



TELECOMMUNICATION SERVICES MANAGEMENT SYSTEM

AISHWARYA SAWANT

ESHANEE THAKUR

GNANA TEJ CHERUKURU

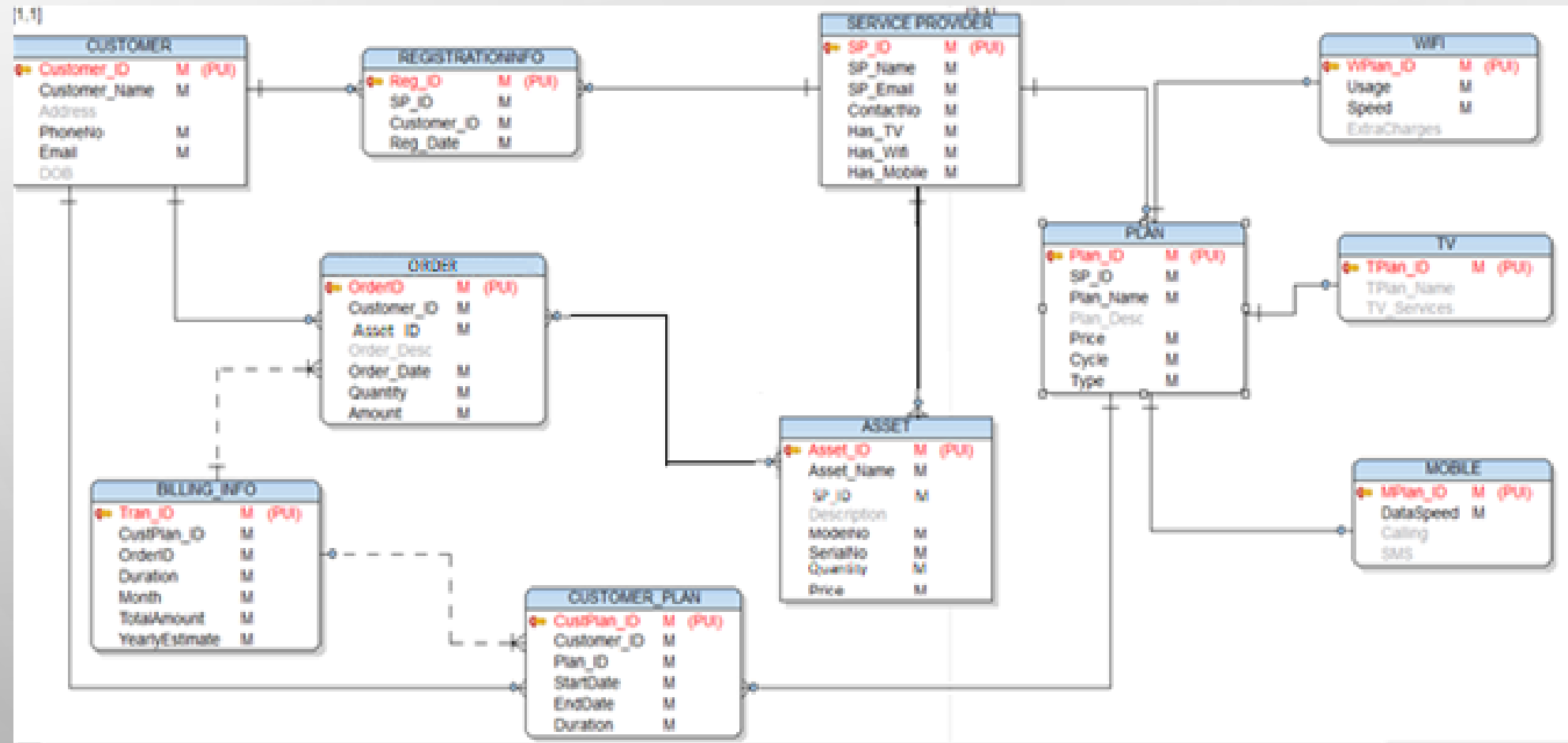
OMKAR KALANGE

RITESH PENDURKAR

SUMMARY

- To provide a consolidated view of billing of all the utilities the customer has subscribed
- Enabling the user to analyze the previous months spends for these subscribed services
- Enabling the service providers to analyze the subscriptions for the current plans
- To help determine the service provider the interests and current demands to enhance the current plans and propose more convenient plans for the customers

FINAL ERD



SQL DDL STATEMENTS

```
CREATE TABLE Customer
(
    Customer_ID bigint not null identity(10000000,1),
    Customer_fname varchar (100) not null,
    Customer_Lname varchar (100) not null,
    Address_line1 varchar (225) not null,
    City varchar (100) not null,
    [State] varchar (2) not null,
    zipcode int not null,
    phone_number VARCHAR(20),
    email varchar(225) not null,
    Constraint email check ( email like '%@%.%'),
    DateofBirth date,
    constraint customr_pk primary key (Customer_Id)
);
GO
```

```
CREATE TABLE ServiceProvider
(
    Sp_id int not null identity(3000,1),
    Sp_Name varchar(50) not null ,
    Sp_Email varchar(50) not null,
    Sp_Contact varchar(50) not null,
    CONSTRAINT PK_ServiceProvider_Sp_id PRIMARY KEY(Sp_id),
);
GO
```

```
CREATE TABLE Registrationinfo
(
    Reg_ID bigint not null identity(10000000,1),
    SP_ID INT not null,
    Customer_id bigint not null,
    reg_date date,
    Constraint PK_RegistrationInfo_ID primary key (Reg_id),
    Constraint FK_RegistrationInfo_Customer_ID foreign key(Customer_id) references customer(customer_id),
    Constraint FK_RegistrationInfo_SP_ID foreign key(SP_ID) references ServiceProvider(SP_ID)
);
GO
```

```

Create table [Plan]
(
    Plan_id int not null identity(1,1),
    Sp_id int not null,
    Plan_Name varchar(20),
    Plan_Desc varchar(50),
    Price decimal(6,2),
    Cycle varchar(20),
    Has_Tv bit not null,
    Has_Wifi bit not null,
    Has_Mobile bit not null,
    CONSTRAINT CHK_Cycle CHECK (Cycle in ('Yearly','Half-Yearly','Quarterly','Monthly')),
    CONSTRAINT PK_Plan_Plan_id PRIMARY KEY(Plan_id),
    CONSTRAINT FK_Plan_Sp_id FOREIGN KEY(Sp_id) REFERENCES ServiceProvider(Sp_id)
);
GO

```

```

Create table Mobile
(
    MPlan_id int not null,
    DataSpeed decimal(6,2),
    Calling varchar(20),
    SMS varchar(20),
    CONSTRAINT PK_Mobile_MPlan_id PRIMARY KEY(MPlan_id),
    CONSTRAINT FK_Mobile_MPlan_id FOREIGN KEY(MPlan_id) REFERENCES [Plan](Plan_id)
);
GO

```

```

CREATE TABLE [Customer_Plan]
(
    CustPlan_ID Integer Not Null identity(1,1),
    Customer_ID BIGINT,
    Plan_ID Integer,
    [Start Date] Date,
    [End Date] DATE,

    CONSTRAINT PK_CustPlan_ID PRIMARY KEY (CustPlan_ID),
    CONSTRAINT FK_CustPlan_Customer_ID FOREIGN KEY (Customer_ID) REFERENCES CUSTOMER (Customer_ID),
    CONSTRAINT FK_CustPlan_Plan_ID FOREIGN KEY (Plan_ID) REFERENCES [PLAN] (Plan_ID)
)
GO

```

```

Create table Wifi
(
    WPlan_id int not null,
    Usage decimal(6,2),
    speed decimal (6,2),
    Extra_charge decimal (6,2),
    CONSTRAINT PK_Wifi_WPlan_id PRIMARY KEY(WPlan_id),
    CONSTRAINT FK_Wifi_WPlan_id FOREIGN KEY(WPlan_id) REFERENCES [Plan](Plan_id)
);
GO

```

```

Create table Television
(
    TPlan_id int not null,
    TName VARCHAR(40),
    Tv_Services varchar(50),
    CONSTRAINT PK_TV_TPlan_id PRIMARY KEY(TPlan_id),
    CONSTRAINT FK_TV_TPlan_id FOREIGN KEY(TPlan_id) REFERENCES [Plan](Plan_id)
);
GO

```

```

CREATE TABLE [Asset]
(
    Asset_ID Integer Not Null identity(100,1),
    Asset_Name varchar(20),
    SP_ID Integer,
    [Description] varchar(50),
    ModelNo Integer,
    SerialNo Integer,
    Quantity Integer,
    Price Decimal(6,2),
    CONSTRAINT PK_Asset_ID PRIMARY KEY (Asset_ID),
    CONSTRAINT FK_Asset_SP_ID FOREIGN KEY (SP_ID) REFERENCES ServiceProvider (SP_ID)
)
GO

```

```

CREATE TABLE dbo.[ORDER]
(
    Order_ID BIGINT NOT NULL IDENTITY(100000,1)
    ,Customer_ID BIGINT NOT NULL
    ,Asset_ID INT NOT NULL
    ,Order_Desc VARCHAR(200)
    ,Order_Date DATE NOT NULL
    ,Quantity INT NOT NULL
    ,Amount AS dbo.CalculatePrice(Asset_ID,Quantity)
    CONSTRAINT PK_Order_Order_ID PRIMARY KEY (Order_ID)
    ,CONSTRAINT FK_Order_Customer_ID FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
    ,CONSTRAINT FK_Order_Asset_ID FOREIGN KEY (Asset_ID) REFERENCES Asset(Asset_ID)
)
GO

```

```

CREATE TABLE dbo.BillingInfo
(
    Transaction_ID BIGINT NOT NULL IDENTITY(500000,1)
    ,Customer_ID BIGINT NOT NULL
    ,[Month] INT NOT NULL
    ,[Year] INT NOT NULL
    ,TotalAmount AS CONVERT(DECIMAL(8,3),dbo.CalculateTotalAmount(Customer_ID,[Month],[Year]))
    ,YearlyEstimate AS (dbo.CalculateTotalAmount(Customer_ID,[Month],[Year])*12)
    CONSTRAINT PK_BillingInfo_Transaction_ID PRIMARY KEY (Transaction_ID)
    ,CONSTRAINT FK_BillingInfo_Customer_ID FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
)
GO

```

```

CREATE TABLE [dbo].[Customer_Audit](

    [Customer_AuditID] [bigint] primary key IDENTITY(1,1) NOT NULL,
    [Customer_ID] [bigint] NOT NULL,
    [Customer_fname] [varchar](100) NOT NULL,
    [Customer_Lname] [varchar](100) NOT NULL,
    [Address_line1] [varchar](225) NOT NULL,
    [City] [varchar](100) NOT NULL,
    [State] [varchar](2) NOT NULL,
    [zipcode] [int] NOT NULL,
    [phone_number] [varchar](20) NULL,
    [email] [varchar](225) NOT NULL,
    [DateofBirth] [date] NULL,
    Action char(1),
    ActionDate datetime);

GO

CREATE TABLE [dbo].[Customer_Plan_Audit](
    [CustomerPlanAudit_ID] [int] IDENTITY(1,1) NOT NULL primary key,
    [CustPlan_ID] [int] NOT NULL,
    [Customer_ID] [bigint] NULL,
    [Plan_ID] [int] NULL,
    [Start Date] [date] NULL,
    [End Date] [date] NULL,
    [Action] [char](1) NULL,
    [ActionDate] [datetime] NULL
)

GO

```

```

-----COLUMN ENCRYPTION-----
--CREATE A MASTER KEY FOR THE DATABASE
CREATE MASTER KEY ENCRYPTION BY
PASSWORD = 'Email'
GO

--CREATE A SECURITY CERTIFICATE
CREATE CERTIFICATE SelfSignedCertificate
WITH SUBJECT = 'Email Encryption';
GO

--CREATING A KEY FOR ENCODING/DECODING THE COLUMNS
CREATE SYMMETRIC KEY SQLSymmetricKey
WITH ALGORITHM = AES_128
ENCRYPTION BY CERTIFICATE SelfSignedCertificate;
GO

--ADDING NEW COLUMN FOR ENCRYPTED FIELD
ALTER TABLE Customer
ADD EncryptedEmail VARBINARY(MAX) NULL
GO

```

DATABASE OBJECTS - VIEWS

```
CREATE VIEW dbo.VW_ServiceProviderPlan
AS
    SELECT
        SP.SP_ID, SP.SP_Name, SP_Email, SP.SP_Contact
        , P.Plan_Name, P.Cycle, P.Price
        , CONCAT(IIF(Has_Wifi = 1, 'WIFI, ', ''), IIF(Has_TV = 1, 'TV, ', ''), IIF(Has_Mobile = 1, 'Mobile', '')) AS ServicesIncluded
    FROM ServiceProvider SP
    INNER JOIN [Plan] P ON P.SP_ID = SP.SP_ID
GO

CREATE VIEW VW_CustomerPlan
AS
    SELECT
        c.Customer_ID, c.Customer_fname, c.Customer_Lname, c.email,
        SP.SP_Name, p.Plan_ID, p.Plan_Name, p.Price, CP.[Start Date], CP.[End date]
    FROM Customer c
    INNER JOIN Customer_Plan cp ON c.Customer_ID = cp.Customer_ID
    INNER JOIN [Plan] p ON cp.Plan_ID = p.Plan_ID
    INNER JOIN ServiceProvider sp ON p.SP_ID = sp.SP_ID
GO

CREATE VIEW VW_CustomerAssets
AS
    SELECT c.Customer_ID, c.Customer_fname, c.Customer_lname, a.Asset_Name
    FROM (Customer c
    INNER JOIN [Order] o ON c.Customer_ID = o.Customer_ID)
    INNER JOIN Asset a ON o.Asset_ID = a.Asset_ID;
GO
```


DATABASE OBJECTS - FUNCTIONS

```
CREATE FUNCTION dbo.CalculatePrice(@Asset_ID INT,@Quantity INT)
RETURNS DECIMAL(8,3)
AS
BEGIN
    DECLARE @finalResult DECIMAL(8,3) = 0

    SELECT @finalResult = @Quantity*A.Price
    FROM Asset A
    WHERE A.Asset_ID = @Asset_ID

    RETURN @finalResult
END
GO

CREATE FUNCTION dbo.GetMonthlyCharges(@Price DECIMAL(8,3), @Cycle VARCHAR(15))
RETURNS DECIMAL(8,3)
AS
BEGIN
    DECLARE @MonthlyCharge DECIMAL(8,3) = 0

    SELECT @MonthlyCharge =
        CASE WHEN @Cycle = 'YEARLY' THEN (@Price/12)
             WHEN @Cycle = 'HALF-YEARLY' THEN (@Price/6)
             WHEN @Cycle = 'QUARTERLY' THEN (@Price/3)
             WHEN @Cycle = 'MONTHLY' THEN (@Price)
        END

    RETURN @MonthlyCharge
END
GO
```

DATABASE OBJECTS - FUNCTIONS

```
CREATE FUNCTION dbo.CalculateTotalAmount(@Customer_ID BIGINT,@Month INT,@Year INT)
RETURNS DECIMAL(8,3)
AS
BEGIN
    DECLARE @TotalAmount DECIMAL(8,3)

    --Plan charges based on the month and cycle
    SELECT @TotalAmount = dbo.GetMonthlyCharges(P.Price, P.Cycle)
    FROM Customer_Plan CP
    INNER JOIN [Plan] P ON P.Plan_ID = CP.Plan_ID
    WHERE Customer_ID = @Customer_ID
        AND DATEFROMPARTS(@Year, @Month, '01') BETWEEN CP.[Start Date] AND CP.[End Date]

    RETURN @TotalAmount
END
GO
```

DATABASE OBJECTS – STORED PROCEDURES

```
CREATE PROCEDURE splendingServiceProviderByYear (
    @Year int,
    @Sp_id int output,
    @Sp_Name varchar(50) output,
    @Sp_Email varchar(50) output,
    @Sp_Contact varchar(50) output,
    @BillingYear int output,
    @Total decimal(8,3) output
)
AS
BEGIN
    SELECT TOP 1
        @Sp_id=sp.Sp_id,
        @Sp_Name=sp.Sp_Name,
        @Sp_Email = sp.Sp_Email,
        @Sp_Contact=sp.Sp_Contact,
        @BillingYear = b.[Year],
        @Total = SUM(b.TotalAmount)
    FROM ServiceProvider sp
    INNER JOIN [Plan] p ON sp.Sp_id = p.Sp_id
    INNER JOIN Customer_Plan cp ON p.Plan_id = cp.Plan_id
    INNER JOIN Billinginfo b ON cp.Customer_id = b.Customer_id
    WHERE [Year] = @Year
    GROUP BY sp.Sp_id, sp.Sp_Name, sp.Sp_Email, sp.Sp_Contact,b.[Year]
    ORDER BY SUM(b.TotalAmount) DESC

    SELECT @Sp_id AS ServiceProvider_ID,@Sp_Name ServiceProvider_Name,
        @Sp_Email ServiceProvider_Email,@Sp_Contact ServiceProvider_Contact,
        @BillingYear Billing_Year,@Total Total_Revenue

END
GO
```

```
CREATE PROCEDURE ServiceProviderSales(@spid Integer, @month Integer, @year Integer)
AS
BEGIN

    SELECT @spid, MONTH(cp.[Start Date]),YEAR(cp.[Start Date]),sum(p.Price) as Sales
    FROM Customer_Plan cp
    JOIN [Plan] p ON cp.Plan_ID=p.Plan_ID
    WHERE (MONTH(cp.[Start Date])=@month
        AND SP_ID=@spid
        AND YEAR(cp.[Start Date])=@year)
    GROUP BY SP_ID ,month(cp.[Start Date]),year(cp.[Start Date])

END
GO

CREATE PROCEDURE TotalCustomers (@SP_ID INT)
AS
BEGIN
    SELECT SP.Sp_id, SP.Sp_Name, COUNT(customer_id) AS TotalCustomers
    FROM Registrationinfo RI
    INNER JOIN ServiceProvider SP ON SP.Sp_id = RI.SP_ID
    WHERE RI.[SP_ID]=@SP_ID
    GROUP BY SP.Sp_id, SP.Sp_Name

END
GO
```

DATABASE OBJECTS – STORED PROCEDURES

```
CREATE PROCEDURE dbo.GenerateConsolidatedBill
( @Customer_ID INT,@FromDate DATE,@ToDate DATE)
AS
BEGIN
    DECLARE @OrderSum DECIMAL(8,3) = 0, @PlanSum DECIMAL(8,3) = 0

    CREATE TABLE #CustomerPlanMapping
    ( Customer_ID BIGINT
      ,SP_ID INT
      ,StartDate DATE
      ,EndDate DATE
      ,Plan_Price DECIMAL(8,3)
      ,Plan_Cycle VARCHAR(10)
      ,ServicesIncluded VARCHAR(20)
      ,MonthlyCharges DECIMAL(8,3) )

    --Customer plans list within the given time period
    --CASE 1 : StartDate and EndDate both are between FromDate and ToDate
    INSERT INTO #CustomerPlanMapping
    SELECT
        CP.Customer_ID, SP_ID
        ,CP.[Start Date]
        ,CP.[End date]
        ,P.Price, P.Cycle
        ,CONCAT(IIF(Has_Wifi = 1, 'WIFI, ', ''),IIF(Has_TV = 1, 'TV, ', ''),IIF(Has_Mobile = 1, 'Mobile', ''))
        ,dbo.GetMonthlyCharges(P.Price, P.Cycle)
    FROM Customer_Plan CP
    INNER JOIN [Plan] P ON P.Plan_ID = CP.Plan_ID
    WHERE Customer_ID = @Customer_ID
        AND (CP.[Start date] BETWEEN @FromDate AND @ToDate AND CP.[End date] BETWEEN @FromDate AND @ToDate)
    UNION
    --CASE 2 : FromDate and ToDate are both between StartDate and EndDate
    SELECT
        CP.Customer_ID, SP_ID
        ,@FromDate
        ,@ToDate
        ,P.Price, P.Cycle
        ,CONCAT(IIF(Has_Wifi = 1, 'WIFI, ', ''),IIF(Has_TV = 1, 'TV, ', ''),IIF(Has_Mobile = 1, 'Mobile', ''))
        ,dbo.GetMonthlyCharges(P.Price, P.Cycle)
    FROM Customer_Plan CP
    INNER JOIN [Plan] P ON P.Plan_ID = CP.Plan_ID
    WHERE Customer_ID = @Customer_ID
        AND (@FromDate BETWEEN CP.[Start Date] AND CP.[End Date] AND @ToDate BETWEEN CP.[Start Date] AND CP.[End Date])
    UNION
    --CASE 3 : StartDate FromDate EndDate ToDate
    SELECT
        CP.Customer_ID, SP_ID
        ,@FromDate
        ,CP.[End Date]
        ,P.Price, P.Cycle
        ,CONCAT(IIF(Has_Wifi = 1, 'WIFI, ', ''),IIF(Has_TV = 1, 'TV, ', ''),IIF(Has_Mobile = 1, 'Mobile', ''))
        ,dbo.GetMonthlyCharges(P.Price, P.Cycle)
    FROM Customer_Plan CP
    INNER JOIN [Plan] P ON P.Plan_ID = CP.Plan_ID
    WHERE Customer_ID = @Customer_ID
        AND (@FromDate BETWEEN CP.[Start Date] AND CP.[End Date] AND CP.[End date] BETWEEN @FromDate AND @ToDate)
    UNION
    --CASE 4 : FromDate StartDate ToDate EndDate
    SELECT
        CP.Customer_ID, SP_ID,CP.[Start Date],@ToDate,P.Price, P.Cycle
        ,CONCAT(IIF(Has_Wifi = 1, 'WIFI, ', ''),IIF(Has_TV = 1, 'TV, ', ''),IIF(Has_Mobile = 1, 'Mobile', ''))
        ,dbo.GetMonthlyCharges(P.Price, P.Cycle)
    FROM Customer_Plan CP
    INNER JOIN [Plan] P ON P.Plan_ID = CP.Plan_ID
    WHERE Customer_ID = @Customer_ID
        AND (CP.[Start Date] BETWEEN @FromDate AND @ToDate AND @ToDate BETWEEN CP.[Start Date] AND CP.[End Date])

    --Asset Owned by the customer
    SELECT *
    FROM [Order]
    WHERE Customer_ID = @Customer_ID AND Order_Date BETWEEN @FromDate AND @ToDate

    --Asset charges in case any exist
    SELECT @OrderSum = SUM(Amount)
    FROM [Order]
    WHERE Customer_ID = @Customer_ID AND Order_Date BETWEEN @FromDate AND @ToDate

    --Plan charges based on the month and cycle
    SELECT @PlanSum = SUM(CPM.MonthlyCharges*(DATEDIFF(MONTH,StartDate,EndDate))) FROM #CustomerPlanMapping CPM

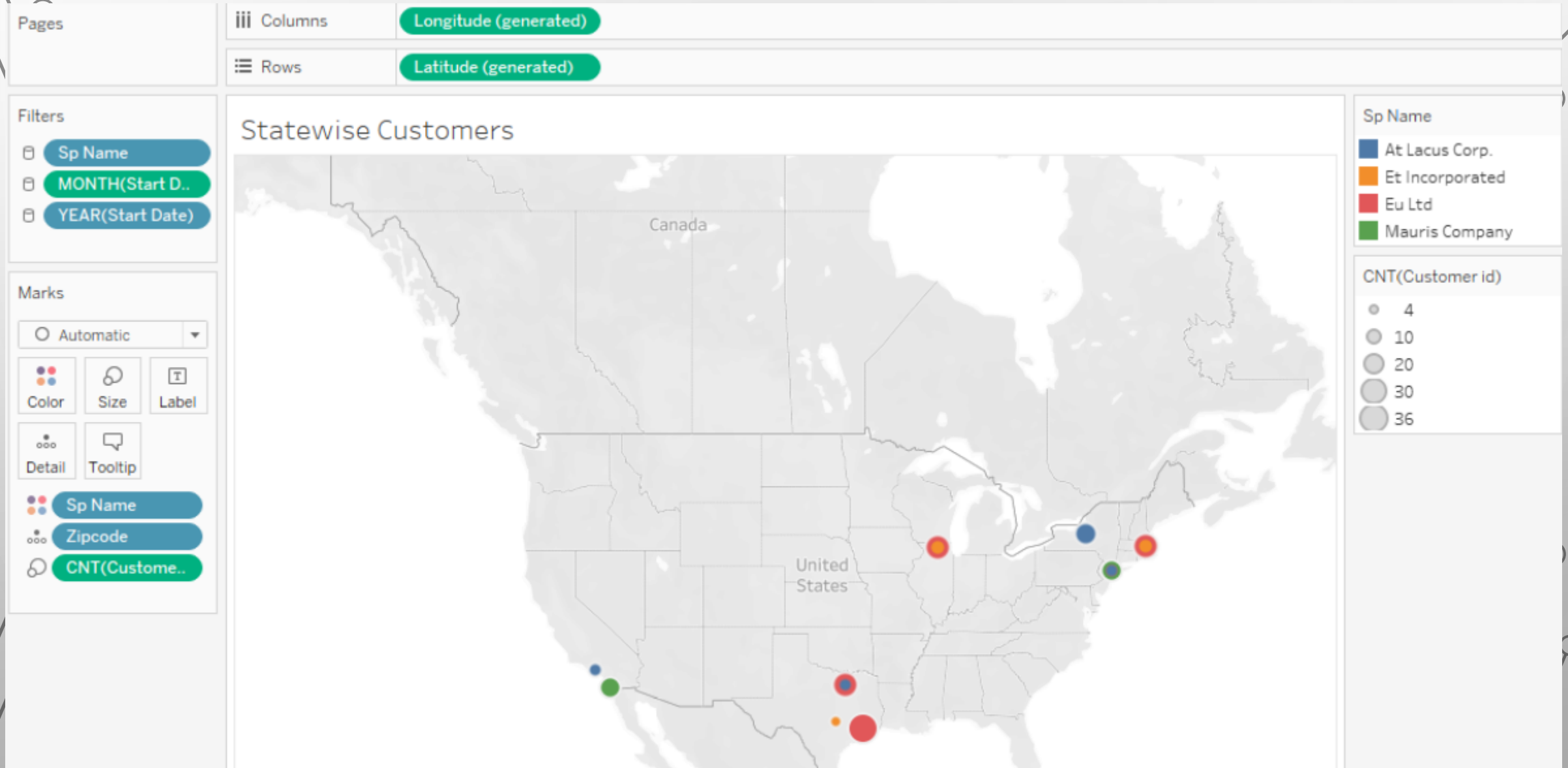
    --Total charges
    SELECT (ISNULL(@OrderSum,0) + ISNULL(@PlanSum,0)) AS Total_Charges
END
GO
```

DATABASE OBJECTS - TRIGGERS

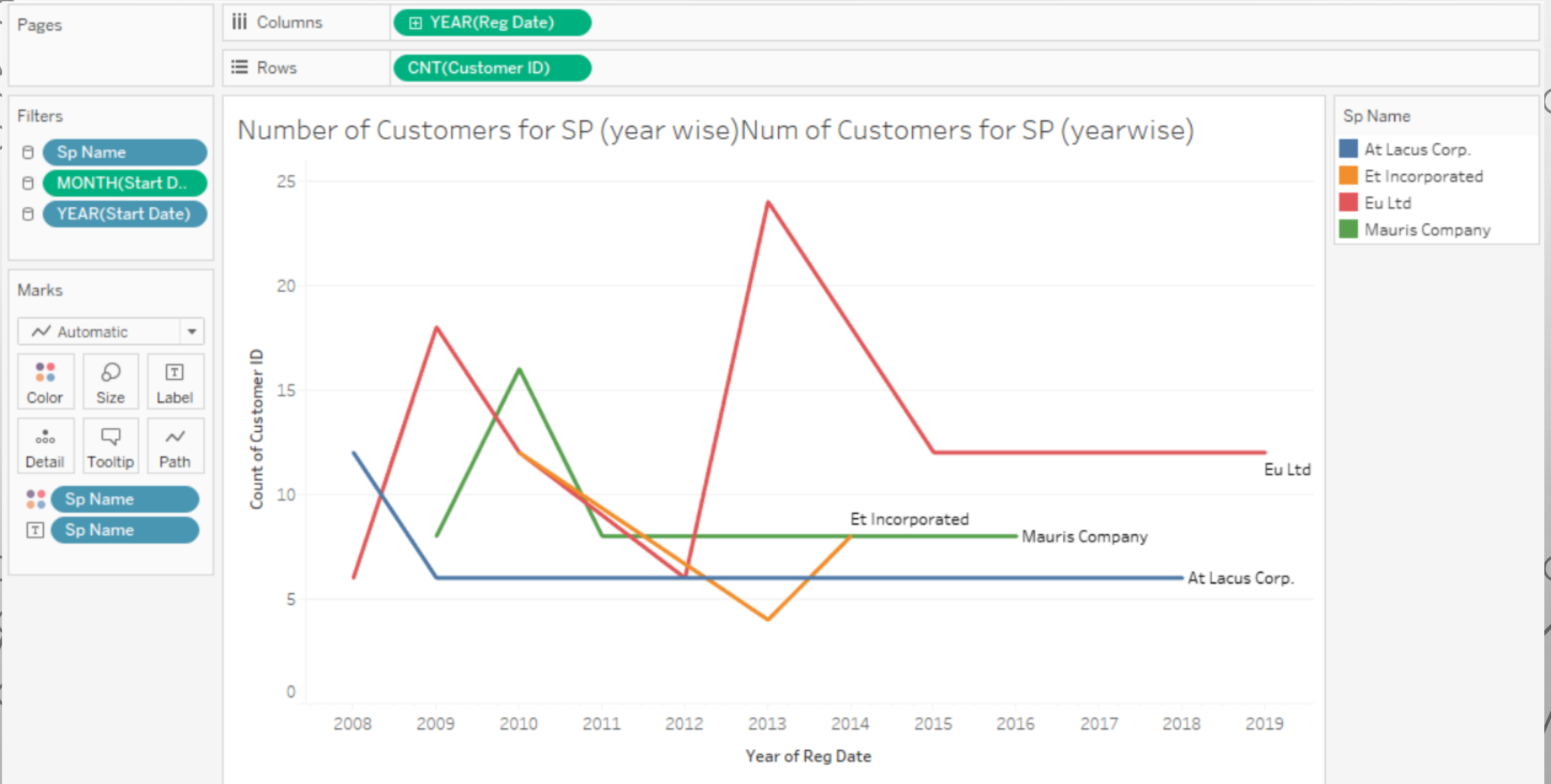
```
CREATE TRIGGER CustomerAudit ON customer FOR UPDATE,INSERT,DELETE
AS
    IF EXISTS(SELECT * FROM inserted) AND EXISTS (SELECT * FROM deleted)
    BEGIN
        INSERT INTO [Customer_Audit] (Customer_ID, Customer_fname, Customer_Lname, Address_line1,
        City, [State], [zipcode], [phone_number], [email], [DateofBirth], Action, ActionDate)
        SELECT [Customer_ID], [Customer_fname], [Customer_Lname], [Address_line1],
        [City], [State], [zipcode], [phone_number], [email], [DateofBirth], 'U', getdate() from deleted
    END
    IF EXISTS (SELECT * FROM INSERTED) AND NOT EXISTS(SELECT * FROM DELETED)
    BEGIN
        INSERT INTO [Customer_Audit] (Customer_ID, Customer_fname, Customer_Lname, Address_line1,
        City, [State], [zipcode], [phone_number], [email], [DateofBirth], Action, ActionDate)
        SELECT [Customer_ID], [Customer_fname], [Customer_Lname], [Address_line1],
        [City], [State], [zipcode], [phone_number], [email], [DateofBirth], 'I', getdate() from Inserted
    END
    IF EXISTS(SELECT * FROM DELETED) AND NOT EXISTS(SELECT * FROM INSERTED)
    BEGIN
        INSERT INTO [Customer_Audit] (Customer_ID, Customer_fname, Customer_Lname, Address_line1,
        City, [State], [zipcode], [phone_number], [email], [DateofBirth], Action, ActionDate)
        SELECT [Customer_ID], [Customer_fname], [Customer_Lname], [Address_line1],
        [City], [State], [zipcode], [phone_number], [email], [DateofBirth], 'D', getdate() from Deleted
    END
GO

CREATE TRIGGER CustomerPlanAudit ON Customer_Plan FOR UPDATE,INSERT,DELETE
AS
    IF EXISTS(SELECT * FROM INSERTED) AND EXISTS (SELECT * FROM DELETED)
    BEGIN
        INSERT INTO Customer_Plan_Audit( CustPlan_ID, Customer_ID, Plan_ID, [Start Date], [End Date], Action, ActionDate)
        SELECT CUSTPLAN_ID, CUSTOMER_ID, PLAN_ID, [START DATE], [END DATE], 'U', GETDATE() FROM DELETED
    END
    IF EXISTS (SELECT * FROM INSERTED) AND NOT EXISTS(SELECT * FROM DELETED)
    BEGIN
        INSERT INTO Customer_Plan_Audit( CustPlan_ID, Customer_ID, Plan_ID, [Start Date], [End Date], Action, ActionDate)
        SELECT CustPlan_ID, Customer_ID, Plan_ID, [Start Date], [End Date], 'I', getdate() from Inserted
    END
    IF EXISTS(SELECT * FROM DELETED) AND NOT EXISTS(SELECT * FROM INSERTED)
    BEGIN
        INSERT INTO Customer_Plan_Audit(CustPlan_ID, Customer_ID, Plan_ID, [Start Date], [End Date], Action, ActionDate)
        SELECT CustPlan_ID, Customer_ID, Plan_ID, [Start Date], [End Date], 'D', getdate() from Deleted
    END
GO
```

LOCATION WISE CUSTOMERS

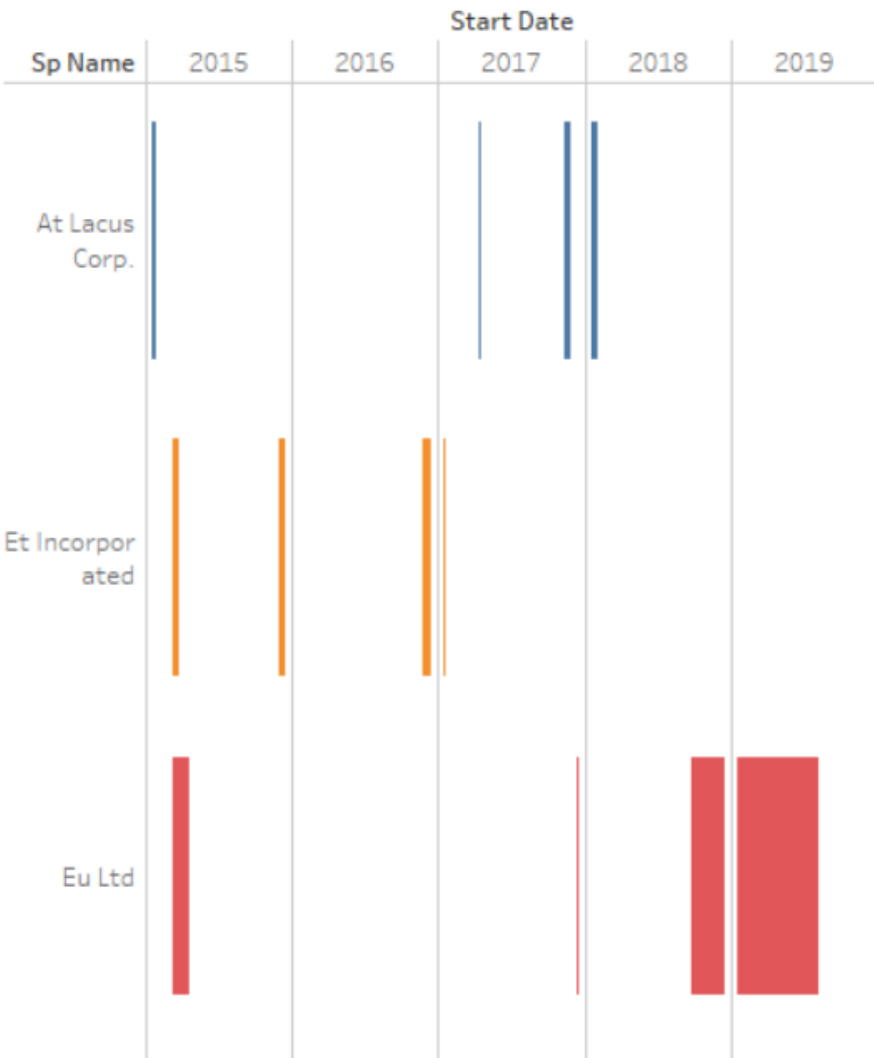


YEAR WISE CUSTOMERS FOR SERVICE PROVIDERS

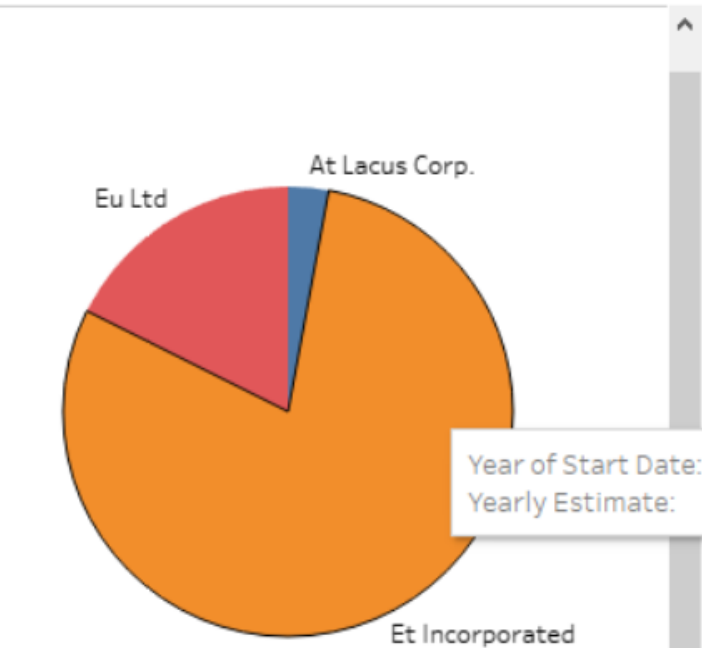


DASHBOARD

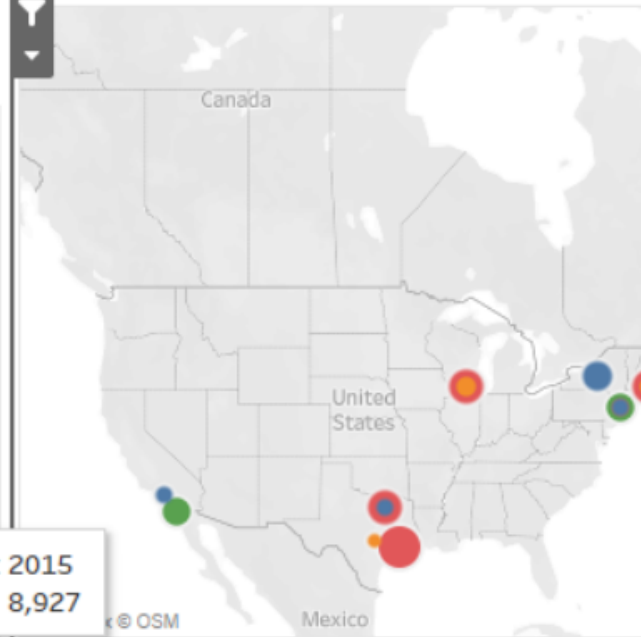
Year-wise Customer Count for Service provider



Service Provider and yearly revenue



Statewise Customers



Month of S.. January 2011 November 2019

Sp Name
(Multiple values)



THANK YOU!