

# **TELECOM CHURN**

---

**Prepared by:**

**Mohamed Atef**

**Mostafa Yassin**

**Lotfy Ahmed**

**Mahmoud Ali**

**Kerolos Mahrous**

**Esalm Ahmed**



# **Introduction**

Customer churn refers to the phenomenon of customers ceasing to do business with a company or brand. In the context of the telecommunication industry, churn occurs when customers end their subscription with a particular telecom service provider and switch to another provider or terminate the service altogether. Churn can be a significant problem for companies as it can result in lost revenue, decreased market share, and increased costs associated with acquiring new customers to replace those who have left.

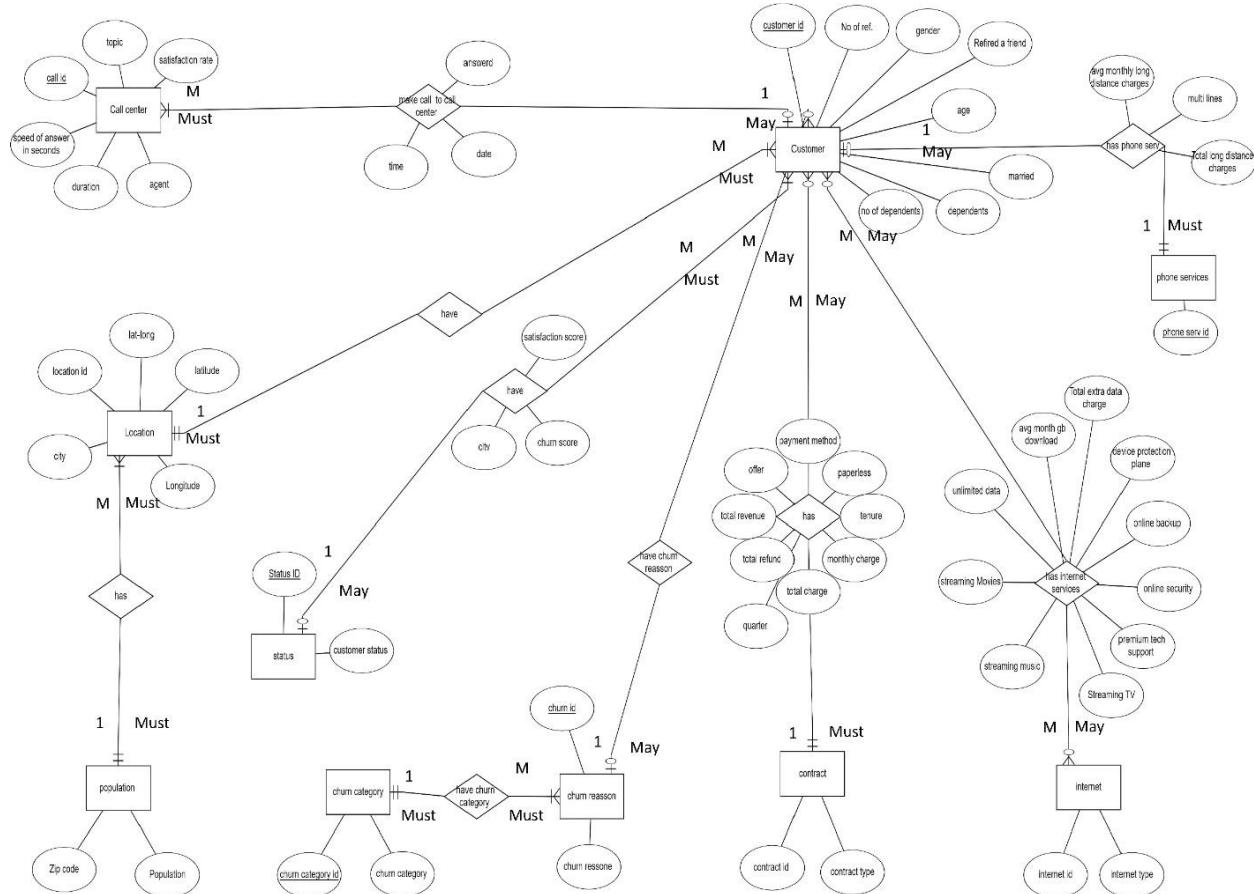
In this report initial conceptualization for the Database project requirements for the Power BI Track at the Information Technology Institute. Our project focuses on designing a Data Model, creating a Database, and Analyzing the service performance of a telecom company within the telecommunications field. We selected this field due to its paramount importance, as the ability to share and receive information is essential for the smooth functioning of modern society. Telecommunication serves as the foundation of today's digital environment.

## **Project Value:**

The objective of this project was to examine the rate at which customers discontinue using a business's product or service, known as the Customer Churn Rate, and identify the key factors that influence it. Additionally, the project aims to evaluate the satisfaction level of customers with the call service.

In the telecommunications industry, the churn rate serves as a crucial performance metric, allowing companies to forecast future revenue trends and gauge overall customer satisfaction. By understanding why customers leave and how many are departing, businesses can.

# ERD(entity relationship diagram):

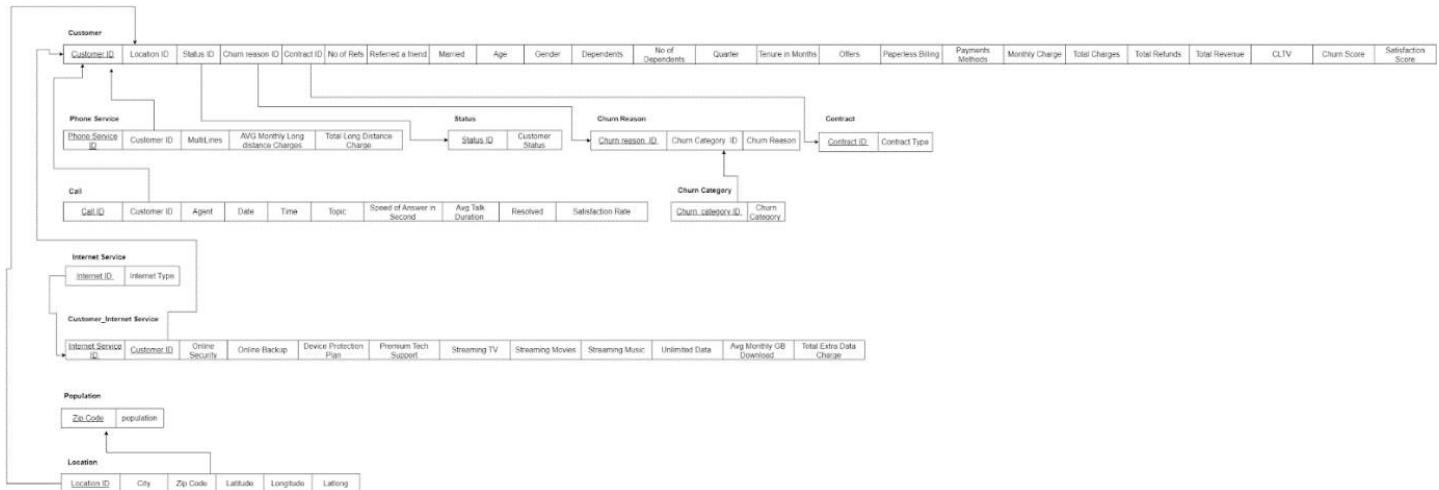


Our ERD consists of:

1. Main Entities:
  - a. Customers.
  - b. Phone Services.
  - c. Internet Service.
  - d. Contract.
  - e. Churn Reason.
  - f. Churn Category
  - g. Status.
  - h. Location.
  - i. Population.
  - j. Call Center.
2. Main Entities:
  - a. Customers - Phone Services Relationship.
  - b. Customers - Internet Service Relationship.
  - c. Customers - Contract Relationship.
  - d. Customers - Churn Reason Relationship.
  - e. Churn Reason - Churn Category Relationship.

- f. Customers - Status Relationship.
- g. Customers - Location Relationship.
- h. Location - Population Relationship.
- i. Customers – Call Center Relationship.

## Mapping:



## Customers

Customer ID PK, Location ID FK, Status ID FK, Churn reason ID FK, Contract ID FK, Number of refs, Referred a friend, Gender, Age, Married, Dependents, Number of Dependents, Quarter, Tenure in Months, Offer, Paperless Billing, Payment Method, Monthly Charge, Total Charges, Total Refunds, Total Revenue, CLTV, Churn Score, Satisfaction Score, Join Date.

## Customers & Location

Relation between Customers and Locations is many to one, must from both sides. A customer must have one location, and the location must belong to one or many customers. So, the PRIMARY KEY of the “one” table Locations will be added as a FOREIGN KEY in the “many” Table Customers.

## Customers & Status

The relation between Customers and Status is one to many, must from the Customers side and may from the Status side. A customer must have one status, and the status may belong to one or many customers. So, the PRIMARY KEY of the “one” table Status will be added as a FOREIGNKEY in the “many” Table Customers. Also, the relation attributes will go to the “many” Table Customers. Relation attributes are: CLTV, Churn Score, and Satisfaction Score.

## **Customers & Churn Reasons**

The relation between Customers and Churn Reasons is one to many, may from both sides. A customer may be churned so in this case he/she will have one churn reason, and the Churn Reasons may belong to one or many customers. So, the PRIMARY KEY of the “one” table ChurnReasons will be added as a FOREIGN KEY in the “many” Table Customers.

## **Customers & Contracts**

The relation between Customers and Contracts is one to many, must from the Customer's side and may from the Contracts side. A customer must have a contract, and the contract may belong to one or many customers. So, the PRIMARY KEY of the “one” table Contracts will be added as a FOREIGN KEY in the “many” Table Customers.

## **Customer Internet Service**

(Internet Service ID FK, Customer ID FK) PK, Online Security, Online Backup, Device Protection Plan, Premium Tech Support, Streaming TV, Streaming Movies, Streaming Music, Unlimited Data, Avg Monthly GB Download, Total Extra Data Charges.

## **Customers & Internet Service**

The relation between Customers and Internet Service is many to many, may from both sides. A customer may have one or more Internet Services, and the Internet Service may belong to one or many customers. So, we will create a new table Customer Internet Service that contains Customers & Internet Service PRIMARY KEYS as a composite key (Internet Service ID FK, Customer ID FK) PK. Also, the relation attributes will be added to the new table. The relation attributes are:  
Online Security, Online Backup, Device Protection Plan, Premium Tech Support, Streaming TV, Streaming Movies, Streaming Music, Unlimited Data, Avg Monthly GB Download, Total Extra Data Charges.

## **Phone Service**

The relation between Customer & Phone Service is one to one .The customer May have Phone service and the phone service Must have Customer so get PRIMARY KEY from Customer and put it in Phone service as Foreign key.  
Phone Service ID PK, Customer ID FK, Multiple Lines, average monthly Long-Distance Charges, Total Long-Distance Charges.

## **Location**

The relation between Location and Population is one from Population side to many from Location side, So the Primary of Population will be FOREIGN KEY in Location. Location Must have Zip Code and Population Must Zip Code in Location.  
(Location ID PK, Zip code FK, Latitude, Longitude, Latlong, City).

## Call

The relation between Customer and Call is one to many. The customer has many calls, but there is one Customer for one call. The Customer may have Call but Call Must have Customer.

(Call ID PK, Customer ID FK, Date, Time, Topic, Agent, Resolved, Speed of answer, Duration, Answered, Avg\_Talk\_Duration, Satisfaction rate)

## Churn Reason

The relation between Churn Category and Churn Reason is one to many. The Churn Category has many reasons but one reason has one Churn Category. The reason must have Category and Category must have Reason, so the PRIMARY KEY of Churn Category in Churn Reason as foreign key

(Churn reason ID PK, Churn Category ID FK, Churn Reason).

## Create database by SQL Server:

### Create Population Table:

```
create Table Population (
    Zipcode_ID int primary key ,
    Population_Num int check (Population_Num>0 ));
```

### Create Location Table:

```
create table Location (
    Location_ID int IDENTITY primary key ,
    City Varchar(50) ,
    Zipcode int foreign key references Population(Zipcode_ID),
    Latitude FLOAT,
    Longitude FLOAT
);
```

### Create Churn Category Table:

```
CREATE TABLE Churn_Category(
    Churn_Category_ID INT PRIMARY KEY ,
    Churn_Category VARCHAR(50) CHECK (Churn_Category IN
        ('Competitor','Dissatisfaction','Price', 'Attitude', 'Other')));
```

### Create Churn Reason Table:

```
CREATE TABLE Churn_Reason(
    Churn_Reason_ID INT PRIMARY KEY ,
    Churn_Category_ID INT foreign key references Churn_Category(Churn_Category_ID),
    Churn_Reason VARCHAR(150));
```

## Create Churn Reason Table:

```
CREATE TABLE Contract (
    Contract_ID INT PRIMARY KEY,
    Contract_Type VARCHAR(50) CHECK
        (Contract_Type IN ('Month-to-Month', 'One Year', 'Two Year')));
```

## Create Status Table:

```
CREATE TABLE Status (
    Status_ID INT PRIMARY KEY,
    Customer_Status VARCHAR(50) CHECK
        (Customer_Status IN ('Churned', 'Stayed', 'Joined')));
```

## Create Customer Table:

```
create table Customer(
Customer_ID int Primary Key,
Location_ID int foreign key references Location(Location_ID),
Status_ID int foreign key references Status(Status_ID),
Churn_reason_ID int foreign key references Churn_Reason(Churn_Reason_ID),
Contract_ID int foreign key references Contract(Contract_ID),
Number_Of_Referrals int Default 0,
Referred_A_Friend bit Default 0 CHECK(Referred_A_Friend in (0,1)),
Gender varchar(10) Default 'Male' CHECK(Gender in ('Female','Male')),
Age int CHECK(Age between 18 and 120),
Married bit Default 0 CHECK(Married in (0,1)),
Dependents bit Default 0 CHECK(Dependents in (0,1)),
Number_Of_Dependents int Default 0,
Quarter varchar(10),
Tenure_In_Months int,
Offer varchar(10) CHECK(Offer in ('A','B','C','D','E','None')),
Paperless_Billing bit Default 0 CHECK(Paperless_Billing in (0,1)) ,
Payment_Method varchar(50) Default 'Bank_Transfe' CHECK
(Payment_Method in ('Bank Withdrawal', 'Credit Card', 'Mailed Check')),
Monthly_Charge float,
Total_Charges float,
Total_Refunds float,
Total_Revenue float,
CLTV float,
Churn_Score int CHECK(Churn_Score between 0 and 100),
Satisfaction_Score int CHECK(Satisfaction_Score between 1 and 5));
```

## Create Internet Service Table:

```
CREATE TABLE Internet_Service (
    Internet_Service_ID INT PRIMARY KEY,
    Internet_Type VARCHAR(50) CHECK
        (Internet_Type IN ('Cable', 'DSL', 'Fiber optic')));
```

## Create Phone Service Table:

```
create Table Phone_Service (
Phone_Service_ID int PRIMARY KEY ,
Customer_Id int foreign key references Customer(Customer_ID) ,
Multiple_Lines bit Default 0 check (Multiple_Lines in(0,1)) ,
Avg_Monthly_Long_Distance_Charges decimal(10,3) CHECK
(Avg_Monthly_Long_Distance_Charges >= 0),
Total_Long_Distance_Charges decimal(10,3) CHECK
(Total_Long_Distance_Charges >= 0));
```

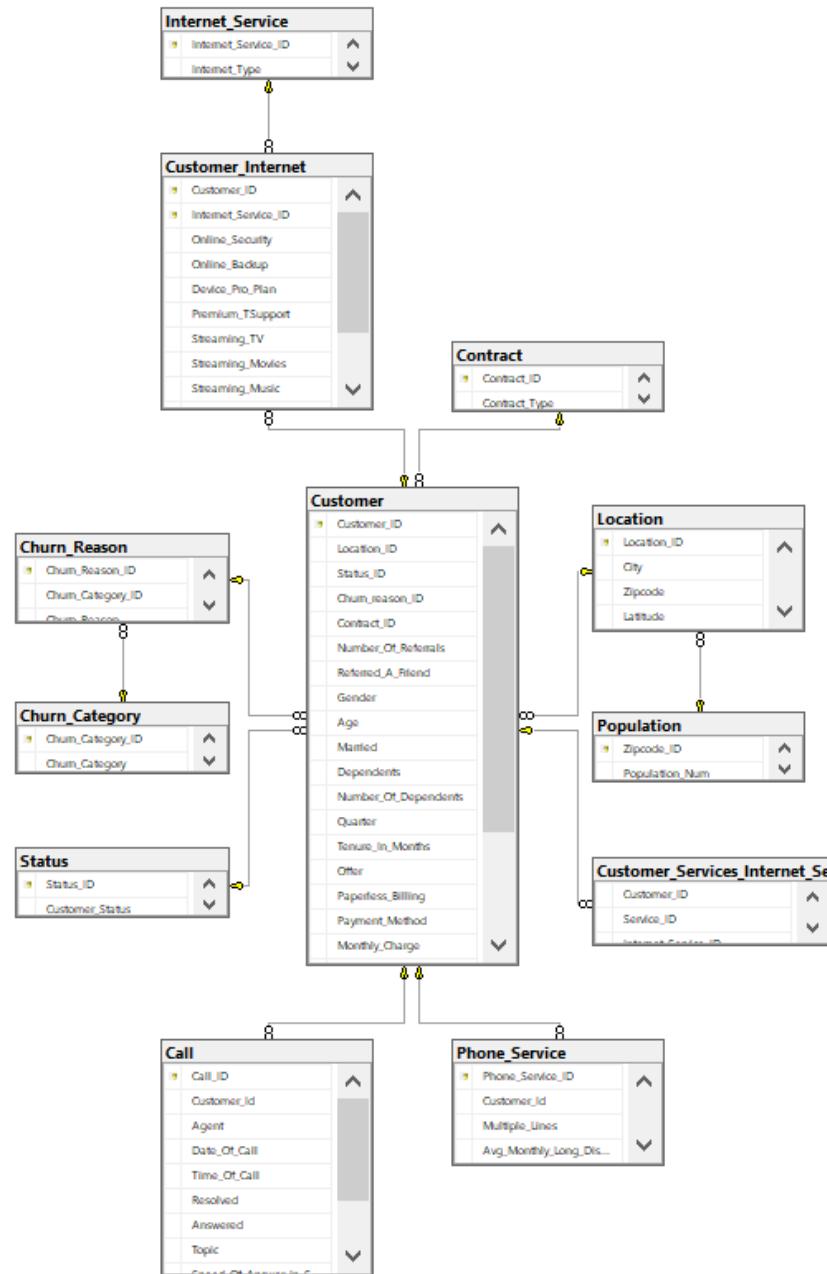
## Create Customer internet Table:

```
CREATE TABLE Customer_Internet (
    Customer_ID int foreign key references Customer(Customer_ID),
    Internet_Service_ID int foreign key references
Internet_Service(Internet_Service_ID),
    PRIMARY KEY (Customer_ID, Internet_Service_ID),
    Online_Security bit Default 0 CHECK(Online_Security in (0,1)),
    Online_Backup bit Default 0 CHECK(Online_Backup in (0,1)),
    Device_Pro_Plan bit Default 0 CHECK(Device_Pro_Plan in (0,1)),
    Premium_TSupport bit Default 0 CHECK(Premium_TSupport in (0,1)),
    Streaming_TV bit Default 0 CHECK(Streaming_TV in (0,1)),
    Streaming_Movies bit Default 0 CHECK(Streaming_Movies in (0,1)),
    Streaming_Music bit Default 0 CHECK(Streaming_Music in (0,1)),
    Unlimited_Data bit Default 0 CHECK(Unlimited_Data in (0,1)),
    Avg_Monthly_GB int Default 0,
    Total_Extra_Data_Charges int Default 0);
```

## Create Call Center Table:

```
CREATE TABLE Call(
Call_ID int primary key ,
Customer_Id INT FOREIGN KEY REFERENCES Customer (Customer_ID),
Agent VARCHAR(50),
Date_Of_Call DATE ,
Time_Of_Call TIME,
Resolved bit Default 0 CHECK(Resolved in (0,1)),
Answered bit Default 0 CHECK(Answered in (0,1)),
Topic varchar(50) CHECK(Topic in
('Contract related','Technical Support', 'Payment related', 'Admin Support',
'Streaming')),
Speed_Of_Answer_In_Seconds int CHECK (Speed_Of_Answer_In_Seconds >= 0),
Duration Time ,
Satisfaction_Rating int CHECK (Satisfaction_Rating BETWEEN 1 AND 5),)
```

## Create Diagrams:



## Insights:

- Number of churns per churn category.

```
SELECT churcat.Churn_Category ,COUNT(Customer_ID) [Total number of churns]
FROM Customer cus INNER JOIN Churn_Reason churea
ON cus.Churn_reason_ID = churea.Churn_Reason_ID
INNER JOIN Churn_Category churcat
ON churea.Churn_Category_ID = churcat.Churn_Category_ID
GROUP BY churcat.Churn_Category.
```

	Churn_Category	Total number of churns
1	Attitude	314
2	Other	182
3	Price	211
4	Competitor	841
5	Dissatisfaction	321

- The churn rate by agent

```
SELECT cal.Agent,
ROUND(CAST(COUNT(cust.Customer_ID ) AS float)/(SELECT COUNT(cust1.Customer_ID) FROM
Customer cust1, call cal1
WHERE cal1.Customer_Id = cust1.Customer_ID AND cal.Agent = cal1.Agent
GROUP BY cal1.Agent), 3)*100 [Churn Rate]
FROM Customer cust, call cal
WHERE cal.Customer_Id = cust.Customer_ID AND cust.Churn_reason_ID IN (SELECT
Churn_reason_ID FROM Churn_Reason)
GROUP BY cal.Agent
```

	Agent	Churn Rate
1	Diane	25.1
2	Dan	25.9
3	Joe	25.3
4	Becky	27.3
5	Stewart	26.8
6	Greg	27.2
7	Martha	26.5
8	Jim	29.9

- Number of churn by offer.

```
SELECT c.Offer,COUNT(c.Customer_ID) as NumberCusomerChurn ,AVG(c.Churn_Score) as
Avg_churn_score
from Customer c INNER JOIN Churn_Reason cr
ON cr.Churn_Reason_ID=c.Churn_reason_ID
GROUP BY c.Offer
```

	Offer	NumberCusomerChurn	Avg_churn_score
1	None	1051	81
2	Offer A	35	82
3	Offer B	101	82
4	Offer D	161	80
5	Offer E	426	81
6	Offer C	95	82

- Number of customers per services

```

SELECT
    SUM(CAST(Premium_TSupport AS int)) Premium_TSupport, SUM(CAST(Online_Backup AS int))
    Online_Backup, SUM(CAST(Online_Security AS int)) Online_Security,
    SUM(CAST(Streaming_Movies AS int)) Streaming_Movies, +SUM(CAST(Streaming_Music AS int))
    Streaming_Music, SUM(CAST(Streaming_TV AS int)) Streaming_TV,
    SUM(CAST(Unlimited_Data AS int)) Unlimited_Data
    FROM Customer_Internet

```

	Premium_TSupport	Online_Backup	Online_Security	Streaming_Movies	Streaming_Music	Streaming_TV	Unlimited_Data
1	2044	2429	2019	2732	2488	2707	4745

- Number of population, customers, Customer Number ,churn Number for city.

```

CREATE or alter VIEW city_states AS
SELECT City,sum(Population_Num) as PopNum,cast( COUNT(Customer_ID) as float) as CustNum ,
    isnull((select COUNT(c1.Customer_ID)
    from Customer c1 ,Location l1
    WHERE c1.Location_ID=l1.Location_ID and l1.City=l.city AND c1.Status_ID=3
    GROUP by l1.City),0) as ChurnNum
from Customer INNER join Location l on Customer.Location_ID=l.Location_ID
INNER JOIN Population on l.Zipcode =Zipcode_ID
GROUP by City

```

	City	PopNum	CustNum	ChurnNum
1	Acampo	25268	4	3
2	Acton	31324	4	0
3	Adelanto	94900	5	1
4	Adin	2460	4	2
5	Agoura Hills	126515	5	2
6	Aguanga	9732	4	1
7	Ahwahnee	7872	4	1
8	Alameda	290204	8	0
9	Alamo	60748	4	0
10	Albany	63528	4	2
11	Albion	4216	4	0
12	Alderpoint	1044	4	1
13	Alhambra	370703	9	3
14	Aliso Viejo	164948	4	0
15	Alleghany	472	4	0
16	Alpaugh	4216	4	3

- Ranking agents based on Average satisfaction rating
 

```
SELECT c.Agent,
ROUND(AVG(CAST(c.Satisfaction_rating AS float)), 3) Satisfaction_Rating
FROM Call c
GROUP BY c.Agent
ORDER BY Satisfaction_Rating DESC
```

	Agent	Satisfaction_Rating
1	Martha	3.471
2	Dan	3.447
3	Diane	3.405
4	Greg	3.404
5	Stewart	3.4
6	Jim	3.394
7	Becky	3.371
8	Joe	3.331

- Ranking payment methods based on total revenue.

```
SELECT c.Payment_Method ,SUM(c.Total_Revenue) AS 'Total Revenue'
FROM Customer c
GROUP BY c.Payment_Method
ORDER BY SUM(c.Total_Revenue) DESC
```

	Payment_Method	Total Revenue
1	Bank Withdrawal	12672091.3
2	Credit Card	8157186.15999999
3	Mailed Check	541854.23

- Ranking cities based total revenue , phoneservice charges,internet monthly GB.

```
SELECT L.City,SUM(c.Total_Revenue)AS'Total Revenue',SUM(ps.Total_Long_Distance_Charges)
AS'Phone Service Total Charges'
,SUM(ci.Avg_Monthly_GB)as'AvgMonthly GB'
FROM Customer c,Customer_Internet ci,Phone_Service ps,Location l
WHERE L.Location_ID=c.Location_ID AND c.Customer_ID=ci.Customer_ID AND
c.Customer_ID=ps.Customer_ID
GROUP BY L.City
```

	City	Total Revenue	Phone Service Total Charges	AvgMonthly GB
1	Angelus Oaks	744.72	95.310	111
2	Daly City	18755.84	2370.640	156
3	Alviso	3979.35	1102.950	116
4	Santa Rosa	45757.73	12401.300	610
5	Indio	34048.74	6621.360	181
6	Lafayette	11405.31	4853.010	148
7	Posey	9217.3	2806.300	112
8	Port Hueneme	9467.09	2090.560	16
9	Ocotillo	15191.79	5370.840	54
10	Fremont	46102.46	12540.070	363
11	Palo Alto	41177.51	10376.350	279
12	El Centro	15315.31	3347.260	109
13	Platina	4372.41	2217.060	32
14	Wishon	10440.82	3920.120	50
15	Pismo Beach	2378.12	422.620	133
16	Sugarloaf	10620.41	3909.510	122

- **Ranking contract based on number of customers**

```
SELECT c.Contract_Type,COUNT(c1.Customer_ID)AS 'NumOfCustomers'
FROM Contract c,Customer c1
WHERE c.Contract_ID=c1.Contract_ID
GROUP BY c.Contract_Type
```

	Contract_Type	NumOfCustomers
1	Month-to-Month	3610
2	One Year	1550
3	Two Year	1883

- **Ranking calls topics**

```
SELECT c.Topic ,COUNT(c.Call_ID) AS NumOfCalls
FROM Call c
GROUP BY c.Topic
ORDER by NumOfCalls desc
```

	Topic	NumOfCalls
1	Streaming	1022
2	Technical Support	1019
3	Payment related	1007
4	Contract related	976
5	Admin Support	976

- **Total calls,problem solved,num of droped calls by agent,(proc)**

```
CREATE PROC agent @ag VARCHAR(50)
AS
SELECT c.Agent,SUM(CAST(c.Resolved AS INT)) AS 'problems solved',SUM(CAST(c.Answered AS INT)) AS 'NumOfCalls'
FROM Call c
WHERE c.Agent=@ag
GROUP BY c.Agent
```

	Agent	problems solved	NumOfCalls
1	Diane	452	501

- **Total Number customer and average duration joined for all internet type**

```
SELECT I.Internet_Type,count(c.Customer_ID) as CustNum,avg(c.Tenure_In_Months)as avgDuration
from Customer c,Customer_Internet ci,Internet_Service I
where c.Customer_ID=ci.Customer_ID and i.Internet_Service_ID=ci.Internet_Service_ID
GROUP BY I.Internet_Type
ORDER by CustNum desc
```

	Internet_Type	CustNum	avgDuration
1	Fiber Optic	3035	33
2	DSL	1652	32
3	No Internet	1526	30
4	Cable	830	31

- set categories or classes for customers based on how many services they use, starting from class A (highest number of services) to class D (lowest)

```

CREATE VIEW customer_class AS
SELECT Customer_ID, NUMBER_OF_SERVICES,
CASE
    WHEN NUMBER_OF_SERVICES > 6 THEN 'A'
    WHEN NUMBER_OF_SERVICES > 4 AND NUMBER_OF_SERVICES <=6 THEN 'B'
    WHEN NUMBER_OF_SERVICES > 2 AND NUMBER_OF_SERVICES <=4 THEN 'C'
    WHEN NUMBER_OF_SERVICES > 0 AND NUMBER_OF_SERVICES <=2 THEN 'D'
    WHEN NUMBER_OF_SERVICES = 0 THEN 'Z'
END AS CLASS
FROM (SELECT cust.Customer_ID,
CAST(custint.Premium_Tsupport AS int)+CAST(custint.Online_Backup AS
int)+CAST(custint.Online_Security AS int)+CAST(custint.Premium_Tsupport AS int)+
CAST(custint.Streaming_Movies AS int)+CAST(custint.Streaming_Music AS
int)+CAST(custint.Streaming_TV AS int)+CAST(custint.Unlimited_Data AS int) AS
NUMBER_OF_SERVICES
FROM Customer cust, Customer_Internet custint
WHERE cust.Customer_ID = custint.Customer_ID) a

```

	Customer_ID	NUMBER_OF_SERVICES	CLASS
1	1	5	B
2	2	2	D
3	3	1	D
4	4	4	C
5	5	4	C
6	6	6	B
7	7	8	A
8	8	3	C
9	9	2	D
10	10	8	A
11	11	5	B
12	12	8	A
13	13	1	D
14	14	4	C
15	15	5	B
16	16	4	C

- Trigger on customer table if any update done record at in audit table

```

CREATE TABLE [dbo].[Customer_Audit](
    [Customer_ID] [int] NULL,
    [ColumnName] [varchar](50) NULL,
    [OldValue] [varchar](max) NULL,
    [NewValue] [varchar](max) NULL,
    [UpdatedBy] [varchar](500) NULL,
    [UpdatedAt] [datetime] NULL)

go

CREATE OR alter TRIGGER CustomerTable_Audit
ON [dbo].[Customer]
AFTER UPDATE
AS
BEGIN
    DECLARE @customerID INT, @oldStatusID INT, @newStatusID INT, @oldContractID INT,
    @newContractID INT;

    SELECT
        @customerID = Customer_ID,
        @newStatusID = Status_ID,
        @newContractID = Contract_ID
    FROM inserted
    SELECT
        @oldStatusID = Status_ID,
        @oldcontractID = Contract_ID
    FROM deleted
    IF @customerID IS NOT NULL
    BEGIN

        IF UPDATE(Status_ID)
        BEGIN
            INSERT INTO [dbo].[Customer_Audit] (Customer_ID, ColumnName, OldValue,
NewValue, UpdatedBy, UpdatedAt)
                VALUES (@customerID, 'Status_ID', @oldStatusID , @newStatusID ,
SUSER_SNAME(), GETDATE());
        END

        IF UPDATE(Contract_ID)
        BEGIN
            INSERT INTO [dbo].[Customer_Audit] (Customer_ID, ColumnName, OldValue,
NewValue, UpdatedBy, UpdatedAt)
                VALUES (@customerID, 'Contract_ID', @oldContractID , @newContractID ,
SUSER_SNAME(), GETDATE());
        END
    END
END

```

At The End

# Thank you



