STEP 1 – ETL Process in SQL Server

So the first step in churn analysis is to load the data from our source file. For this purpose i will be using Microsoft SQL server because it is a widely used solution across the industry and also because a full-fledged Database System is better at handling recurring data loads and maintaining data integrity compared to an excel file.

In order for us to run our sql queries Microsoft provides us with GUI interface which is known as **SQL Server Management Studio**.

Creating Database

```
CREATE DATABASE db_Churn
```

Data Exploration – Check Distinct Values

```
SELECT Gender, Count(Gender) as TotalCount,
Count(Gender)  1.0 / (Select Count() from stg_Churn)  as Percentage
from stg_Churn
Group by Gender

SELECT Contract, Count(Contract) as TotalCount,
Count(Contract)  1.0 / (Select Count() from stg_Churn)  as Percentage
from stg_Churn
Group by Contract

SELECT Customer_Status, Count(Customer_Status) as TotalCount, Sum(Total_R
evenue) as TotalRev,
Sum(Total_Revenue) / (Select sum(Total_Revenue) from stg_Churn) * 100  as R
evPercentage
```

```
from stg_Churn
Group by Customer_Status

SELECT State, Count(State) as TotalCount,
Count(State)  1.0 / (Select Count() from stg_Churn)  as Percentage
from stg_Churn
Group by State
Order by Percentage desc
```

Data Exploration – Check Nulls

```
SELECT
   SUM(CASE WHEN Customer_ID IS NULL THEN 1 ELSE 0
END) AS Customer_ID_Null_Count,

   SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0
END) AS Gender_Null_Count,

   SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) AS Age_Null_Count,

   SUM(CASE WHEN Married IS NULL THEN 1 ELSE 0
END) AS Married_Null_Count,

   SUM(CASE WHEN State IS NULL THEN 1 ELSE 0 END) AS State_Null_Count,

   SUM(CASE WHEN Number_of_Referrals IS NULL THEN 1 ELSE 0
END) AS Number_of_Referrals_Null_Count,
```

```
   SUM(CASE WHEN Tenure_in_Months IS NULL THEN 1 ELSE 0
END) AS Tenure_in_Months_Null_Count,

   SUM(CASE WHEN Value_Deal IS NULL THEN 1 ELSE 0
END) AS Value_Deal_Null_Count,

   SUM(CASE WHEN Phone_Service IS NULL THEN 1 ELSE 0
END) AS Phone_Service_Null_Count,

   SUM(CASE WHEN Multiple_Lines IS NULL THEN 1 ELSE 0
END) AS Multiple_Lines_Null_Count,

   SUM(CASE WHEN Internet_Service IS NULL THEN 1 ELSE 0
END) AS Internet_Service_Null_Count,

   SUM(CASE WHEN Internet_Type IS NULL THEN 1 ELSE 0
END) AS Internet_Type_Null_Count,

   SUM(CASE WHEN Online_Security IS NULL THEN 1 ELSE 0
END) AS Online_Security_Null_Count,

   SUM(CASE WHEN Online_Backup IS NULL THEN 1 ELSE 0
END) AS Online_Backup_Null_Count,

   SUM(CASE WHEN Device_Protection_Plan IS NULL THEN 1 ELSE 0
END) AS Device_Protection_Plan_Null_Count,
```

```
   SUM(CASE WHEN Premium_Support IS NULL THEN 1 ELSE 0
END) AS Premium_Support_Null_Count,

   SUM(CASE WHEN Streaming_TV IS NULL THEN 1 ELSE 0
END) AS Streaming_TV_Null_Count,

   SUM(CASE WHEN Streaming_Movies IS NULL THEN 1 ELSE 0
END) AS Streaming_Movies_Null_Count,

   SUM(CASE WHEN Streaming_Music IS NULL THEN 1 ELSE 0
END) AS Streaming_Music_Null_Count,

   SUM(CASE WHEN Unlimited_Data IS NULL THEN 1 ELSE 0
END) AS Unlimited_Data_Null_Count,

   SUM(CASE WHEN Contract IS NULL THEN 1 ELSE 0
END) AS Contract_Null_Count,

   SUM(CASE WHEN Paperless_Billing IS NULL THEN 1 ELSE 0
END) AS Paperless_Billing_Null_Count,

   SUM(CASE WHEN Payment_Method IS NULL THEN 1 ELSE 0
END) AS Payment_Method_Null_Count,

   SUM(CASE WHEN Monthly_Charge IS NULL THEN 1 ELSE 0
END) AS Monthly_Charge_Null_Count,

   SUM(CASE WHEN Total_Charges IS NULL THEN 1 ELSE 0
END) AS Total_Charges_Null_Count,
```

```
   SUM(CASE WHEN Total_Refunds IS NULL THEN 1 ELSE 0
END) AS Total_Refunds_Null_Count,

   SUM(CASE WHEN Total_Extra_Data_Charges IS NULL THEN 1 ELSE 0
END) AS Total_Extra_Data_Charges_Null_Count,

   SUM(CASE WHEN Total_Long_Distance_Charges IS NULL THEN 1 ELSE 0
END) AS Total_Long_Distance_Charges_Null_Count,

   SUM(CASE WHEN Total_Revenue IS NULL THEN 1 ELSE 0
END) AS Total_Revenue_Null_Count,

   SUM(CASE WHEN Customer_Status IS NULL THEN 1 ELSE 0
END) AS Customer_Status_Null_Count,

   SUM(CASE WHEN Churn_Category IS NULL THEN 1 ELSE 0
END) AS Churn_Category_Null_Count,

   SUM(CASE WHEN Churn_Reason IS NULL THEN 1 ELSE 0
END) AS Churn_Reason_Null_Count

FROM stg_Churn;
```

Remove null and insert the new data into Prod table

```
SELECT
   Customer_ID,
   Gender,
```

```
Age
Married,
State,
Number_of_Referrals,
Tenure_in_Months,
ISNULL(Value_Deal, 'None') AS Value_Deal,
Phone_Service,
ISNULL(Multiple_Lines, 'No') As Multiple_Lines,
Internet_Service,
ISNULL(Internet_Type, 'None') AS Internet_Type,
ISNULL(Online_Security, 'No') AS Online_Security,
ISNULL(Online_Backup, 'No') AS Online_Backup,
ISNULL(Device_Protection_Plan, 'No') AS Device_Protection_Plan,
ISNULL(Premium_Support, 'No') AS Premium_Support,
ISNULL(Streaming_TV, 'No') AS Streaming_TV
ISNULL(Streaming_Movies, 'No') AS Streaming_Movies,
ISNULL(Streaming_Music, 'No') AS Streaming_Music,
ISNULL(Unlimited_Data, 'No') AS Unlimited_Data,
Contract,
Paperless_Billing,
Payment_Method,
Monthly_Charge,
Total_Charges,
Total_Refunds,
Total_Extra_Data_Charges,
```

```
    Total_Long_Distance_Charges,
    Total_Revenue,
    Customer_Status,
    ISNULL(Churn_Category, 'Others') AS Churn_Category
    ISNULL(Churn_Reason , 'Others') AS Churn_Reason
INTO [db_Churn].[dbo].[prod_Churn]
FROM [db_Churn].[dbo].[stg_Churn];
```

Create View for Power BI

```
Create View vw_ChurnData as
    select * from prod_Churn where Customer_Status In ('Churned', 'Stayed')

Create View vw_JoinData as
    select * from prod_Churn where Customer_Status = 'Joined'
```

STEP 2 – Power BI Transform

Add a new column in prod_Churn

1. Churn Status = if [Customer_Status] = “Churned” then 1 else 0
2. Change Churn Status data type to numbers
3. Monthly Charge Range = if [Monthly_Charge] < 20 then “< 20” else if [Monthly_Charge] < 50 then “20-50” else if [Monthly_Charge] < 100 then “50-100” else “> 100”

Create a New Table Reference for mapping_AgeGrp

1. Keep only Age column and remove duplicates
2. Age Group = if [Age] < 20 then “< 20” else if [Age] < 36 then “20 – 35” else if [Age] < 51 then “36 – 50” else “> 50”

3. AgeGrpSorting = if [Age Group] = “< 20” then 1 else if [Age Group] = “20 – 35” then 2 else if [Age Group] = “36 – 50” then 3 else 4

4. Change data type of AgeGrpSorting to Numbers

Create a new table reference for mapping_TenureGrp

1. Keep only Tenure_in_Months and remove duplicates

2. Tenure Group = if [Tenure_in_Months] < 6 then “< 6 Months” else if [Tenure_in_Months] < 12 then “6-12 Months” else if [Tenure_in_Months] < 18 then “12-18 Months” else if [Tenure_in_Months] < 24 then “18-24 Months” else “>= 24 Months”

3. TenureGrpSorting = if [Tenure_in_Months] = “< 6 Months” then 1 else if [Tenure_in_Months] = “6-12 Months” then 2 else if [Tenure_in_Months] = “12-18 Months” then 3 else if [Tenure_in_Months] = “18-24 Months ” then 4 else 5

4. Change data type of TenureGrpSorting to Numbers

Create a new table reference for prod_Services

1. Unpivot services columns

2. Rename Column