# Milestone 2
# “Description”
# Deadline for Milestone 2 is on 22/11/2024 at 11:59

1. Important Guidelines:
    - **The Milestone description is subject to minor changes that would be announced on CMS. Do check the CMS periodically.**
    - All alpha-numeric attributes should be defined with type varchar(50)
    - All numeric values should be defined with type int or decimal
    - You must follow the names of the required functions, stored procedures and views. **Not matching the names will result in grade loss**
    - You must follow the order and data types of the parameters required for a stored procedure or function with a specific order and certain data type. Failing to follow this guideline will result in grade loss of that function/stored procedure/view
    - You must follow the order of the needed columns in case a table is required to be returned from a function or fetched from a view
    - You should do the appropriate join if you are required to implement a function/view that fetches data from different tables in your database
    - You should insert the data in the appropriate tables in your database if the required stored

procedure inserts data in multiple tables in your database

- Always think **about all consequences** and required actions to be done in every requirement
- The type of any Account can either be Post Paid, Prepaid, or Pay_as_you_go.
- The Status of any Account can either be ‘active’, or ‘onhold’
- The status of Subscription of the Service Plan can either be ‘active’ or ‘onhold’.
- The status of any Payment can either be successful, pending or rejected.
- The status of any Benefits can either be active or expired.
- The status of any Technical support Ticket can either be Open, In Progress, or Resolved.
- The method of any Payment can either be ‘cash’, or ‘credit’.
- Points in table customer_Account should have a default value of 0.
- All IDs in the project are identity(1,1) except for the mobileNo and nationalID
- Cashback is calculated as 10% of the payment amount.
- **Any compilation/syntax error in any of the requirements will result in grade loss**

# 2. Requirements:

**You are required to implement the following:**

## 2.1 Basic Structure of the Database:

a) Write an SQL query to create database called "Telecom_Team_Your Team Number"

b) Put the queries that create all the tables of your database with their definition inside this procedure.

   i) Type: stored procedure
   ii) Name: createAllTables
   iii) Input:Nothing
   iv) output:Nothing

c) Drop all tables that your database has inside this procedure.

   i) Type: stored procedure
   ii) Name: dropAllTables
   iii) Input:Nothing
   iv) output:Nothing

d) Drop all implemented stored procedures (except this one), functions and views that you implemented in this milestone.

   i) Type: stored procedure
   ii) Name: dropAllProceduresFunctionsViews
   iii) Input:Nothing
   iv) output:Nothing

e) Clear all records in all tables existing in your database.
   i) Type: stored procedure
   ii) Name: clearAllTables
   iii) Input:Nothing
   iv) output:Nothing

## 2.2 Basic Data Retrieval

a) Fetch details for all customer profiles along with their active accounts.
   i) Type: view
   ii) Name: allCustomerAccounts
   iii) Input:Nothing
   iv) Output: Table

b) Fetch details for all offered Service Plans.
   i) Type: view
   ii) Name: allServicePlans
   iii) Input: Nothing
   iv) Output: Table

c) Fetch details for all active Benefits.
   i) Type: view
   ii) Name: allBenefits
   iii) Input:Nothing
   iv) Output: Table

d) Fetch details for all payments along with their corresponding Accounts.
   i) Type: view
   ii) Name: AccountPayments
   iii) Input: Nothing
   iv) Output: Table

e) Fetch details for all shops.

i) Type: view
ii) Name: allShops
iii) Input:Nothing
iv) Output: Table

f) Fetch details for all resolved tickets.

i) Type: view
ii) Name: allResolvedTickets
iii) Input:Nothing
iv) Output: Table

g) Fetch details of all wallets along with their customer names.

i) Type: view
ii) Name: CustomerWallet
iii) Input:Nothing
iv) Output: Table

h) Fetch the list of all E-shops along with their redeemed vouchers's ids and values.

i) Type: view
ii) Name: E_shopVouchers
iii) Input:Nothing
iv) Output: Table

i) Fetch the list of all physical stores along with their redeemed vouchers's ids and values.

i) Type: view
ii) Name: PhysicalStoreVouchers
iii) Input:Nothing
iv) Output: Table

j) Fetch number of cashback transactions per each wallet.
i) Type: view
ii) Name: Num_of_cashback
iii) Input:Nothing
iv) Output: Table

## 2.3 As an admin I should be able to:

a) List all accounts along with the service plans they are subscribed to.
   i) Name: Account_Plan
   ii) Type: Stored Procedure
   iii) Input: Nothing
   iv) output : Table

b) Retrieve the list of accounts subscribed to the input plan on a certain date.
   i) Name: Account_Plan_date
   ii) Type: Table Valued Function
   iii) Input: Subscription_Date date and Plan_id int
   iv) output : Table includes (Account mobile number, Service plan ID and name)

c) Retrieve the total usage of the input account on each subscribed plan from a given input date.
   i) Name: Account_Usage_Plan
   ii) Type: Table Valued Function
   iii) Input: MobileNo char(11), from_date date
   iv) output : Table includes (planID, total data consumed, total minutes used, and total SMS).

d) Delete all benefits offered to the input account for a certain plan.
   i) Name: Benefits_Account
   ii) Type: Stored Procedure
   iii) Input: MobileNo char(11), plan ID int
   iv) output : Table

e) Retrieve the list of gained offers of type ‘SMS’ for the input account.
   i) Name: Account_SMS_Offers
   ii) Type: Table Valued Function
   iii) Input: MobileNo char(11)
   iv) output : Table

f) Retrieve the number of accepted payment transactions initiated by the input account during the last year along with the total amount of earned points.
   i) Name: Account_Payment_Points
   ii) Type: Stored Procedure
   iii) Input: MobileNo char(11)
   iv) Output: Total Number of transactions, Total Amount of points

g) Retrieve the amount of cashback returned on the input wallet based on a certain plan.
   i) Name: Wallet_Cashback_Amount
   ii) Type: Function
   iii) Input: WalletId int, planId int
   iv) output : Amount of cashback

h) Retrieve the average of the sent transaction amounts from the input wallet within a certain duration.

i) Name: Wallet_Transfer_Amount
ii) Type: Function
iii) Input: Wallet_id int, start_date date, end_date date
iv) output : Transaction amount average

i) Take mobileNo as an input, return true if this number is linked to a wallet, otherwise return false.

i) Name: Wallet_MobileNo
ii) Type: Function
iii) Input: MobileNo char(11)
iv) output : bit

j) Update the total number of points that the input account should have.

i) Name: Total_Points_Account
ii) Type: Stored Procedure
iii) Input: MobileNo char(11)
iv) Output: Total points

## 2.4 As a customer I should be able to:

a) login using my mobileNo and password.

i) Name: AccountLoginValidation
ii) Type: Function
iii) Input: MobileNo char(11), password varchar(50)
iv) Output: Success bit

b) Retrieve the total SMS, Mins and Internet consumption for an input plan within a certain duration.

   i) Name: Consumption
   ii) Type: Table Valued Function
   iii) Input: Plan_name varchar(50), start_date date, end_date date
   iv) Output: Table includes (Data consumption, Minutes used, SMS sent )

c) Retrieve all offered plans that the input customer is not subscribed to.

   i) Name: Unsubscribed_Plans
   ii) Type : Stored Procedure
   iii) Input: MobileNo char(11)
   iv) Output: Table

d) Retrieve the usage of all active plans for the input account in the current month.

   i) Name: Usage_Plan_CurrentMonth
   ii) Type: Table Valued Function
   iii) Input: MobileNo char(11)
   iv) Output: Table includes (Data consumption, Minutes used, SMS sent)

e) Retrieve all cashback transactions related to the wallet of the input customer.

i) Name: Cashback_Wallet_Customer
ii) Type: Table Valued Function
iii) Input: NationalID int
iv) Output: Table

f) Retrieve the number of technical support tickets that are **NOT** ‘Resolved’ for each account of the input customer.

i) Name: Ticket_Account_Customer
ii) Type: Stored Procedure
iii) Input: NationalID int
iv) Output: Number of technical support tickets

g) Return the voucher with the highest value for the input account.

i) Name: Account_Highest_Voucher
ii) Type: Stored Procedure
iii) Input: MobileNo char(11)
iv) Output: Voucher_id int

h) Get the remaining amount for a certain plan based on the payment initiated by the input account.

i) Name: Remaining_plan_amount
ii) Type: Function
iii) Input: MobileNo char(11), plan_name varchar(50)
iv) Output: Remaining amount

i) Get the extra amount from a payment initiated by the input account for a certain plan.

i) Name: Extra_plan_amount
ii) Type: Function
iii) Input: MobileNo char(11), plan_name varchar(50)
iv) Output: Extra amount

j) Retrieve the top 10 successful payments with highest value for the input account.

i) Name: Top_Successful_Payments
ii) Type: Stored Procedure
iii) Input: MobileNo char(11)
iv) Output: Table

k) Retrieve all service plans the input account subscribed to in the past 5 months.

i) Name: Subscribed_plans_5_Months
ii) Type: Table Valued Function
iii) Input: MobileNo char(11)
iv) Output: Table

l) Initiate an accepted payment for the input account for plan renewal and update the status of the subscription accordingly.

i) Name: Initiate_plan_payment
ii) Type: Stored Procedure
iii) Input: MobileNo char(11) ,amount decimal(10,1), payment_method varchar(50),plan_id int
iv) Output: Nothing

m) Calculate the amount of cashback that will be returned on the wallet of the customer of the input account from a certain payment for the specified benefit and update the wallet’s balance accordingly.

i) Name: Payment_wallet_cashback
ii) Type: Stored Procedure
iii) Input: MobileNo char(11), payment_id int, benefit_id int
iv) Output: Nothing

n) Initiate an accepted payment for the input account for balance recharge.

i) Name: Initiate_balance_payment
ii) Type: Stored Procedure
iii) Input: MobileNo char(11) ,amount decimal(10,1), payment_method varchar(50)
iv) Output: Nothing

o) Redeem a voucher for the input account and update the total points of the account accordingly.

i) Name: Redeem_voucher_points
ii) Type: Stored Procedure
iii) Input: MobileNo char(11), voucher_id int
iv) Output: Nothing