drag and drop Segments from products data in Explain by

Right click on April – by month name

---

Drag and drop Categories from products data in Explain by

Right click on convenience – by category

---

Drag and drop Manufacturer from products data in Explain by

Right click on Urban – by manufacturer

**Note:** If we want to see Leo manufacturer data – click on it visual automatically highlight its visual.

## **Q&A Visual**

Select Q&A visual – Write question: Total units sold by product category – pin the visual – to change it into different chart like into area chart (just select the area chart – it will automatically convert)

## **Narrative Visual**

Go to first page – select narrative visual (it will automatically generate text)

Select the text and increase its font size to 14

## **Key Influencers Visual**

## To Publish File on PowerBI.com

Log in to Power BI

Go to Power BI Desktop – Save the report – Home – Publish

Select Finance Workspace – Okay

Go to Power BI.com – Go to my workspace – Finance (it contains both data and report)

**Note:** Anything that user change in Power BI Desktop will not automatically change in PowerBI.com and vice versa. Unless sync these two by installing local gate way.

Without that, Delete previous file from Power BI.com and republish changed Power BI Desktop file in PowerBI.com

## To edit file in PowerBI.com

Go to Edit – Add Pie Chart – Drag and drop Total Units from sales data and Segments from product data

Go to Mobile layout – drag and drop pie visual in it (also can resize the visual)

For Page 1 - Suppose drag and drop fig 1, 2 and 4 (not 3)

**Dashboard** is the static and summarized representation of the data that is collected from multiple tables, source

To download report as pdf – Go to File – Export as Pdf

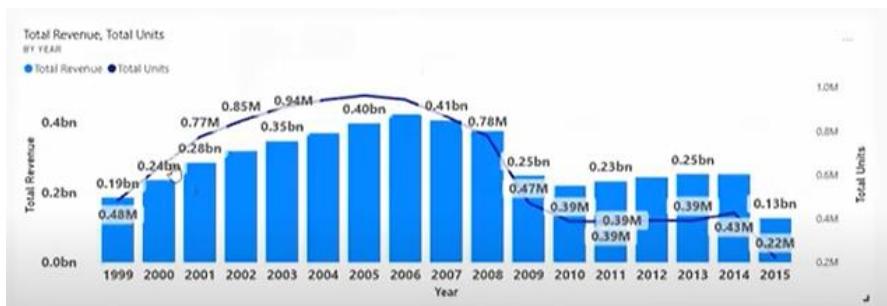
To download report as power point – Go to File – Export to power point

Select visual from page 1 and pin it

Create Dashboard – Dashboard Name (NED Dashboard) – Pin (can adjust its size)

Select visual from page 2 and pin it

Select visual from page 3 and pin it





**For Mobile view** – go to mobile view – unpin image that user didn't want to show on mobile view

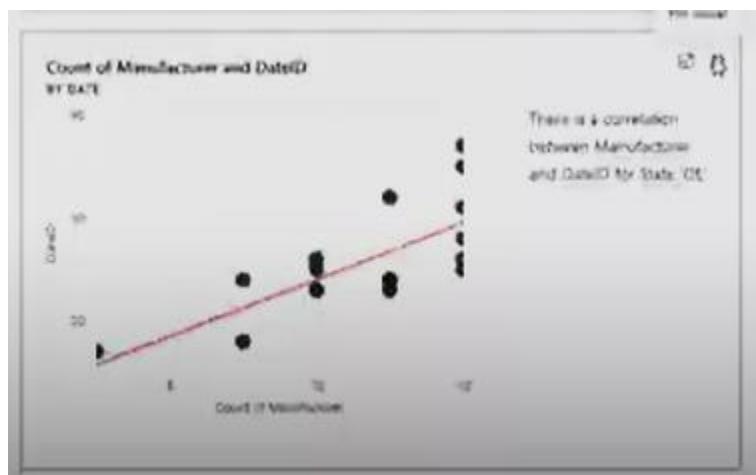
**To add more visuals:** Click add – add a tile

Report can also Share

## Insights by AI

Go to my workspace – click three dots on dataset – click on to get quick insights – view insights

Pin the image to Dashboard



## **CLASS # 11**

### **US\_Accidents\_May19**

Data set related to accidents reports 911

Total number of accidents

Which state has higher number of accidents

Find state that has highest accidents recorded in 2015

Which state has higher number of accidents plus what are the weather conditions plus how much surveyor are accidents

What kind of visual you add

What are your findings

## **Structured Query Language – SQL**

Language we use to interact with database is SQL

### **What is SQL**

**SQL (Structured Query Language)** is a standardized programming language used for managing relational databases and performing various operations on the data.

Initially created in the 1970s, SQL is regularly used by database administrators, as well as by developers writing data integration scripts and data analysts looking to set up and run analytical queries.

The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements -- commonly used SQL statements include select, add, insert, update, delete, create, alter and truncate.

**Database** is a collection of information

**Relational Database** – structure which links data with certain other fields

**Table** is kind of a spreadsheet – used to store data in form of grid (here column is filed and row is record)

**Note:** Need to command on – string, date and time, number, Boolean data types

# Database Terminology

**Database** A database is a collection of information that is organized in such a way that it can be easily accessed, managed and updated.

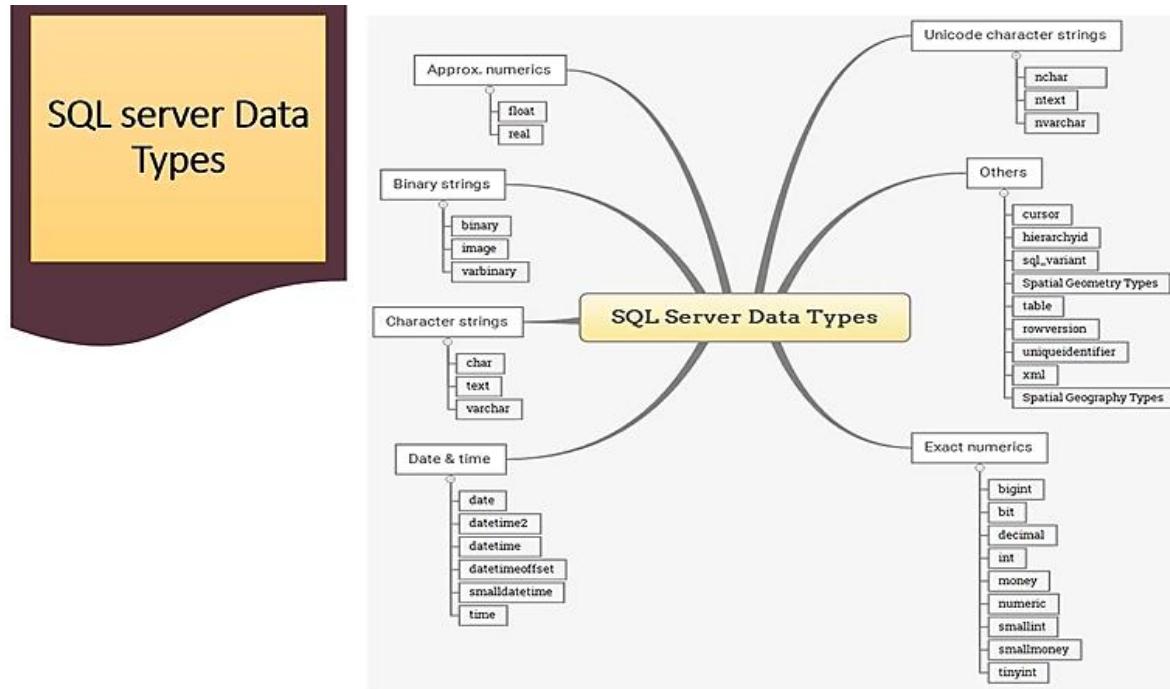
**Table** In computer programming, table is a data structure used to organize information just like it is on a paper.

Each table in Access database contains information about a single entity, such as a person or a product. The data in a table is organized in rows and columns.

**Records and fields** Each row in a table is called a record while a column is called a field.

**Values** At the intersection of a record and fields is called VALUE – the actual data element.

**Schema** A SQL database contains multiple objects such as tables, views, stored procedures, functions, indexes, triggers. **Schema** is a logical collection of database objects. It is a useful mechanism to segregate database objects for different applications, access rights, managing the security administration of databases.



By default two schema are available in MySql Workbook

## MySQL Workbook

To load data in My Sql Workbook – Click on left area blank space – New schema or on the above icons (new schema) – Name (recipies) – Apply

Right click on recipies – Table Data Import – Browse file path – Ingredient Class – Next – Next – Apply

Refresh schema

Click on the name of the schema to see the data

To see the expanded data – go to recipies (then select grid) query will automatically show the data

---

Right click on recipies – Table Data Import – Browse file path – Ingredient– Next – Next – Apply

Refresh schema

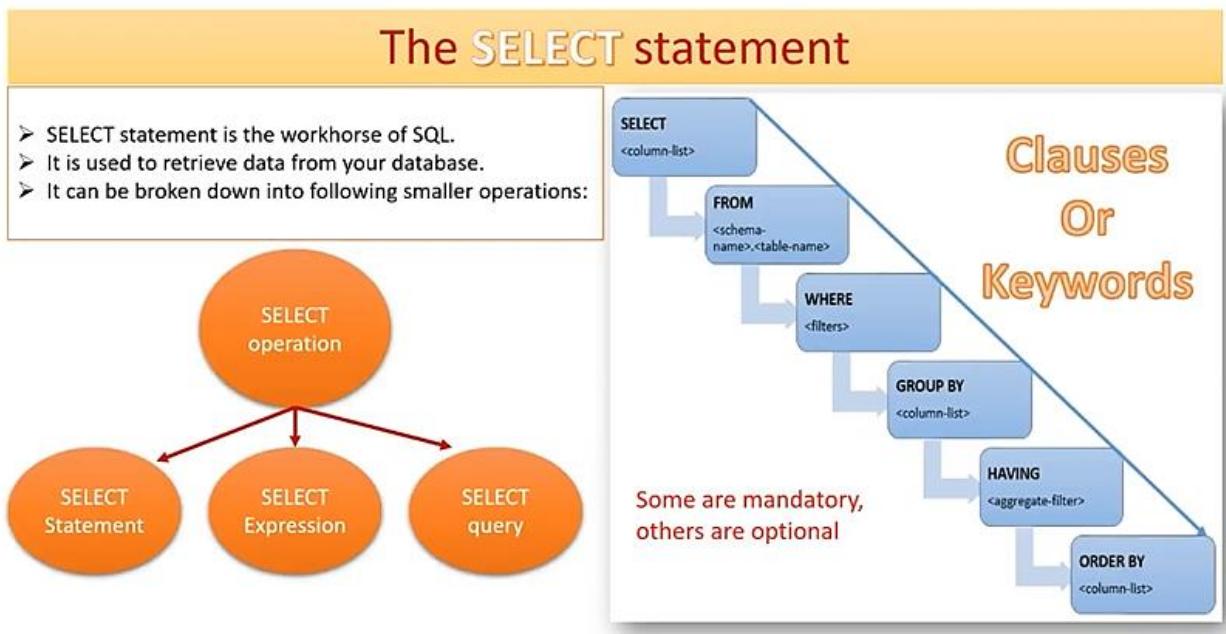
Click on the name of the schema to see the data

---

Load all the files using the above method

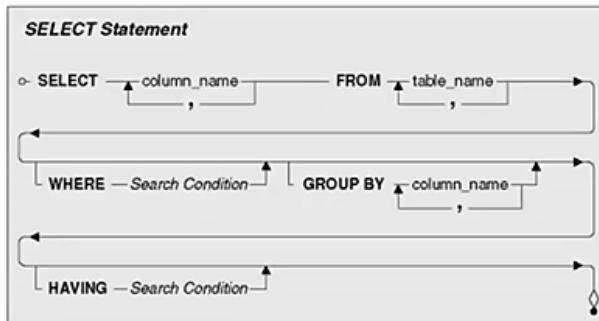
## The Select Statement

Select Statement is used to extract or to find data from a database or a dataset.



Select and From Clause are mandatory others are optional

## Syntax of **SELECT** statement



### Sales Order Database

"I want names and addresses of all our employees"

```
SELECT EmpFirstName, EmpLastName, EmpStreetAddress, EmpCity, EmpState, EmpZipCode  
FROM Employees;
```

Select sales data – select customers table from it

**Query:**

```
USE Sales_ Orders: (here it tells about database)
```

```
SELECT custFirstName, custLastName, custCity, custState, custStreetAddress
```

From Customers – Run

Sales Order Database - A short cut to request all columns

"Show me all the records and fields of Employees table"

```
SELECT *  
FROM Employees;
```

**Query to get everything from customers table:**

```
USE Sales_ Orders:
```

```
SELECT *
```

From Customers – Run

## Eliminating Duplicate Rows

Bowling League Database

"What cities are represented by our bowling league membership?"

```
SELECT BowlerCity  
FROM Bowlers;
```

```
SELECT DISTINCT BowlerCity  
FROM Bowlers;
```

## Sorting Information

### Sales Order Database

"Show me a list of vendor names in ZIP Code order"

```
SELECT VendName, VendZipCode  
FROM Vendors  
ORDER BY VendZipCode;
```

### Sales Order Database

"Display the names of our employees, phone and ID numbers and list them by last name and first name"

```
SELECT EmpFirstName, EmpLastName, EmpPhoneNumber,  
EmployeeID  
FROM Employees  
ORDER BY EmpLastName, EmpFirstName;
```

### Query to represent cities of bowling league membership

Use + icon to add new query

USE Bowling\_Legend:

SELECT DISTINCT BowlerCity (distinct eleimate duplicate value)

From Bowlers

ORDER BY BowlerCity – Run

## PRACTICE QUESTIONS

### Sample Statements – Sales Order Database

What are the names and prices of all the products we carry?

```
SELECT ProductName, RetailPrice  
FROM Products;
```

Expected Results  
40 records

Which state do our customers come from?

```
SELECT DISTINCT CustState  
FROM Customers;
```

Expected Results  
4 records

## Sample Statements – Entertainment Agency Database

List all entertainers and the cities they are based in, sort the results by city and name in ascending order. *Code for Ascending order is ASC and for Descending order is DESC.*

```
SELECT EntStageName, EntCity  
FROM Entertainers  
ORDER BY EntCity ASC, EntStageName ASC;
```

Expected Results  
13 records

Give me a unique list of engagement dates. I am not concerned with how many engagements per date are there.

```
SELECT DISTINCT StartDate  
FROM Engagements;
```

Expected Results  
64 records

## Sample Statements – School Scheduling Database

Can we view complete class information?

```
SELECT *  
FROM Classes;
```

Expected Results  
147 records

Give me a list of the buildings on campus and the number of floors for each building. Sort the list by building in ascending order.

```
SELECT BuildingName, NumberOfFloors  
FROM Buildings  
ORDER BY BuildingName ASC;
```

Expected Results  
6 records

## CLASS # 12

### What is an Expression?

An Expression is some form of operation involving numbers, characters, strings, or date and time. It can use values drawn from specific columns in a table, constant or literal values or combination of both.

Expressions are especially useful when you are asking “what if” questions. Here are few example requests that you can answer using expressions”

*“Show me the start time, end time and duration for each class.”*

*“What if we raise prices of our products by 5 percent.”*

*“What is the estimated per hour rate for each engagement.”*

**Note:** To write an expression firstly check field of data type and then check its end result.

### Specifying Explicit Values

SQL provides flexibility for enhancing the information returned from a SELECT statement by allowing the use of constant values such as character strings, numbers, dates, times, or a suitable combination of these items in any valid expression used within a SELECT statement.

#### Character String Literals

```
SELECT VendWebPage, "is the web site for" As Remark, VendName  
FROM Vendors;
```

#### Numeric Literals

```
SELECT 'Retirement age of', empFullName, 'is', 65  
FROM Employees;
```

#### DateTime Literals

```
SELECT 'The price of product', prdProductName, 'was determined on', cast("2016-11-31", As Date) As Date  
FROM Products;
```

**Note:** Here String/ Numerical value is Explicit Value

# Types of SQL Expressions

Concatenation	Mathematical	Date and Time Arithmetic
<p>Two or more columns or literals, use &amp; signs.</p> <p>ColumnOne &amp; ColumnTwo EmpFirstName &amp; EmpLastName</p>	<p>RetailPrice * QuantityOnHand 25 +35 CostPrice + (CostPrice*2.5%)</p>	<p>ReviewDate + 90 StopTime – StartTime StartTime + '00:19'</p>

**Concatenation:** Merge strings

**Mathematical:** Merge numerical value

**Date and Time Arithmetic:** Merge date and time value E.g. Addition, Subtraction

The end result of value does not exist on table. So we have to assign a name to that expression known as **Naming Expression**

## Naming the Expression

When you use an expression in a SELECT clause, the result set includes a new column that displays the result of the operation defined in the expression. This new column is known as a calculated column and it must be named by using AS keyword. For example:

```
SELECT EmpFirstName & '' & EmpLastName, EmpPhoneNumber, EmpCity
FROM Employees;
```

Rewritten

```
SELECT Concat(EmpFirstName, ' ', EmpLastName) As EmpFullName, EmpPhoneNumber, EmpCity
FROM Employees;
```

## For Sales Order Database – Questions

### Examples – SQL Expressions - Sales Order Database

*"Show me value of inventory on hand by product?"*

*"How many days elapsed between the order date and the ship date for each order?"*

*"What will be new price if we reduce existing wholesale price by 5%?"*

#### Show me value of inventory on hand by product?

**Note:** For inventory: Retail Price (text data type) x Quality On Hand (integer data type) – but both have different data types so multiplication gives error

```
1 * USE sales_Orders;
2
3 * SELECT ProductName, RetailPrice, QuantityOnHand, RetailPrice * QuantityOnhand As InventoryValue
4   FROM Products
```

When we run above query it gives error

#### Note:

- To remove something from string – use **Replace** Function i.e. Replace(Name of column, “value which is going to replace”, “jo value hum replace karkey likhna chahtey han”)
- To convert text value in numerical value – use **Convert** Function
- To subtract two dates – use **Date if** Function
- To convert from string to date data type – **STR\_TO\_DATE** Function i.e. str\_to\_date(Text value which needs to convert into date, tell on which format need to be converted)
- For Difference between two dates – use **DateDiff** Function

**To convert text value in number value**

USE sales\_Orders,

```
SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice, ",",""), Decimal(15,2)) As NewRetailPrice
```

FROM Products – Run

Here 15 means – first 15 characters and 2 means – upto 2 decimal places

```
USE sales_Orders;

SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice, ",",""), Decimal(15,2)) As NewRetailP
FROM Products
```

---

**Show me value of inventory on hand by product?**

**Exprsion:**

Use sales\_Orders,

```
SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice, ",",""), Decimal(15,2))*QuantityOnHand As InventoryValue
```

FROM Products – Run

```
Price, QuantityOnHand, convert(replace(RetailPrice, ",",""), Decimal(15,2))*QuantityonHand As InventoryValue
```

---

**How many days elapsed between the order date and ship date for each order?**

Use sales\_Orders,

```
SELECT OrderName, OrderDate, ShipDate
```

FROM Orders – Run

```
USE sales_Orders;

-- SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice, ",",""), Decimal(15,2))*Quantityon
-- FROM Products

SELECT OrderNumber, OrderDate, ShipDate
FROM Orders
```

## To Convert String value into date time

USE sales\_Orders,

```
SELECT OrderName, OrderDate, ShipDate, str_to_date(Orderdate, "%d, %m, %Y") as  
NewOrderDate
```

FROM Orders – Run

```
USE sales_Orders;  
  
-- SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice,",",""),Decimal(15,2))*Quantityo  
-- FROM Products  
  
SELECT OrderNumber, OrderDate, ShipDate, str_to_date(Orderdate, "%d/%m/%Y") as NewOrderDate  
FROM Orders
```

## To calculate difference between two dates

USE sales\_Orders,

```
SELECT OrderName, OrderDate, ShipDate, DateDiff(str_to_date(ShipDate, "%d, %m, %Y") str_to_date(Orderdate, "%d, %m, %Y")) As DaysLapsed
```

FROM Orders – Run {Final Exprsion}

```
OrderDate, ShipDate, DateDiff(str_to_date(ShipDate, "%d/%m/%Y"), str_to_date(Orderdate, "%d/%m/%Y")) As DaysLapsed
```

### Note:

- OrderName (integer data type), OrderDate (text data type), ShipDate (text data type)
- Between expressions is in comments so it will not execute
- Date format in data: dd/mm/yy
- "%d, %m, %Y – means 2digit date, 2digit month, 4digit year

---

## What will be the new price if we reduce existing whole sale price by 5%?

USE sales\_Orders,

```
SELECT ProductNumber, convert(replace(wholesalePrice, ",",""), Decimal(15,2))*0.95  
As NewWholesalePrice
```

FROM Product\_vendors – Run

---

```

USE sales_Orders;

-- SELECT ProductName, RetailPrice, QuantityOnHand, convert(replace(RetailPrice,"","",),Decimal(15,2))*Quantity
-- FROM Products

-- SELECT OrderNumber, OrderDate, ShipDate, DateDiff(str_to_date(ShipDate, "%d/%m/%Y"), str_to_date(Orderdate,
-- -- FROM Orders

SELECT ProductNumber, convert(replace(WholesalePrice, "", ""), Decimal(15,2))*0.95 As NewWholesalePrice
FROM Product_vendors

```

## ANSWERS

### Examples – SQL Expressions - Sales Order Database

*"Show me value of inventory on hand by product?"*

```

SELECT ProductName, RetailPrice, QuantityOnHand,
       Convert(Replace(RetailPrice,"","",), Decimal(15,2)) * QuantityOnHand As NewPrice
FROM Products;

```

*"How many days elapsed between the order date and the ship date for each order?"*

```

SELECT OrderNumber, OrderDate, Shipdate,
       DateDiff(str_to_date(ShipDate, "%d/%m/%Y"), str_to_date(OrderDate, "%d/%m/%Y")) As DaysLapsed
FROM Orders;

```

*"What will be new price if we reduce existing wholesale price by 5%?"*

```

SELECT ProductNumber, WholesalePrice,
       Convert(Replace(WholesalePrice, "", ""), Decimal(15,2)) * 0.95 As NewPrice
FROM Product_Vendors;

```

### Sample Statements – Entertainment Agency Database

**How long is each engagement due to run?**

```

SELECT EngagementNumber, StartDate, EndDate,
       DateDiff(str_to_date(EndDate, "%d/%m/%Y"), str_to_date(StartDate, "%d/%m/%Y")) + 1 As EngagementWillLast
FROM Engagements;

```

Expected Results  
111 records

**Show the date of each agent's first six month performance review? Sort results by First name.**

```

SELECT AgtFirstName, AgtLastName, DateHired,
       Date_Add(str_to_date(DateHired, "%d/%m/%Y"), INTERVAL 6 MONTH) As PerformanceReviewDate
FROM Agents
       Order by AgtFirstName;

```

Expected Results  
9 records

## Sample Statements – School Scheduling Database

List how many complete years each staff member has been with the school as of October 1, 2017 and sort the result by last name and first name?

```
SELECT StfFirstName, StfLastName, DateHired,  
       Format(DateDiff("2017-10-01", STR_TO_DATE(DateHired,"%d/%m/%Y")) / 365, 0) As ServiceYears  
FROM Staff;
```

Show me a list of staff members, their salaries, and a proposed 7% bonus for each staff.

```
SELECT Concat(StfFirstName, " ", StfLastName) As FullName, Salary,  
       Convert(Replace(Salary, ",", ""), Decimal(15,2)) * 0.07 As Proposed_Bonus  
FROM Staff;
```

## Sample Statements – Bowling League Database

Display a list of all bowlers and addresses formatted suitably for a mailing list, sorted by ZIP Code?

```
SELECT Concat(BowlerFirstName, " ", BowlerMiddleInit, " ", BowlerLastName) As Name,  
       BowlerAddress, Concat(BowlerCity, ", ", BowlerState) As CityAndState,  
       BowlerZip, BowlerPhoneNumber  
FROM Bowlers  
ORDER BY BowlerZip;
```

Give me a listing of each team's lineup.

```
SELECT TeamID, Concat(BowlerFirstName, " ", BowlerLastName) As Bowler  
FROM Bowlers;
```

## FILTRATION

### Filtering your data

- ✓ To filter data, use WHERE clause in SELECT statement.
- ✓ The WHERE clause contains a search condition that it uses as filter.
- ✓ A search condition contains one or more predicates, each of which is an expression that test one or more value expressions and returns a true, false or unknown answer.
- ✓ A **predicate** defines a logical condition being applied to rows in a table.
- ✓ There are five basic predicates : COMPARISON, BETWEEN, IN, LIKE and IS NULL.

Comparison	Between	IN	LIKE	IS NULL
= Equal to <> Not equal to < Less than > Greater than <= <= = >= >= =	The BETWEEN predicate lets you test whether the value expression falls within specified range of values.	Whether the values of a given value expression matches an item in a given list of values.	Whether a character string value expression matches a specified character string pattern.	Use the IS NULL predicate to determine whether a value expression evaluates to Null.

**For range of values – use Between**

**For list of values – use IN**

**Null** – means nothing while **Blank** – means there is something but it is not visible like space

**Note:**

- We use **WHERE** clause for criteria
- **ON** means – equal to
- After equal to date should be opposite of date format  
e.g. "%d, %m, %Y" = "2000-02-05"

---

**Show me the first and last names of all the agents who were hired on Feb 05 2000.**

USE Entertainment\_Agency;

SELECT AgtFirstname, AgtLastname, DateHired

FROM Agents

WHERE str\_to\_date(DateHired, "%d, %m, %Y") = "2000-02-05" – Run

## COMPARISON OPERATOR QUESTIONS

### Examples – Filtering your data

Entertainment Agency	Entertainment Agency
<p><i>"Show me the first and last names of all the agents who were hired on Feb. 05 2000."</i></p> <pre>SELECT AgtFirstName, AgtLastName, DateHired, str_to_date(DateHired, "%d/%m/%Y") As newdate FROM Agents WHERE str_to_date(DateHired, "%d/%m/%Y") = "2000-02-05"</pre>	<p><i>"Give me a list of customers whose zip code is not equal to 98033."</i></p> <pre>SELECT CustFirstName, CustLastName, CustZipCode FROM Customers WHERE CustZipCode &lt;&gt; "98033"</pre>
Sales Order	Sales Order
<p><i>"Give me record of those orders that were shipped after 01/01/2018."</i></p> <pre>SELECT OrderNumber, ShipDate, str_to_date(ShipDate, "%d/%m/%Y") As NewDate FROM Orders WHERE str_to_date(ShipDate, "%d/%m/%Y") &gt;"2018-01-01"</pre>	<p><i>"Are there any orders where the ship date was accidentally posted earlier than the order date?"</i></p> <pre>SELECT OrderNumber, OrderDate, ShipDate FROM Orders WHERE str_to_date(ShipDate, "%d/%m/%Y") &lt; str_to_date(OrderDate, "%d/%m/%Y")</pre>

## REMAINING OPERATOR QUESTIONS

### Examples – Filtering your data

Sales Order	Sales Order
<p><i>"Give me detail of orders that were shipped during the month of December 2017."</i></p> <div style="background-color: #E0F2F1; height: 100px; width: 100%;"></div>	<p><i>"Show me list of customers whose street address contains the word "Kingman".</i></p> <div style="background-color: #E0F2F1; height: 100px; width: 100%;"></div>
Sales Order	Sales Order
<p><i>"Give me complete details of all those orders that were shipped in 2018 on Jan 1, Feb 10, and Mar 15."</i></p> <div style="background-color: #E0F2F1; height: 100px; width: 100%;"></div>	<p><i>"Give me list of vendors who did not provide us email address."</i></p> <div style="background-color: #E0F2F1; height: 100px; width: 100%;"></div>

**Give me detail of orders that were shipped during the month of December 2017.**

USE sales\_Orders;

SELECT \*

FROM Orders

WHERE str\_to\_date(Shipdate, "%d, %m, %Y") Between "2017-12-01" And "2017-12-31"

– Run

```
1 • USE sales_Orders;
2
3 • SELECT *
4   FROM Orders
5 WHERE str_to_date(Shipdate, "%d/%m/%Y") Between "2017-12-01" And "2017-12-31"
```

---

**Show me list of customers whose street address contains the word “Kingman”.**

USE sales\_Orders;

SELECT \*

FROM Customers

WHERE CustStreetAddress Like "&Kingman&" – Run

```
1 • USE sales_Orders;
2
3 • SELECT *
4   FROM Customers
5 WHERE CustStreetAddress Like "%Kingman%"
```

---

**Give me complete details of all those orders that were shipped in 2018 on Jan 1, Feb 10 and Mar 15.**

USE sales\_Orders;

SELECT \*

FROM Orders

WHERE str\_to\_date(Shipdate, "%d, %m, %Y") IN("2018-01-01", "2018-02-10", "2018-03-15") – Run

```

1 • USE sales_Orders;
2
3 • SELECT *
4   FROM Orders
5   WHERE str_to_date(ShipDate, "%d/%m/%Y") IN("2018-01-01", "2018-02-10", "2018-03-15")

```

**Give me list of vendors who do not provide us email address.**

USE sales\_Orders;

```

SELECT *
FROM Vendors
WHERE VendEmailAddress = "" – Run

```

```

1 • USE sales_Orders;
2
3 • SELECT *
4   FROM Vendors
5   -- WHERE VendEmailAddress Is Not Null
6   WHERE VendEmailAddress = ""

```

**Note:** Here we do not use is not null – because the column contains some data which is not visible so instead of that we use blank in double quotation to get the answer

## Examples – Filtering your data

Sales Order	Sales Order
<p><i>"Give me detail of orders that were shipped during the month of December 2017."</i></p> <pre> SELECT * FROM Orders WHERE str_to_date(Shipdate, "%d/%m/%Y") Between "2017-12-01" And "2017-12-31" </pre>	<p><i>"Show me list of customers whose street address contains the word "Kingman".</i></p> <pre> SELECT * FROM Customers WHERE CustStreetAddress Like "%Kingman%" </pre>
Sales Order	Sales Order
<p><i>"Give me complete details of all those orders that were shipped in 2018 on Jan 1, Feb 10, and Mar 15."</i></p> <pre> SELECT * FROM Orders WHERE str_to_date(Shipdate, "%d/%m/%Y") IN("2018-01-01", "2018-02-10", "2018-03-15") </pre>	<p><i>"Give me list of vendors who did not provide us email address."</i></p> <pre> SELECT * FROM Vendors -- WHERE VendEmailAddress Is Null -- WHERE VendEmailAddress Is Not Null WHERE VendEmailAddress = "" </pre>

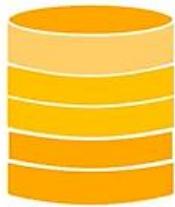
# What is Big Data?

**Big Data**=Enormous amounts and various types of data that is collected for a certain purpose

## Volume

Enormous amounts of data reaches a few terabytes to dozens of terabytes or more

Enormous capacity



## Variety

Various data formats, such as documents, images, audio, videos, and search and browsing history



## Velocity

Real-time updates



## SQL vs. MySQL vs NoSQL

**SQL** or the **Structured Query Language** is a programming language that enables the function of retrieving, managing, storing the data in the relational database management system.

**MySQL**, now owned and managed by Oracle Corporation, is a type of relational database management system. It is an open-source platform that allows one to store, retrieve and manage relational databases.

**NoSQL**, or commonly known as **Not only SQL database**, provides a mechanism for storage and retrieval of unstructured data. This type of database can handle a humongous amount of data and has a dynamic schema. So, a NoSQL database has no specific query language, no or a very few relationships, but has data stored in the format of collections and documents.

Non structure / No relationships – NoSQL

Fourth One – Microsoft SQL

Format	Description
%a	Abbreviated weekday name (Sun to Sat)
%b	Abbreviated month name (Jan to Dec)
%c	Numeric month name (0 to 12)
%D	Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ...)
%d	Day of the month as a numeric value (01 to 31)
%e	Day of the month as a numeric value (0 to 31)
%f	Microseconds (000000 to 999999)
%H	Hour (00 to 23)
%h	Hour (00 to 12)
%I	Hour (00 to 12)

%i	Minutes (00 to 59)
%j	Day of the year (001 to 366)
%k	Hour (0 to 23)
%l	Hour (1 to 12)
%M	Month name in full (January to December)
%m	Month name as a numeric value (01 to 12)
%p	AM or PM
%r	Time in 12 hour AM or PM format (hh:mm:ss AM/PM)
%S	Seconds (00 to 59)
%s	Seconds (00 to 59)
%T	Time in 24 hour format (hh:mm:ss)
%U	Week where Sunday is the first day of the week (00 to 53)
%u	Week where Monday is the first day of the week (00 to 53)
%V	Week where Sunday is the first day of the week (01 to 53). Used with %X
%v	Week where Monday is the first day of the week (01 to 53). Used with %X
%W	Weekday name in full (Sunday to Saturday)
%w	Day of the week where Sunday=0 and Saturday=6
%X	Year for the week where Sunday is the first day of the week. Used with %V
%x	Year for the week where Monday is the first day of the week. Used with %V
%Y	Year as a numeric, 4-digit value
%y	Year as a numeric, 2-digit value