SHOP MANAGEMENT SYSTEM

ARRAFI AHSAN SOMUDRO

Contents:

1. Introduction:

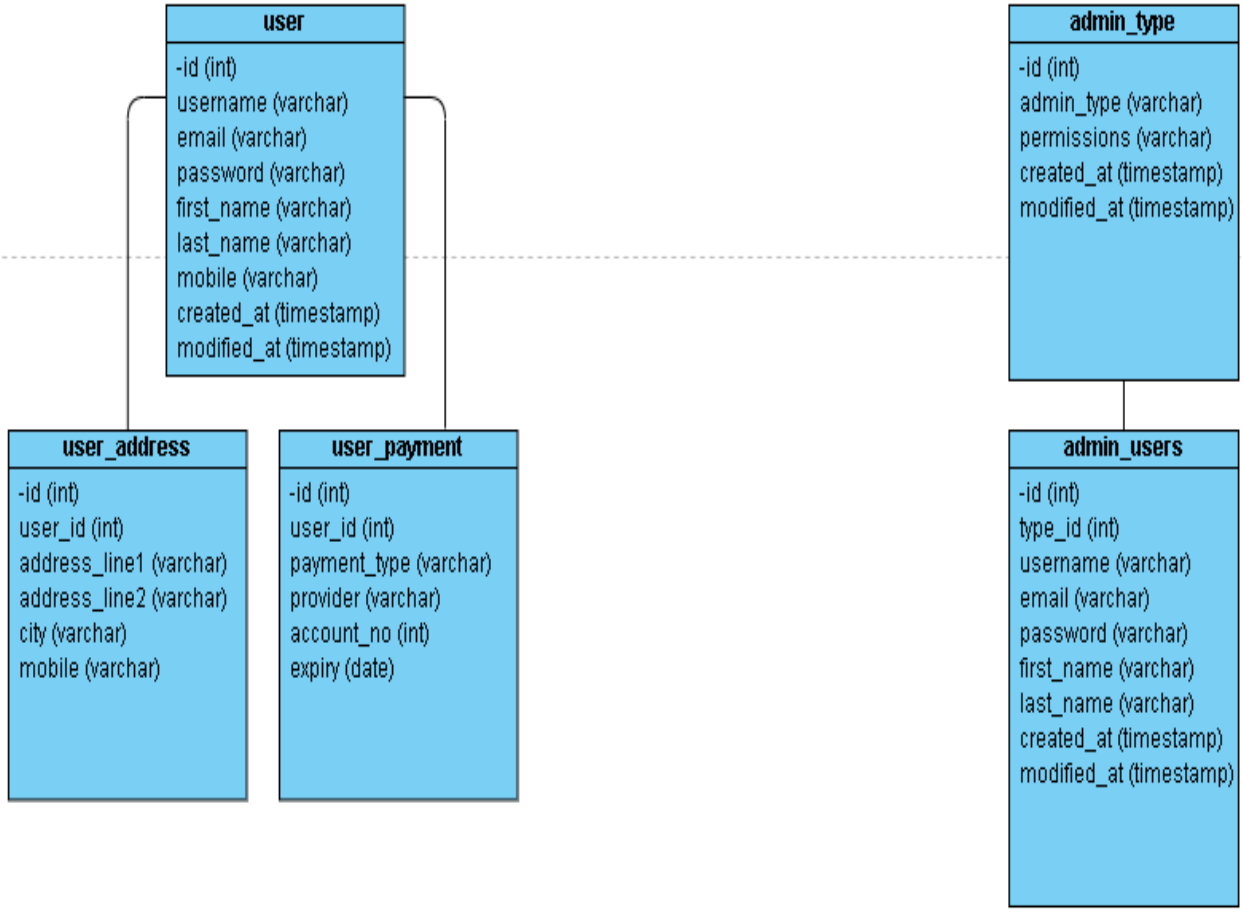
E-Commerce websites are online portals that facilitate online transactions of goods and services through means of the transfer of information and funds over the Internet. We all know the importance of e-commerce, especially in this pandemic situation. There are lots of e-commerce websites available such as Amazon, Daraz, E-valy and many more. But to handle the data of this type of giant e-commerce websites it is required to build a well-structured database system. A database system helps e-commerce sites pinpoint potential customers based on compiled information. Marketing teams can use customer data that is stored within the database to create targeted lists that will be used for directing marketing efforts. The more information a marketing team has, the better they can identify and tailor communications with them. Not only will this method help to retain customers, but also helps to gain new customers as well. The database also stores the information of employees and other stuffs of the company. In this modern era, we cannot even think any website or software working without a database. In this project we will build a database management system that will represent any other e-commerce websites currently available in the market.

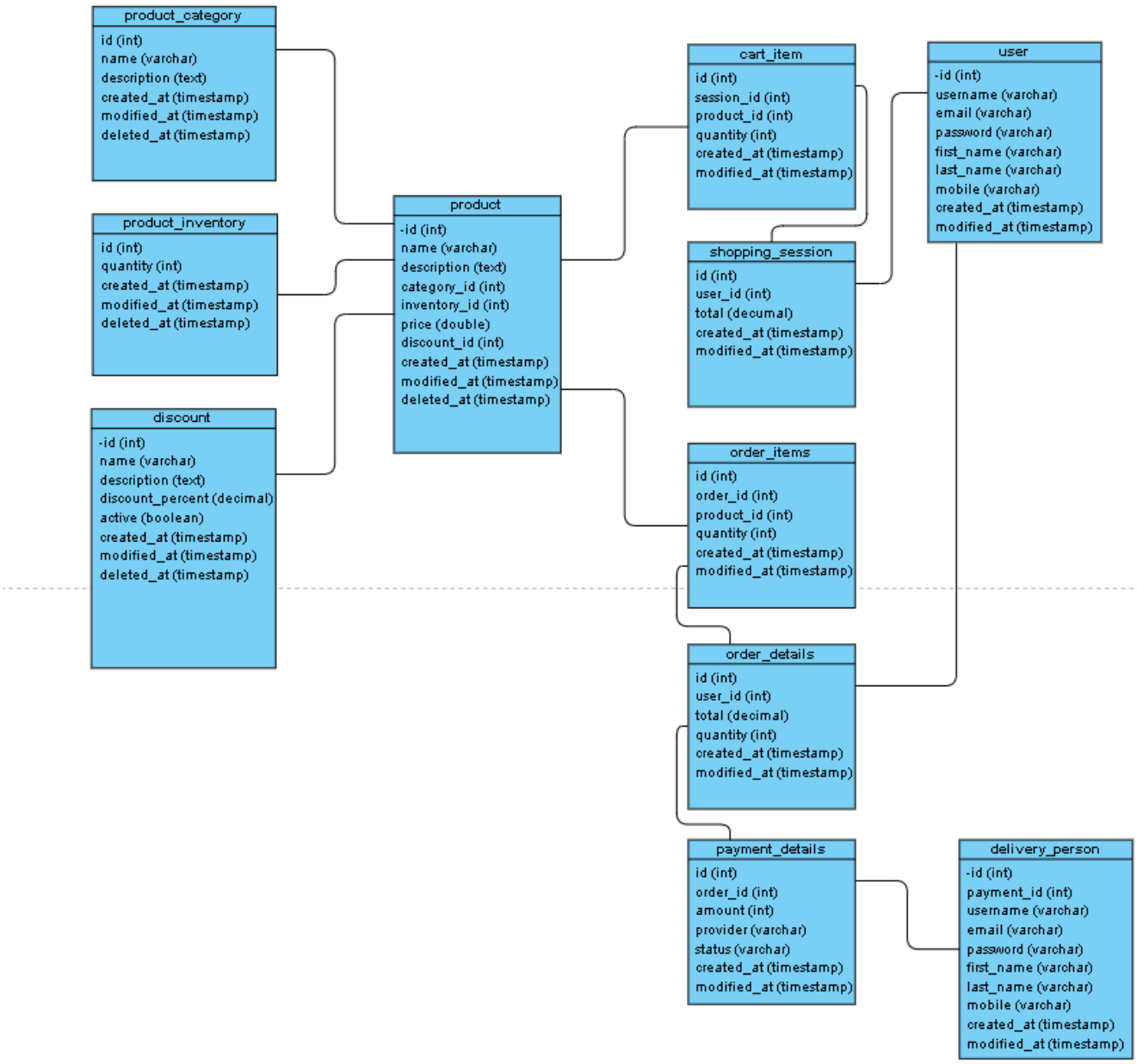
2. Project Proposal:

Our goal is to develop a standard database management system for an e-commerce website that can handle all the functionalities related to database. With this system we will be able to manage the database according to the need of the company. We will be able to handle the data of new customers registered to website, order of customers, data related to each order like delivery status, delivery time, delivery completion etc. We can also manage the data of employees and stuffs of the company such as details of employees, departments, accounts, and many more. Our project will help the authority to generate any report regarding profits, marketing, performance of employees, customer involvements, warehouse availabilities etc. A brief description of our database system is given in the scenario description section.

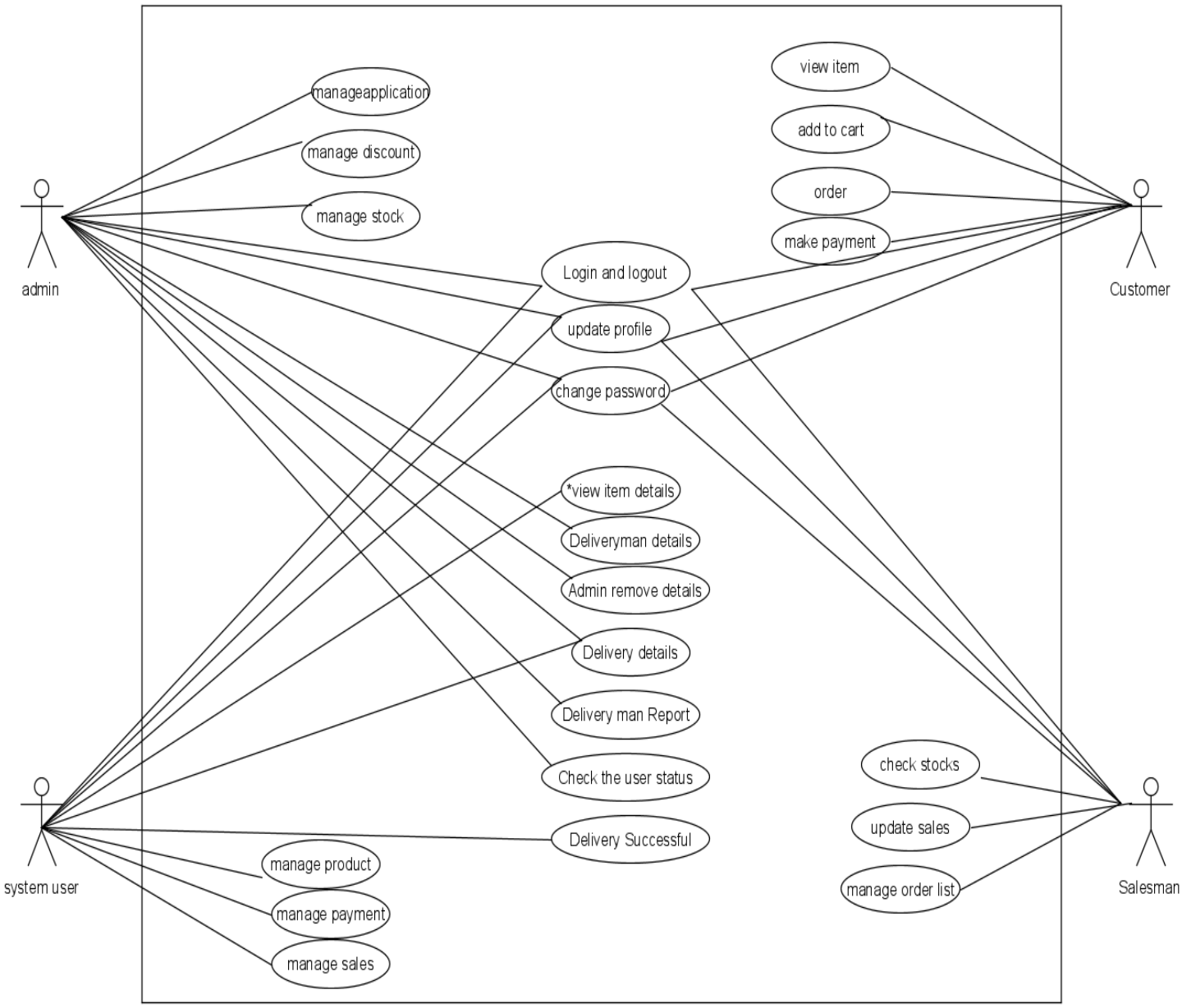
3. Diagram

a) Class Diagram:




product_category
id (int)
name (varchar)
description (text)
created_at (timestamp)
modified_at (timestamp)
deleted_at (timestamp)
product_inventory
id (int)
quantity (int)
created_at (timestamp)
modified_at (timestamp)
deleted_at (timestamp)
discount
-id (int)
name (varchar)
description (text)
discount_percent (decimal)
active (boolean)
created_at (timestamp)
modified_at (timestamp)
deleted_at (timestamp)
product
-id (int)
name (varchar)
description (text)
category_id (int)
inventory_id (int)
price (double)
discount_id (int)
created_at (timestamp)
modified_at (timestamp)
deleted_at (timestamp)
cart_item
id (int)
session_id (int)
product_id (int)
quantity (int)
created_at (timestamp)
modified_at (timestamp)
shopping_session
id (int)
user_id (int)
total (decumal)
created_at (timestamp)
modified_at (timestamp)
user
-id (int)
username (varchar)
email (varchar)
password (varchar)
first_name (varchar)
last_name (varchar)
mobile (varchar)
created_at (timestamp)
modified_at (timestamp)
order_items
id (int)
order_id (int)
product_id (int)
quantity (int)
created_at (timestamp)
modified_at (timestamp)
order_details
id (int)
user_id (int)
total (decimal)
quantity (int)
created_at (timestamp)
modified_at (timestamp)
payment_details
id (int)
order_id (int)
amount (int)
provider (varchar)
status (varchar)
created_at (timestamp)
modified_at (timestamp)
delivery_person
-id (int)
payment_id (int)
username (varchar)
email (varchar)
password (varchar)
first_name (varchar)
last_name (varchar)
mobile (varchar)
created_at (timestamp)
modified_at (timestamp)

b) **Use case Diagram:** This use case diagram there are four actor Admin, System user, Customer, Salesman.



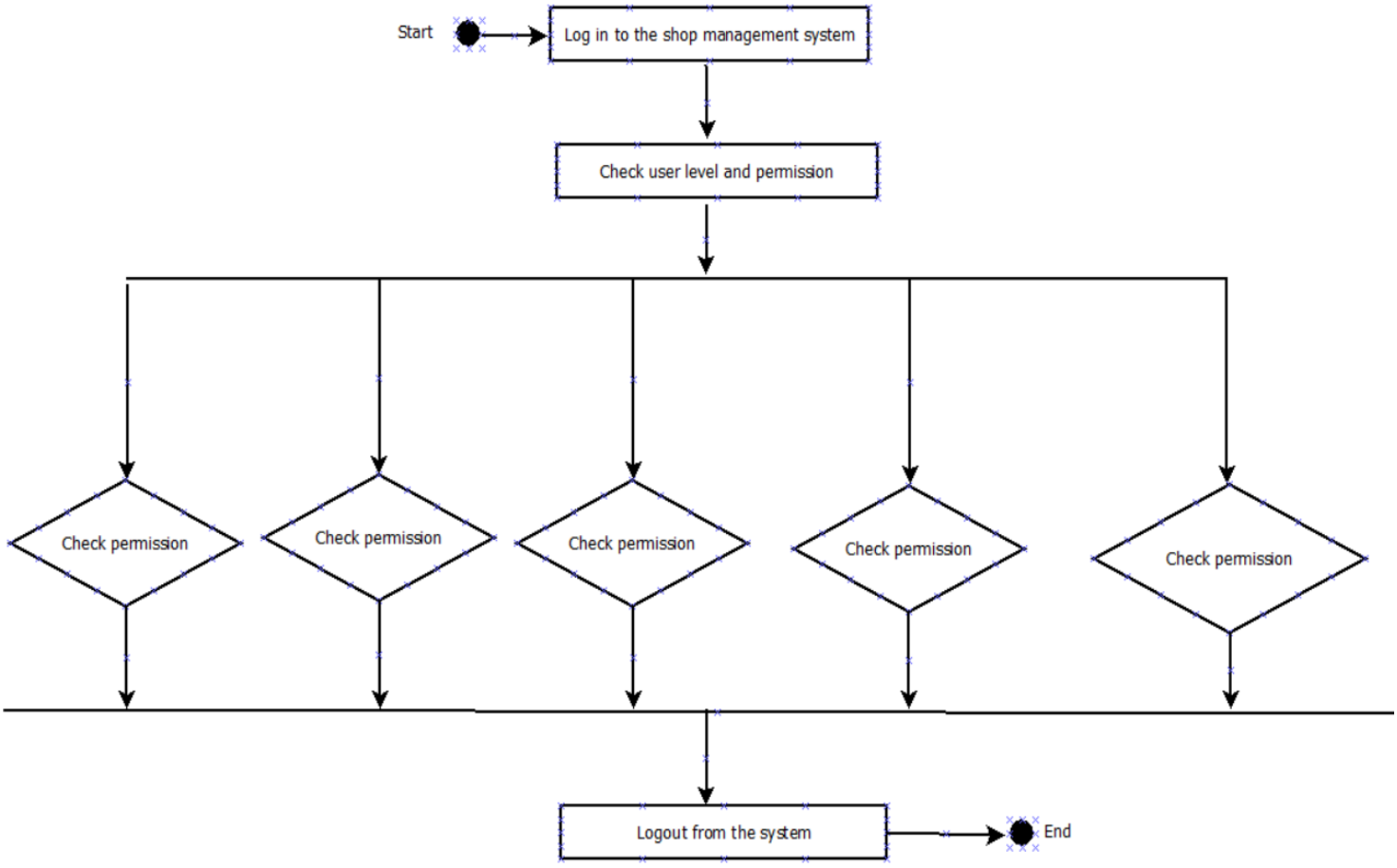
C) Activity Diagram:

This is activity diagram of shop management system which shows the flow between the activity of sales, stock inventory, payments, discounts, product. The main activity involved in this UMI Activity Diagram of shop management system are as follows:

1. Sales Activity
2.Stock Inventory Activity
3.Payments Activity
4.Discount Activity
5.Product Activity

Feature Of the Activity UML Diagram Of shop management system

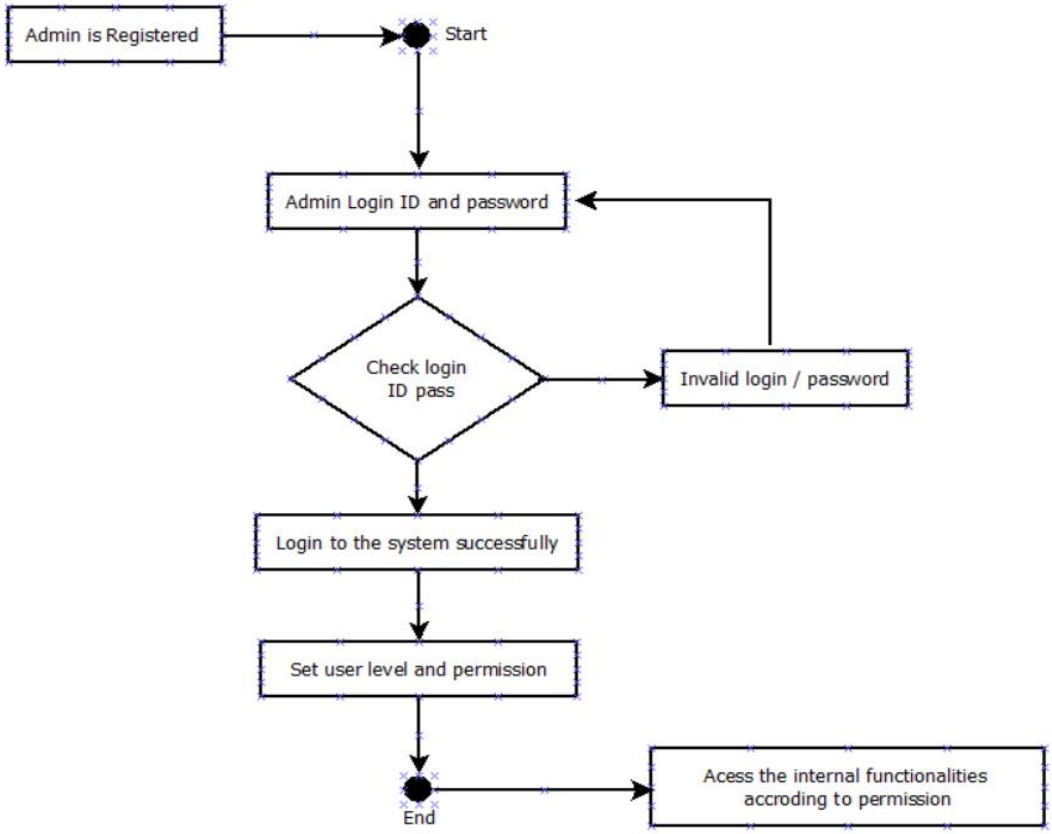
1. Admin user can search sales, view description of a selected sales, add sales, update sales and delete sales
2. It shows the activity flow of editing, adding and updating of stock inventory
3. User will be able to search and generate report of payments, Discounts, product
4. All objects such as (sales, stock inventory, product) are interlinked
5. Its show the full description and flow of sales, Discounts, product, payments, stock inventory

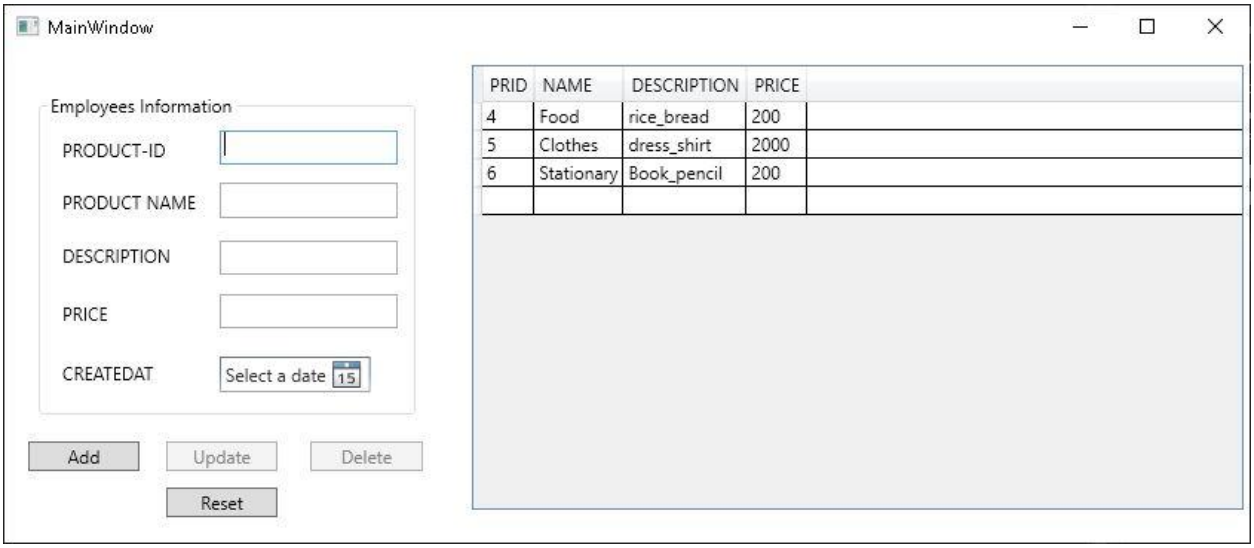


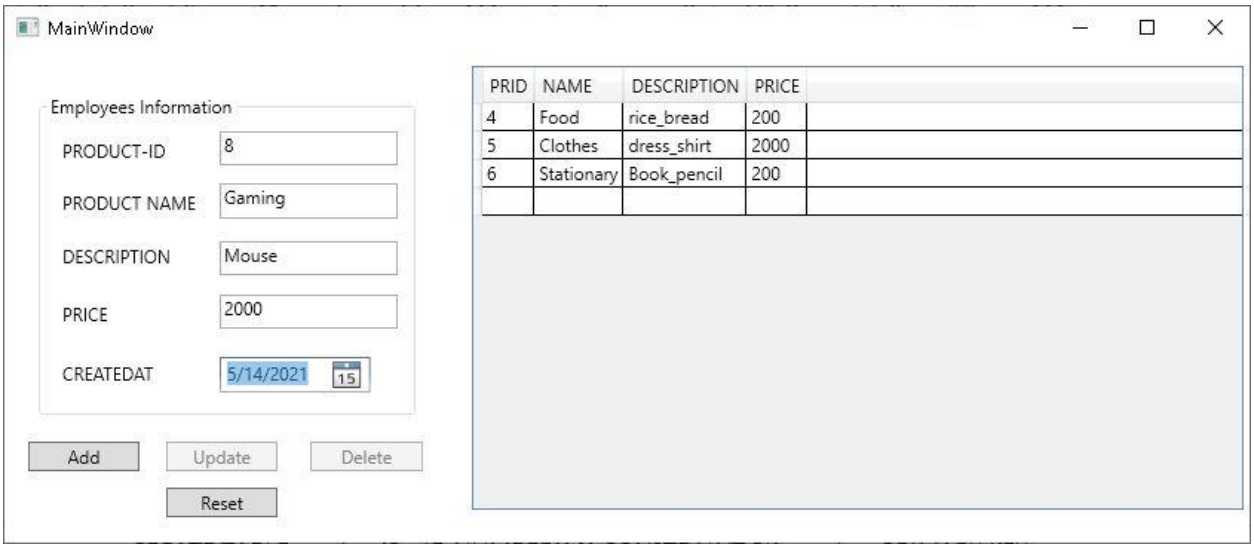
Login Activity Diagram Of shop management system:

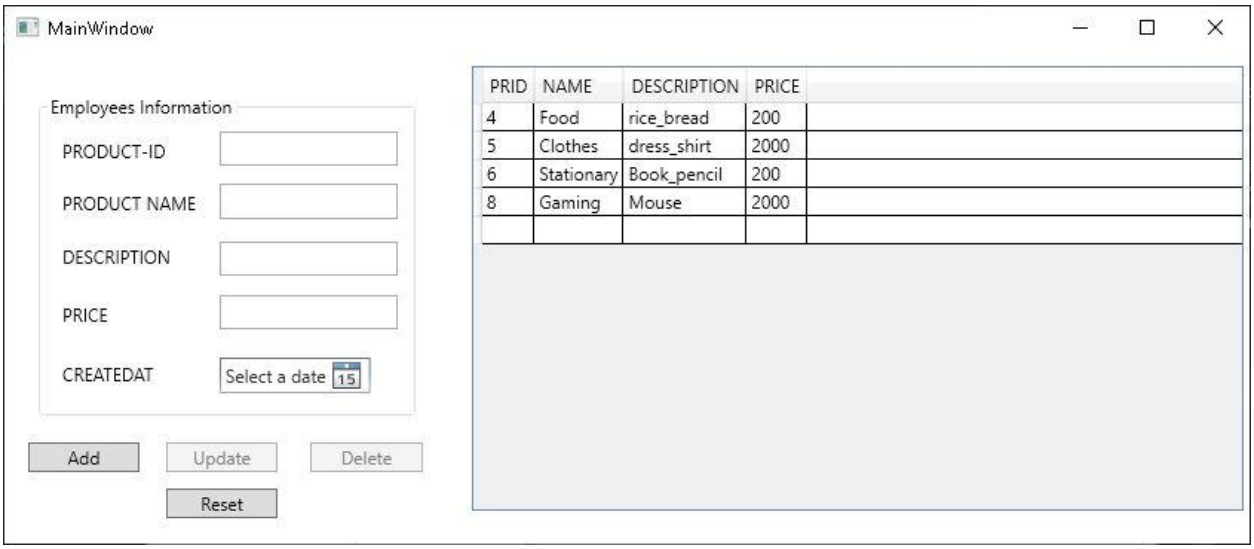
This is the Login Activity diagram of shop management system, which shows the flows of Login activity, where admin will be able to login using their username and password. After login user can manage all the operation on payments, sales, stock inventory, product, Discounts. All the pages such as stock inventory, product, Discounts are secure and user can access these pages after login. The diagram bellow helps demonstrate how the login page works in a shop management system. The various objects in the product, payments, sales, stock inventory and discount page-intact over the course of activity and user will not be able to access this page without verifying their identity

4.User Interface: Here we give two user interface login and registration




MainWindow
Employees Information
PRODUCT-ID
PRODUCT NAME
DESCRIPTION
PRICE
CREATEDAT
Select a date
Add
Update
Delete
Reset
PRID NAME DESCRIPTION PRICE
4 Food rice_bread 200
5 Clothes dress_shirt 2000
6 Stationary Book_pencil 200

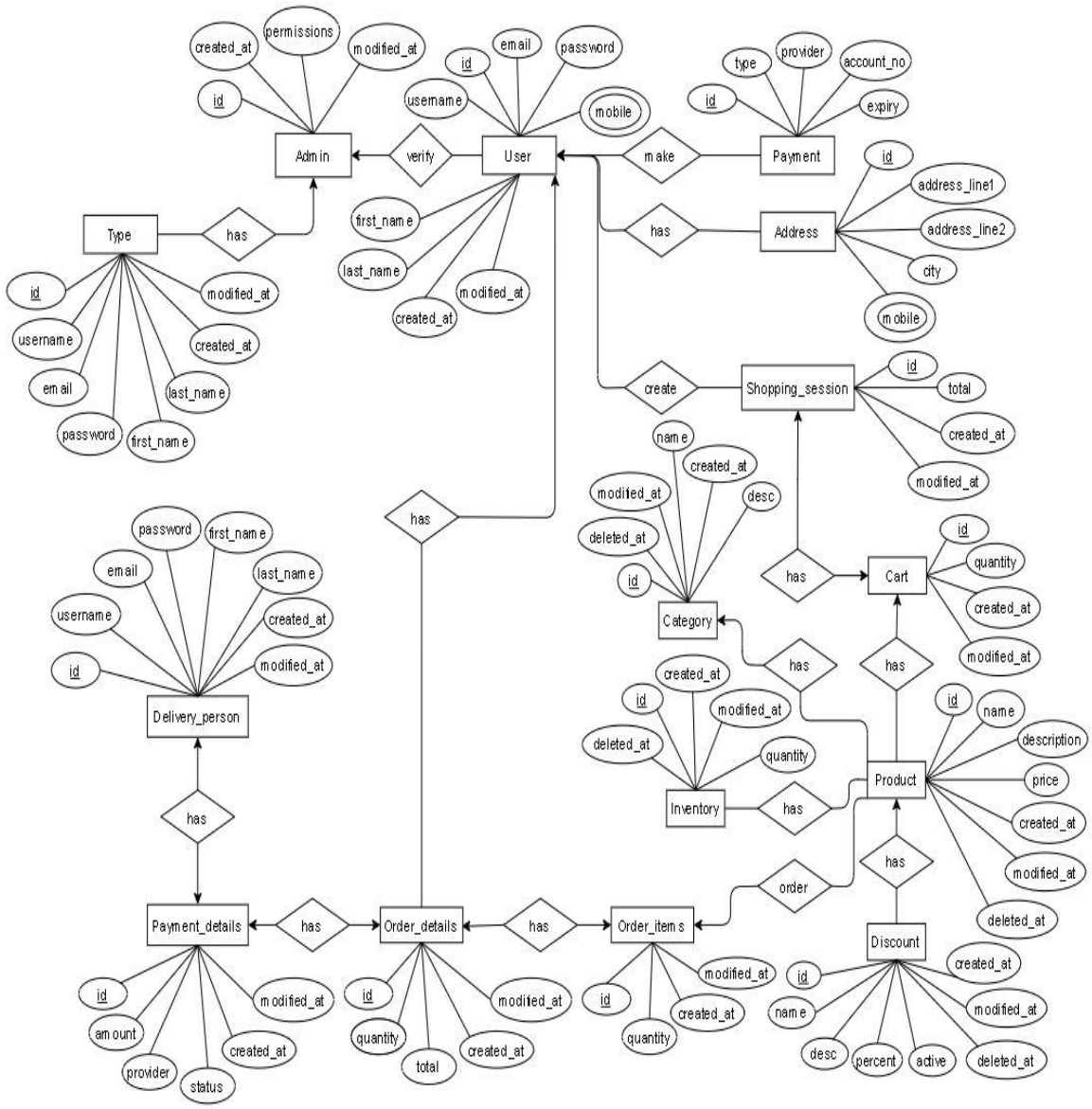

MainWindow
Employees Information
PRODUCT-ID 8
PRODUCT NAME Gaming
DESCRIPTION Mouse
PRICE 2000
CREATEDAT 5/14/2021
Add
Update
Delete
Reset
PRID NAME DESCRIPTION PRICE
4 Food rice_bread 200
5 Clothes dress_shirt 2000
6 Stationary Book_pencil 200


MainWindow
Employees Information
PRODUCT-ID
PRODUCT NAME
DESCRIPTION
PRICE
CREATEDAT
Select a date
Add
Update
Delete
Reset
PRID NAME DESCRIPTION PRICE
4 Food rice_bread 200
5 Clothes dress_shirt 2000
6 Stationary Book_pencil 200
8 Gaming Mouse 2000

5. **Scenario Description:**

This scenario description is about shop management system. We have many type of entities, multiple relationships among them and show all the visual instrument of database table and relations between customers,sells, product etc.The main entities of the shop management system are product,customer,Admin,payment,order,delivery.There is an one admin who can grant all the permission.Admin has type where type has lots of attributes like id,username,email,password,fast_name,last_name,last_name,created_at,modified_at.A lots of/many user create many shopping session here shopping session has attributes like id, total modify_at and create_at.One and many user can selected many product,product have an unique id,name,price,create_at,modified_at,delete_at, description and drop them in the cart.Product have a category.One product have one category.Category has been created_at,modified_at,deleted_at and also category have unique id,name.When one user selected many product and this was store his/her own cart.One user has One cart but Many product has store one cart.Here cart has id,quantity,create_at,modify_at. All the order details will be store in the database.Then user can get discount if there any discount is available.This discount has many product.Here discount have unique id,name,percent. After selecting all the product One user can confirm many order items.Order items have id,quantity,create_at,modify_at and User make payment like online mobile banking,Card etc.This payment details will be stored in database.One user pay their payment many payment methood.When make payment this time payment method created and also modified.Each payment have unique transaction id .After the payment the delivery man will be delivering the product to the user.Here delivery person details like first_name,last_name,email,username,unique id store in database.Many product deliver One and many delivery man.

6. ER Diagram:


Admin
created_at
permissions
modified_at
id
verify
User
id
email
password
username
mobile
make
Payment
type
provider
account_no
id
expiry
Type
has
id
modified_at
username
created_at
email
last_name
password
first_name
first_name
last_name
created_at
modified_at
has
Address
id
address_line1
address_line2
city
mobile
create
Shopping_session
id
total
created_at
modified_at
has
Cart
id
quantity
created_at
modified_at
has
has
name
created_at
modified_at
desc
deleted_at
id
Category
has
created_at
id
modified_at
deleted_at
quantity
Inventory
has
Product
id
name
description
price
created_at
modified_at
deleted_at
has
order
Order_items
id
modified_at
created_at
quantity
has
Order_details
id
modified_at
quantity
total
created_at
has
Payment_details
id
modified_at
amount
provider
status
created_at
has
Delivery_person
password
first_name
email
last_name
username
created_at
id
modified_at
has
Discount
id
created_at
name
modified_at
desc
percent
active
deleted_at

7. Normalization:

Has(T_id , username , email, password , first_name , Last_name , T_created_at , T_modified_at , A_id , A_created_at , T_modified_at , permissions)

1NF: no multivalued available

2NF:

T_id , username , email, password , first_name , Last_name , T_created_at , T_modified_at

A_id , A_created_at , T_modified_at , permissions

Relationship one to many

T_id , username , email, password , first_name , Last_name , T_created_at , T_modified_at

A_id , A_created_at , T_modified_at , permissions

TA_id , T_id , A_id

3NF : No transitive dependency available

Total table

1.**T_id** (primary key), username , email, password , first_name , Last_name , T_created_at , T_modified_at

2. **A_id** (primary key) , A_created_at , T_modified_at , permissions

3.**TA_id** (primary key), **T_id** (foreign key) , **A_id** (foreign key)

Has(D_id , username , email, password, first_name , Last_name , D_created_at , D_modified_at , Pa_id , amount , provider , status , P_created_at , P_modified_at)

1NF : no multivalued available

2NF :

D_id , username , email, password, first_name , Last_name , D_created_at , D_modified_at

Pa_id , amount , provider , status , P_created_at , P_modified_at

Relationship many to many

D_id , username , email, password, first_name , Last_name , D_created_at , D_modified_at

Pa_id , amount , provider , status , P_created_at , P_modified_at

DPa_id , D_id , Pa_id

3NF : No transitive dependency available

Total table

1.**D_id** (primary key) , username , email, password, first_name , Last_name , D_created_at , D_modified_at

2.**Pa_id** (primary key) , amount , provider , status , P_created_at , P_modified_at

3.**DPa_id** (primary key) , **D_id** (foreign key) **, Pa_id** (foreign key)

Has(Pa_id , amount , provider , status , P_created_at , P_modified_at , O_id , quantity , total , O_created_at , O_modified_at)

1NF : no multivalued available

2NF :

Pa_id , amount , provider , status , P_created_at , P_modified_at

O_id , quantity , total , O_created_at , O_modified_at

Relationship many to many

Pa_id , amount , provider , status , P_created_at , P_modified_at

O_id , quantity , total , O_created_at , O_modified_at

PaO_id , pa_id , O_id

3NF : No transitive dependency available

Total table

1.**Pa_id**(Primary key) , amount , provider , status , P_created_at , P_modified_at

2.**O_id**(primary key), quantity , total , O_created_at , O_modified_at

3.**PaO_id**(primary key) , **pa_id**(foreign key) , **O_id**(foreign key)

Verify(A_id , A_created_at , T_modified_at , permissions , U_id , username , email, password , mobile ,first_name , Last_name , U_created_at , U_modified_at)

1NF : one multivalued available

2NF :A_id , A_created_at , T_modified_at , permissions ,

U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

Relationship many to one

A_id , A_created_at , T_modified_at , permissions ,

U_id , username , email, password , mobile ,first_name , Last_name , U_created_at , U_modified_at

AU_id A_id , U_id

3NF : No transitive dependency available

Total table

1.**A_id**(primary key) , A_created_at , T_modified_at , permissions ,

2.**U_id**(primary key) , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

3.**AU_id**(primary key), **A_id**(foreign key) , **U_id**(foreign key)

Has(U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at, O_id , quantity , total , O_created_at , O_modified_at)

1NF : one multivalued available

2NF :U_id , username , email, password , mobile , first_name , Last_name , U_created_at ,

mU_modified_at,

O_id , quantity , total , O_created_at , O_modified_at

Relationship one to many

U_id , username , email, password , mobile ,first_name , Last_name , U_created_at , U_modified_at,

O_id , quantity , total , O_created_at , O_modified_at

Uo-id , U_id , O-Id

Total table

1.**U_id**(primary key) , username , email, password , mobile ,first_name , Last_name , U_created_at , U_modified_at,

2.**O_id**(primary key) , quantity , total , O_created_at , O_modified_at

3.**Uo-id**(primaryn key) **, U_id**(foreign key) **, O-Id**(foreign key)

Has(O_id , quantity , total , O_created_at , O_modified_at, oi_id , oi_quantity , Oi_created_at , Oi_modified)

1NF : no multivalued available

2NF :

O_id , quantity , total , O_created_at , O_modified_at,

Oi_id , oi_quantity , Oi_created_at , Oi_modified

Relationship many to many

O_id , quantity , total , O_created_at , O_modified_at,

Oi_id , oi_quantity , Oi_created_at , Oi_modified

Ooi_id O_id , Oi_id

3NF : No transitive dependency available

Total table

1.**O_id** (primary key), quantity , total , O_created_at , O_modified_at,

2.**Oi_id**(primary key) , oi_quantity , Oi_created_at , Oi_modified

3.**Ooi_id**(primary key) , **O_id**(foreign key) , **Oi_id**(foreign key)

Make(U_id , username , email, password ,mobile , first_name , Last_name , U_created_at , U_modified_at , p_id , type , provider , account_no , expiry)

1NF : one multivalued available

2NF :

U_id , username , email, password , first_name , Last_name , U_created_at , U_modified_at ,

P_id , type , provider , account_no , expiry

Relationship many to one

U_id , username , email, password , first_name , Last_name , U_created_at , U_modified_at

P_id , type , provider , account_no , expiry

Up_id , U_id , P_id

3NF : No transitive dependency available

Total table

1.**U_id**(primary key) , username , email, password , first_name , Last_name , U_created_at , U_modified_at ,

2.**P_id**(primary key) , type , provider , account_no , expiry

3.**Up_id**(primary key) , **U_id**(foreign key) **, P_id**(foreign key)

Has(U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at , Ad_id , address_line1 , Address_line2 , city , mobile)

1NF : one multivalued available

2NF :U_id , username , email, password , first_name , mobile , Last_name , U_created_at , U_modified_at ,

Ad_id , address_line1 , Address_line2 , city , mobile

Relationship one to one

U_id , username , email, password , first_name , Last_name , mobile , U_created_at , U_modified_at

Ad_id , address_line1 , Address_line2 , city , mobile

Uad_id _U_id , Ad_id

3NF : No transitive dependency available

Total table:

1.**U_id**(primary key) , username , email, password , first_name , Last_name , U_created_at , U_modified_at ,

2.**Ad_id** (primary key) , address_line1 , Address_line2 , city , mobile

3.**Uad_id**(primary key),**U_id**(foreign key) , **Ad_id**(foreign key)

Create(U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at , S_id , total , S_created_at , S_modified_at)

1NF : one multivalued available

2NF :

U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

S_id , total , S_created_at , S_modified_at

Relationship many to many

U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

S_id , total , S_created_at , S_modified_at

Us_id U_id , S_id

3NF : No transitive dependency available

Total table

1.**U_id**(primary key) , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

2.**S_id** (primary key) , total , S_created_at , S_modified_at

3.**Us_id**(primary key) **, U_id**(foreign key) , **S_id**(foreign key)

Has(S_id , total , S_created_at , S_modified_at , C_id , quantity , C_created_at , C_modified)

1NF : no multivalued available

2NF :

S_id , total , S_created_at , S_modified_at

C_id , quantity , C_created_at , C_modified

Relationship one to one

S_id , total , S_created_at , S_modified_at

C_id , quantity , C_created_at , C_modified

Sc_id , S_id , C_id

3NF : No transitive dependency available

Total table

1.**S_id** , total (primary key) , S_created_at , S_modified_at

2. **C_id** (primary key) , quantity , C_created_at , C_modified

3.**Sc_id(**primary key) **, S_id**(foreign key) , **C_id(**foreign key)

Has(Ca_id , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc , Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at)

1NF : no multivalued available

2NF :

Ca_id , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc ,

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Relationship many to many

Ca_id , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc ,

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Capr_id , Cr_id , Pr_id

3NF : No transitive dependency available

Total table

1.**Ca_id**(primary key) , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc ,

2.**Pr_id**(primary key) , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

3.**Capr_id** (primary key) ,**Cr_id**(foreign key) **, Pr_id**(foreign key)

Has(C_id , quantity , C_created_at , C_modified , Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at)

1NF : no multivalued available

2NF :

C_id , quantity , C_created_at , C_modified ,

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Relationship many to many

C_id , quantity , C_created_at , C_modified ,

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Cpr_id , C_id , Pr_id

3NF : No transitive dependency available

Total table

1.**C_id**(primary key) , quantity , C_created_at , C_modified ,

2.**Pr_id** (primary key) , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

3.**Cpr_id** (primary key) , **C_id** (foreign key) , **Pr_id**(foreign key)

Has(I_id , quantity , I_created_at , I_modified I_deleted_at , Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at)

1NF : no multivalued available

2NF :I_id , quantity , I_created_at , I_modified I_deleted_at

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Ipr_id I_id , Pr_id

Relationship many to many

I_id , quantity , I_created_at , I_modified I_deleted_at

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Ipr_id I_id , Pr_id

3NF : No transitive dependency available

Total table

1.**I_id**(primary key) , quantity , I_created_at , I_modified I_deleted_at

2. **Pr_id**(primary key) , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

3.**Ipr_id**(primary key) , **I_id**(foreign key) , **Pr_id**(foreign key)

Order(Oi_id , oi_quantity , Oi_created_at , Oi_modified, Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at)

1NF : no multivalued available

2NF :

Oi_id , oi_quantity , Oi_created_at , Oi_modified

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Relationship many to many

Oi_id , oi_quantity , Oi_created_at , Oi_modified

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

Oipr_id , Oi_id , Pr_id

3NF : No transitive dependency available

Total table

1.**Oi_id** (primary key) , oi_quantity , Oi_created_at , Oi_modified

2.**Pr_id** (primary key) , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

3.**Oipr_id** (primary key) , **Oi_id** (foreign key) , **Pr_id**(foreign key)

Has(Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at , Di_id , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name)

1NF : no multivalued available

2NF :

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at ,

Di_id , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name

Relationship many to many

Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at ,

Di_id , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name

Prdi_id , Pr_id , Di_id

3NF : No transitive dependency available

Total table

1.**Pr_id(**primary key) , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at ,

2.**Di_id**(primary key) , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name

3.**Prdi_id**(primary key) , **Pr_id**(foreign key) , **Di_id**(foreign key)

Total Table

1.T_id , username , email, password , first_name , Last_name , T_created_at , T_modified_at

2. A_id , A_created_at , T_modified_at , permissions

3.TA_id , T_id , A_id

4.D_id , username , email, password, first_name , Last_name , D_created_at , D_modified_at

5.Pa_id , amount , provider , status , P_created_at , P_modified_at

6.DPa_id , D_id , Pa_id

7.Pa_id , amount , provider , status , P_created_at , P_modified_at

8.O_id , quantity , total , O_created_at , O_modified_at

9.PaO_id , pa_id , O_id

10.A_id , A_created_at , T_modified_at , permissions

11.U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

12.AU_id A_id , U_id

13.U_id , username , email, password , mobile ,first_name , Last_name , U_created_at , U_modified_at,

14.O_id , quantity , total , O_created_at , O_modified_at

15.Uo-id , U_id , O-Id

16.O_id , quantity , total , O_created_at , O_modified_at,

17.Oi_id , oi_quantity , Oi_created_at , Oi_modified

18.Ooi_id O_id , Oi_id

19.U_id , username , email, password , first_name , Last_name , U_created_at , U_modified_at ,

20.P_id , type , provider , account_no , expiry

21.Up_id , U_id , P_id

22.U_id , username , email, password , first_name , Last_name , U_created_at , U_modified_at

23.Ad_id , address_line1 , Address_line2 , city , mobile

24.Uad_id _U_id , Ad_id

25.U_id , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

26.S_id , total , S_created_at , S_modified_at

27.Us_id U_id , S_id

28.S_id , total , S_created_at , S_modified_at

29. C_id , quantity , C_created_at , C_modified

30.Sc_id , S_id , C_id

31.Ca_id , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc ,

32.Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

33.Capr_id , Cr_id , Pr_id

34.C_id , quantity , C_created_at , C_modified

35.Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

36.Cpr_id , C_id , Pr_id

37.I_id , quantity , I_created_at , I_modified I_deleted_at

38. Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

39.Ipr_id I_id , Pr_id

40.Oi_id , oi_quantity , Oi_created_at , Oi_modified

41.Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

42.Oipr_id , Oi_id , Pr_id

43.Pr_id , name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

44.Di_id , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name

45.Prdi_id , Pr_id , Di_id

Selected Total table

1.T_id**(**primary key**)** , username , email, password , first_name , Last_name , T_created_at , T_modified_at

2. A_id **(**primary key**)**, A_created_at , T_modified_at , permissions

3.TA_id **(**primary key**)**, T_id**(**primary key**)** , **A_id(**foreign key**)**

4.D_id **(**primary key**)**, username , email, password, first_name , Last_name , D_created_at , D_modified_at

5.Pa_id**(**primary key**)** , amount , provider , status , P_created_at , P_modified_at

6.DPa_id**(**primary key**)** , D_id **(**primary key**)** , **Pa_id(**foreign key**)**

7.O_id(primary key**)** , quantity , total , O_created_at , O_modified_at

8.PaO_id**(**primary key**)** , pa_id**(**primary key**)** , **O_id(**foreign key**)**

9.U_id**(**primary key**)** , username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at

10.AU_id**(**primary key**)** , A_id**(**primary key**)** , **U_id (**foreign key**)**

11.Uo-id **(**primary key**)**, U_id(primary key**)** , **O-Id(**foreign key**)**

12.Oi_id**(**primary key**)** , oi_quantity, Oi_created_at , Oi_modified

13.Ooi_id **(**primary key**)** ,O_id**(**primary key**)** , **Oi_id(**foreign key**)**

14.P_id**(**primary key**)** , type , provider , account_no , expiry

15.Up_id**(**primary key**)** , U_id **(**primary key**)**, **P_id(**foreign key**)**

16.Ad_id , address_line1 , Address_line2 , city , mobile

17.Uad_id **(**primary key**)**,U_id**(**primary key**)** , **Ad_id(**foreign key**)**

18.S_id**(**primary key**)** , total , S_created_at , S_modified_at

19.Us_id**(**primary key**)** , U_id**(**primary key**)** **, S_id(**froeign key**)**

20. C_id**(**primary key**)** , quantity , C_created_at , C_modified

21.Sc_id **(**primary key**)**, S_id**(**primary key**)** , **C_id(**foreign key**)**

22.Ca_id**(**primary key**)** , Ca_created_at , Ca_modified, name , Ca_deleted_at , desc ,

23.Pr_id **(**primary key**)**, name , description , price , Pr_created_at , Pr_modified, name , Pr_deleted_at

24.Capr_id**(**primary key**)** , Cr_id**(**primary key**)** , **Pr_id(**foreign key**)**

25.Cpr_id **(**primary key**)**, C_id**(**primary key**)** , **Pr_id(**foreign key**)**

26.I_id**(**primary key**)** , quantity , I_created_at , I_modified I_deleted_at

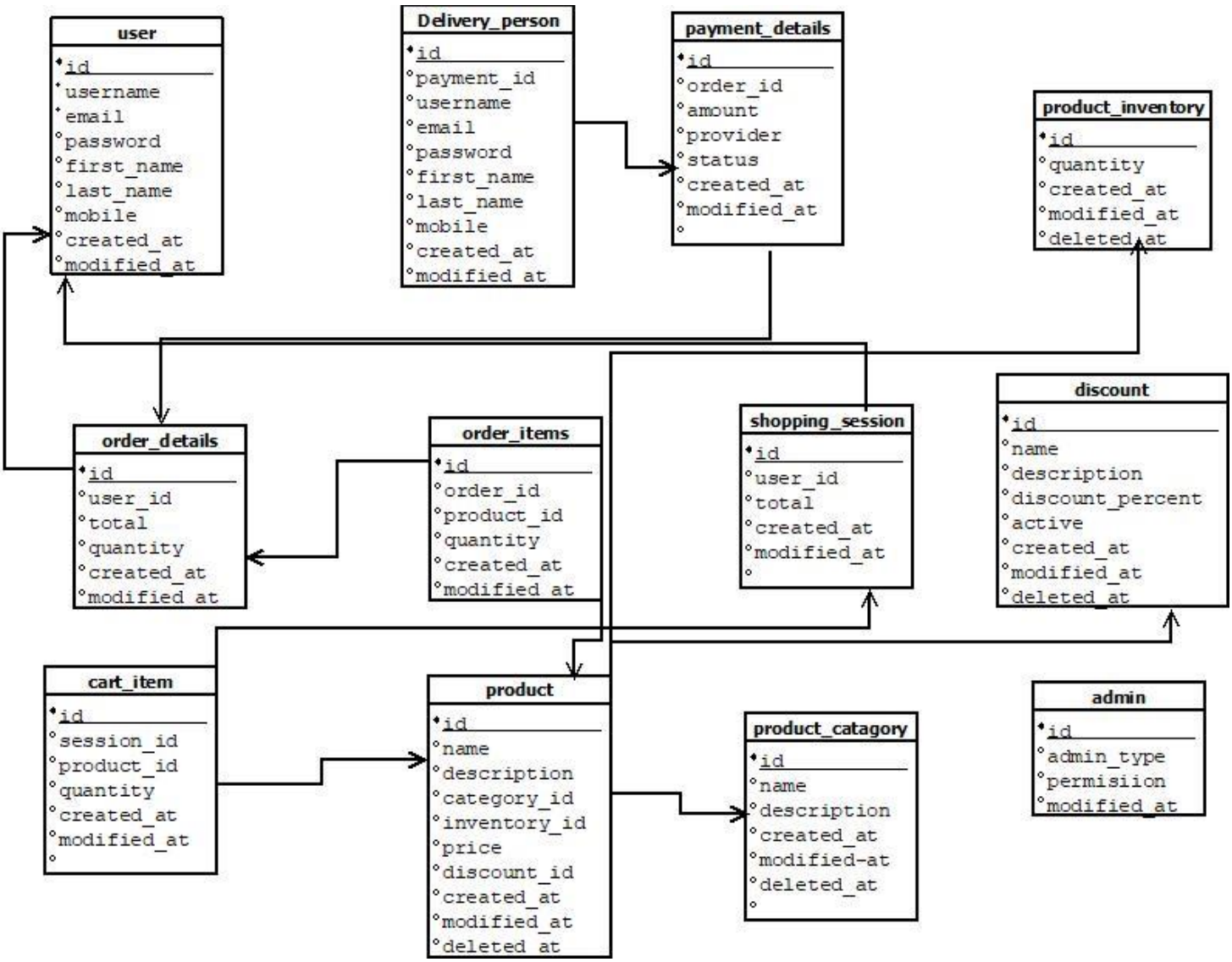
27.Ipr_id**(**primary key**)** , I_id**(**primary key**)** , **Pr_id(**foreign key**)**

28.Oipr_id**(**primary key**)** , Oi_id**(**primary key**) , Pr_id(**foreign key**)**

29.Di_id**(**primary key**)** , name , desc , parcent , active , di_deleted_at , Di_created_at , Di_modified, name

30.Prdi_id **(**primary key**)** , Pr_id **(**primary key**)**, **Di_id(**foreign key**)**

8. Schema Diagram: Here is Schema diagram



9.Table Creation:

1. create sequence type_T_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE type(T_ID NUMBER(10) CONSTRAINT PK_type PRIMARY KEY,username VARCHAR2(25) ,email VARCHAR2(50),password VARCHAR2(20),First_name VARCHAR2(20),Last_name VARCHAR2(50), T_created_at VARCHAR2(50), T_modified_at VARCHAR2(50))

CREATE INDEX type_index ON type(T_ID,username , email, password , first_name , Last_name , T_created_at , T_modified_at);

2. create sequence admin_A_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE admin(A_ID NUMBER(10) CONSTRAINT PK_admin PRIMARY KEY, T_created_at VARCHAR2(50), T_modified_at VARCHAR2(50) , permissions VARCHAR2(50))

CREATE INDEX admin_index ON admin(A_ID, T_created_at , T_modified_at , permissions);

3.create sequence deliveryperson_D_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE deliveryperson(D_ID NUMBER(10) CONSTRAINT PK_deliveryperson PRIMARY KEY,username VARCHAR2(25) ,email VARCHAR2(50),password VARCHAR2(20),First_name VARCHAR2(20),Last_name VARCHAR2(50), D_created_at VARCHAR2(50), D_modified_at VARCHAR2(50))

CREATE INDEX deliveryperson_index ON deliveryperson(D_ID,username , email, password , first_name , Last_name , D_created_at , D_modified_at);

4. create sequence paymentdetails_Pa_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE paymentdetails(Pa_ID NUMBER(10) CONSTRAINT PK_paymentdetails PRIMARY KEY,amount VARCHAR2(50) ,provider VARCHAR2(50) , status VARCHAR2(50) , P_created_at VARCHAR2(50), P_modified_at VARCHAR2(50))

CREATE INDEX paymentdetails_index ON paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at);

5. create sequence orderdetails_O_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE orderdetails(O_ID NUMBER(10) CONSTRAINT PK_orderdetails PRIMARY KEY,quantity VARCHAR2(50) , total VARCHAR2(50) , O_created_at VARCHAR2(50), O_modified_at VARCHAR2(50)) CREATE INDEX orderdetails_index ON orderdetails(O_ID, quantity , total , O_created_at , O_modified_at);

6. create sequence user_U_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE user1(U_ID NUMBER(10) CONSTRAINT PK_user PRIMARY KEY,username VARCHAR2(25) ,email VARCHAR2(50),password VARCHAR2(20),mobile VARCHAR2(20),First_name VARCHAR2(20),Last_name VARCHAR2(50), U_created_at VARCHAR2(50), U_modified_at VARCHAR2(50))

CREATE INDEX user1_index ON user1(U_ID, username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at);

7. create sequence orderitems_Oi_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE orderitems(Oi_ID NUMBER(10) CONSTRAINT PK_orderitems PRIMARY KEY, oi_quantity VARCHAR2(50), Oi_created_at VARCHAR2(50), Oi_modified_at VARCHAR2(50))

CREATE INDEX orderitems_index ON orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at);

8. create sequence payment_P_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE payment(P_ID NUMBER(10) CONSTRAINT PK_payment PRIMARY KEY, type VARCHAR2(50), provider VARCHAR2(50), account_no VARCHAR2(50) , expiry VARCHAR(50))

CREATE INDEX payment_index ON payment(P_ID , type , provider , account_no , expiry);

9. create sequence address_Ad_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE address(Ad_ID NUMBER(10) CONSTRAINT PK_address PRIMARY KEY,address_line1 VARCHAR2(25) ,address_line2 VARCHAR2(50),city VARCHAR2(20),mobile VARCHAR2(20))

CREATE INDEX address_index ON address(Ad_ID,address_line1 , Address_line2 , city , mobile);

10. create sequence shoppingsession_S_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE shoppingsession(S_ID NUMBER(10) CONSTRAINT PK_shoppingsession PRIMARY KEY, total VARCHAR2(50), S_created_at VARCHAR2(50), S_modified_at VARCHAR2(50))

CREATE INDEX shoppingsession_index ON shoppingsession(S_ID, total , S_created_at , S_modified_at);

11. create sequence cart_C_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE cart(C_ID NUMBER(10) CONSTRAINT PK_cart PRIMARY KEY, quantity VARCHAR2(50), C_created_at VARCHAR2(50), C_modified_at VARCHAR2(50))

CREATE INDEX cart_index ON cart(C_ID, quantity , C_created_at , C_modified_at);

12. create sequence catagory_Ca_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE catagory(Ca_ID NUMBER(10) CONSTRAINT PK_catagory PRIMARY KEY,name VARCHAR2(25) , Ca_created_at VARCHAR2(50), Ca_modified_at VARCHAR2(50) ,Ca_deleted_at VARCHAR2(50), describe VARCHAR2(50))

CREATE INDEX catagory_index ON catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe);

13. create sequence product_Pr_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE product(Pr_ID NUMBER(10) CONSTRAINT PK_product PRIMARY KEY,name VARCHAR2(25) , description VARCHAR2(25) , price VARCHAR2(25) ,Pr_created_at VARCHAR2(50), Pr_modified_at VARCHAR2(50) ,Pr_deleted_at VARCHAR2(50))

CREATE INDEX product_index ON product(Pr_ID , name , description , price , Pr_created_at , Pr_modified_at , Pr_deleted_at);

14.create sequence inventory_I_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE inventory(I_ID NUMBER(10) CONSTRAINT PK_inventory PRIMARY KEY,quantity VARCHAR2(50) , I_created_at VARCHAR2(50), I_modified_at VARCHAR2(50), I_deleted_at VARCHAR2(50))

CREATE INDEX inventory_index ON inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at);

15.create sequence discount_Di_ID_seq start with 1 increment by 1 nocache;

CREATE TABLE discount(Di_ID NUMBER(10) CONSTRAINT PK_discount PRIMARY KEY,name VARCHAR2(50) , desc1 VARCHAR2(50) , parcent VARCHAR2(50) , active VARCHAR2(50) ,Di_created_at VARCHAR2(50), Di_modified_at VARCHAR2(50), Di_deleted_at VARCHAR2(50))

CREATE INDEX discount_index ON discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at);

10.Data insertion:

1.

INSERT INTO type(T_ID, username , email, password , first_name , Last_name , T_created_at , T_modified_at)
VALUES(type_T_ID_seq.NEXTVAL,'oishorjo','ishorju2018@gmail.com','12345', 'oisho' , 'agnes' , '1.5.21' ,'2.6.21');

INSERT INTO type(T_ID, username , email, password , first_name , Last_name , T_created_at , T_modified_at) VALUES(type_T_ID_seq.NEXTVAL,'Dip','dip@gmail.com','1234567', 'naimul' , 'Dip' , '3.5.21' ,'4.6.21');

INSERT INTO type(T_ID, username , email, password , first_name , Last_name , T_created_at , T_modified_at) VALUES(type_T_ID_seq.NEXTVAL,'somudro','somudro@gmail.com','23456', 'arafee' , 'Somudro' , '8.5.21' ,'6.6.21');

INSERT INTO type(T_ID, username , email, password , first_name , Last_name , T_created_at , T_modified_at) VALUES(type_T_ID_seq.NEXTVAL,'shuvo','shuvo@gmail.com',' 17545', 'shuvo' , 'rahman' , '4.5.21' ,'5.6.21');

INSERT INTO type(T_ID, username , email, password , first_name , Last_name , T_created_at , T_modified_at) VALUES(type_T_ID_seq.NEXTVAL,'iqbal','iqbal@gmail.com','67545', 'shuvo' , 'rahman' , '14.5.21' ,'15.6.21');

T_ID	USERNAME	EMAIL	PASSWORD	FIRST_NAME	LAST_NAME	T_CREATED_AT	T_MODIFIED_AT
1	oishorjo	ishorju2018@gmail.com	12345	oisho	agnes	1.5.21	2.6.21
2	Dip	dip@gmail.com	1234567	naimul	Dip	3.5.21	4.6.21
3	somudro	somudro@gmail.com	23456	arafee	Somudro	8.5.21	6.6.21
4	shuvo	shuvo@gmail.com	17545	shuvo	rahman	4.5.21	5.6.21
5	iqbal	iqbal@gmail.com	67545	shuvo	rahman	14.5.21	15.6.21

2.

INSERT INTO admin(A_ID, T_created_at , T_modified_at , permissions)
VALUES(admin_A_ID_seq.NEXTVAL,'2.5.21', '1.5.21' ,'Yes');

INSERT INTO admin(A_ID, T_created_at , T_modified_at , permissions)
VALUES(admin_A_ID_seq.NEXTVAL,'7.5.21', '6.5.21' ,'Yes');

INSERT INTO admin(A_ID, T_created_at , T_modified_at , permissions)
VALUES(admin_A_ID_seq.NEXTVAL,'7.5.21', '9.5.21' ,'Yes');

INSERT INTO admin(A_ID, T_created_at , T_modified_at , permissions)
VALUES(admin_A_ID_seq.NEXTVAL,'12.5.21', '11.5.21' ,'Yes');

INSERT INTO admin(A_ID, T_created_at , T_modified_at , permissions)
VALUES(admin_A_ID_seq.NEXTVAL,'11.5.21', '12.5.21' ,'Yes');

A_ID	T_CREATED_AT	T_MODIFIED_AT	PERMISSIONS
1	2.5.21	1.5.21	Yes
2	7.5.21	6.5.21	Yes
3	7.5.21	9.5.21	Yes
4	12.5.21	11.5.21	Yes
5	11.5.21	12.5.21	Yes

3.

INSERT INTO deliveryperson(D_ID, username , email, password , first_name , Last_name , D_created_at , D_modified_at)
VALUES(deliveryperson_D_ID_seq.NEXTVAL,'merina','merina@gmail.com','1234567', 'tanjin' , 'Shithi' , '1.5.21' ,'2.6.21');

INSERT INTO deliveryperson(D_ID, username , email, password , first_name , Last_name , D_created_at , D_modified_at)
VALUES(deliveryperson_D_ID_seq.NEXTVAL,'Bappy','bappy@gmail.com','123456789', 'Supantha' , 'Bappy' , '14.5.21' ,'7.6.21');

INSERT INTO deliveryperson(D_ID, username , email, password , first_name , Last_name , D_created_at , D_modified_at)
VALUES(deliveryperson_D_ID_seq.NEXTVAL,'Sadat','sadat@gmail.com','1234678', 'Jubayer' , 'Sadat' , '8.5.21' ,'21.6.21');

INSERT INTO deliveryperson(D_ID, username , email, password , first_name , Last_name , D_created_at , D_modified_at)

VALUES(deliveryperson_D_ID_seq.NEXTVAL,'karim','karim@gmail.com','1237878', 'sheikh' , 'karim' , '8.5.21' ,'21.6.21');

INSERT INTO deliveryperson(D_ID, username , email, password , first_name , Last_name , D_created_at , D_modified_at)
VALUES(deliveryperson_D_ID_seq.NEXTVAL,'Rahim','rarim@gmail.com','12356778', 'Islam' , 'Rarim' , '18.5.21' ,'22.6.21');

D_ID	USERNAME	EMAIL	PASSWORD	FIRST_NAME	LAST_NAME	D_CREATED_AT	D_MODIFIED_AT
1	merina	merina@gmail.com	1234567	tanjin	Shithi	1.5.21	2.6.21
2	Bappy	bappy@gmail.com	123456789	Supantha	Bappy	14.5.21	7.6.21
3	Sadat	sadat@gmail.com	1234678	Jubayer	Sadat	8.5.21	21.6.21
4	karim	karim@gmail.com	1237878	sheikh	karim	8.5.21	21.6.21
5	Rahim	rarim@gmail.com	12356778	Islam	Rarim	18.5.21	22.6.21

4.

INSERT INTO paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at

) VALUES(paymentdetails_Pa_ID_seq.NEXTVAL,'1000', 'Hasan' ,'Complete' , '2.5.21', '1.5.21');

INSERT INTO paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at

) VALUES(paymentdetails_Pa_ID_seq.NEXTVAL,'5000', 'Faridul' ,'Complete' , '8.2.21', '4.5.21');

INSERT INTO paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at

) VALUES(paymentdetails_Pa_ID_seq.NEXTVAL,'7000', 'Sadia' ,'Complete' , '6.5.21', '10.5.21');

INSERT INTO paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at

) VALUES(paymentdetails_Pa_ID_seq.NEXTVAL,'8000', 'Zohra' ,'Complete' , '16.5.21', '17.5.21');

INSERT INTO paymentdetails(Pa_ID, amount , provider , status , P_created_at , P_modified_at

) VALUES(paymentdetails_Pa_ID_seq.NEXTVAL,'5000', 'tanjin' ,'Complete' , '6.5.21', '7.5.21');

PA_ID	AMOUNT	PROVIDER	STATUS	P_CREATED_AT	P_MODIFIED_AT
1	11000	Hasan	Complete	2.5.21	1.5.21
2	15000	Faridul	Complete	8.2.21	4.5.21
3	17000	Sadia	Complete	6.5.21	10.5.21
4	8000	Zohra	Complete	16.5.21	17.5.21
5	8000	Zohra	Complete	16.5.21	17.5.21
6	5000	tanjin	Complete	6.5.21	7.5.21

5.

```
INSERT INTO orderdetails(O_ID, quantity , total , O_created_at , O_modified_at)
VALUES(orderdetails_O_ID_seq.NEXTVAL,'10' , '10' ,'2.5.21', '1.5.21' );

INSERT INTO orderdetails(O_ID, quantity , total , O_created_at , O_modified_at)
VALUES(orderdetails_O_ID_seq.NEXTVAL,'20' , '20' ,'4.5.21', '8.5.21' );

INSERT INTO orderdetails(O_ID, quantity , total , O_created_at , O_modified_at)
VALUES(orderdetails_O_ID_seq.NEXTVAL,'40' , '40' ,'8.5.21', '9.5.21' );

INSERT INTO orderdetails(O_ID, quantity , total , O_created_at , O_modified_at)
VALUES(orderdetails_O_ID_seq.NEXTVAL,'60' , '60' ,'18.5.21', '19.5.21' );

INSERT INTO orderdetails(O_ID, quantity , total , O_created_at , O_modified_at)
VALUES(orderdetails_O_ID_seq.NEXTVAL,'90' , '90' ,'18.5.21', '19.5.21' );
```

O_ID	QUANTITY	TOTAL	O_CREATED_AT	O_MODIFIED_AT
1	5010	10	2.5.21	1.5.21
2	5020	20	4.5.21	8.5.21
3	5040	40	8.5.21	9.5.21
4	60	60	18.5.21	19.5.21
5	90	90	18.5.21	19.5.21

6.

INSERT INTO user1(U_ID, username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at)
VALUES(user_U_ID_seq.NEXTVAL,'fahim','fahim@gmail.com','1234567','01873678676', 'Islam' , 'Fahim' , '1.5.21' ,'2.6.21');

INSERT INTO user1(U_ID, username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at)
VALUES(user_U_ID_seq.NEXTVAL,'fahim','fahim@gmail.com','1234567','01873678676', 'Islam' , 'Fahim' , '1.5.21' ,'2.6.21');

INSERT INTO user1(U_ID, username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at)
VALUES(user_U_ID_seq.NEXTVAL,'mahadi','mahadi@gmail.com','12356767','01876765632', 'Haque' , 'Mahadi' , '9.5.21' ,'11.6.21');

INSERT INTO user1(U_ID, username , email, password , mobile , first_name , Last_name , U_created_at , U_modified_at)
VALUES(user_U_ID_seq.NEXTVAL,'Ishmam','ishmam@gmail.com','123569090','0171267893', 'Haque' , 'Ishmam' , '9.3.21' ,'11.3.21');

U_ID	USERNAME	EMAIL	PASSWORD	MOBILE	FIRST_NAME	LAST_NAME	U_CREATED_AT	U_MODIFIED_AT
1	Ahmed	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
2	fahim	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
3	fahim	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
4	maha	mahadi@gmail.com	12356767	01876765632	Haque	Mahadi	9.5.21	11.6.21
5	Ishmam	ishmam@gmail.com	123569090	0171267893	Haque	Ishmam	9.3.21	11.3.21

7.

INSERT INTO orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at)
VALUES(orderitems_Oi_ID_seq.NEXTVAL,'10' ,'2.5.21', '1.5.21');

INSERT INTO orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at)
VALUES(orderitems_Oi_ID_seq.NEXTVAL,'22' ,'9.5.21', '11.5.21');

INSERT INTO orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at)
VALUES(orderitems_Oi_ID_seq.NEXTVAL,'20' ,'9.4.21', '11.4.21');

INSERT INTO orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at)
VALUES(orderitems_Oi_ID_seq.NEXTVAL,'10' ,'19.4.21', '20.4.21');

INSERT INTO orderitems(Oi_ID,oi_quantity , Oi_created_at , Oi_modified_at)
VALUES(orderitems_Oi_ID_seq.NEXTVAL,'80' ,'11.4.21', '12.4.21');

O_ID	QUANTITY	TOTAL	O_CREATED_AT	O_MODIFIED_AT
1	5010	10	2.5.21	1.5.21
2	5020	20	4.5.21	8.5.21
3	5040	40	8.5.21	9.5.21
4	60	60	18.5.21	19.5.21
5	90	90	18.5.21	19.5.21

8.

INSERT INTO payment(P_ID , type , provider , account_no , expiry)
VALUES(payment_P_ID_seq.NEXTVAL,'cash' ,'Hasan', '123','1.5.21');

INSERT INTO payment(P_ID , type , provider , account_no , expiry)
VALUES(payment_P_ID_seq.NEXTVAL,'credit' ,'Iqbal', '124','12.5.21');

INSERT INTO payment(P_ID , type , provider , account_no , expiry)
VALUES(payment_P_ID_seq.NEXTVAL,'cash' ,'Rahman', '126','1.6.21');

INSERT INTO payment(P_ID , type , provider , account_no , expiry)
VALUES(payment_P_ID_seq.NEXTVAL,'cash' ,'Sam', '120','11.6.21');

INSERT INTO payment(P_ID , type , provider , account_no , expiry)
VALUES(payment_P_ID_seq.NEXTVAL,'cash' ,'masud', '121','20.6.21');

P_ID	TYPE	PROVIDER	ACCOUNT_NO	EXPIRY
1	cash	Hasan	123	1.5.21
2	credit	Iqbal	124	12.5.21
3	cash	Rahman	126	1.6.21
4	cash	Sam	120	11.6.21
5	cash	masud	121	20.6.21

9.

INSERT INTO address(Ad_ID, address_line1 , Address_line2 , city , mobile)
VALUES(address_Ad_ID_seq.NEXTVAL,'Shahinbag','mirpur','Dhaka', '01765789876');

INSERT INTO address(Ad_ID, address_line1 , Address_line2 , city , mobile)
VALUES(address_Ad_ID_seq.NEXTVAL,'East Rajabazar','Farmgate','Dhaka', '01765788766');

INSERT INTO address(Ad_ID, address_line1 , Address_line2 , city , mobile)
VALUES(address_Ad_ID_seq.NEXTVAL,'Lake park','Gulshan','Dhaka', '01967893765');

INSERT INTO address(Ad_ID, address_line1 , Address_line2 , city , mobile)
VALUES(address_Ad_ID_seq.NEXTVAL,'Residential area','Bashundhara','Dhaka',
'01978903657');

INSERT INTO address(Ad_ID, address_line1 , Address_line2 , city , mobile)
VALUES(address_Ad_ID_seq.NEXTVAL,'Kuratoli','Bisshoroad','Dhaka', '01789026546');

AD_ID	ADDRESS_LINE1	ADDRESS_LINE2	CITY	MOBILE
1	Shahinbag	mirpur	Dhaka	01765789876
2	East Rajabazar	Farmgate	Dhaka	01765788766
3	Lake park	Gulshan	Dhaka	01967893765
4	Residential area	Bashundhara	Dhaka	01978903657
5	Kuratoli	Bisshoroad	Dhaka	01789026546

10.

INSERT INTO shoppingsession(S_ID, total , S_created_at , S_modified_at)
VALUES(shoppingsession_S_ID_seq.NEXTVAL,'10' ,'2.5.21', '1.5.21');

INSERT INTO shoppingsession(S_ID, total , S_created_at , S_modified_at)
VALUES(shoppingsession_S_ID_seq.NEXTVAL,'20' ,'21.5.21', '22.5.21');

INSERT INTO shoppingsession(S_ID, total , S_created_at , S_modified_at)
VALUES(shoppingsession_S_ID_seq.NEXTVAL,'30' ,'12.5.21', '13.5.21');

INSERT INTO shoppingsession(S_ID, total , S_created_at , S_modified_at)
VALUES(shoppingsession_S_ID_seq.NEXTVAL,'50' ,'12.5.21', '13.5.21');

INSERT INTO shoppingsession(S_ID, total , S_created_at , S_modified_at)
VALUES(shoppingsession_S_ID_seq.NEXTVAL,'90' ,'2.5.21', '3.5.21');

S_ID	TOTAL	S_CREATED_AT	S_MODIFIED_AT
1	10	2.5.21	1.5.21
2	20	21.5.21	22.5.21
3	30	12.5.21	13.5.21
4	50	12.5.21	13.5.21
5	90	2.5.21	3.5.21

11.

INSERT INTO cart(C_ID, quantity , C_created_at , C_modified_at)
VALUES(cart_C_ID_seq.NEXTVAL,'5' ,'2.5.21', '1.5.21');

INSERT INTO cart(C_ID, quantity , C_created_at , C_modified_at)
VALUES(cart_C_ID_seq.NEXTVAL,'15' ,'12.5.21', '13.5.21');

INSERT INTO cart(C_ID, quantity , C_created_at , C_modified_at)
VALUES(cart_C_ID_seq.NEXTVAL,'12' ,'21.5.21', '22.5.21');

INSERT INTO cart(C_ID, quantity , C_created_at , C_modified_at)
VALUES(cart_C_ID_seq.NEXTVAL,'20' ,'11.5.21', '12.5.21');

INSERT INTO cart(C_ID, quantity , C_created_at , C_modified_at)
VALUES(cart_C_ID_seq.NEXTVAL,'50' ,'1.5.21', '2.5.21');

C_ID	QUANTITY	C_CREATED_AT	C_MODIFIED_AT
1	5	2.5.21	1.5.21
2	15	12.5.21	13.5.21
3	12	21.5.21	22.5.21
4	20	11.5.21	12.5.21
5	50	1.5.21	2.5.21

12.

INSERT INTO catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe) VALUES(catagory_Ca_ID_seq.NEXTVAL, '1.5.21' ,'2.6.21' , ' food' , '5.6.21' , 'rice - bread_butter');

INSERT INTO catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe) VALUES(catagory_Ca_ID_seq.NEXTVAL, '4.5.21' ,'5.6.21' , ' Stationary' , '6.6.21' , 'book -Pencil_pen');

INSERT INTO catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe) VALUES(catagory_Ca_ID_seq.NEXTVAL, '8.5.21' ,'9.6.21' , ' Clothes' , '10.6.21' , 'dress -shirt_pant');

INSERT INTO catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe) VALUES(catagory_Ca_ID_seq.NEXTVAL, '11.5.21' ,'9.6.21' , ' Bags' , '11.6.21' , 'ladiesbag -bagpack');

INSERT INTO catagory(Ca_ID, Ca_created_at , Ca_modified_at, name , Ca_deleted_at , describe) VALUES(catagory_Ca_ID_seq.NEXTVAL, '12.5.21' ,'9.6.21' , ' grocery' , '10.6.21' , 'sugar-flour’);

CA_ID	NAME	CA_CREATED_AT	CA_MODIFIED_AT	CA_DELETED_AT	DESCRIBE
1	food	1.5.21	2.6.21	5.6.21	rice -bread_butter
2	Stationary	4.5.21	5.6.21	6.6.21	book -Pencil_pen
3	Clothes	8.5.21	9.6.21	10.6.21	dress -shirt_pant
4	Bags	11.5.21	9.6.21	11.6.21	ladiesbag -bagpack
5	grocery	12.5.21	9.6.21	10.6.21	sugar-flour

13.

INSERT INTO product(Pr_ID , name , description , price , Pr_created_at , Pr_modified_at , Pr_deleted_at) VALUES(product_Pr_ID_seq.NEXTVAL, 'Food' , 'rice_bread' , '200' , '4.5.21' ,'5.6.21', '6.6.21');

INSERT INTO product(Pr_ID , name , description , price , Pr_created_at , Pr_modified_at , Pr_deleted_at) VALUES(product_Pr_ID_seq.NEXTVAL, 'Clothes' , 'dress_shirt' , '2000' , '4.6.21' ,'15.6.21', '16.6.21');

INSERT INTO product(Pr_ID , name , description , price , Pr_created_at , Pr_modified_at , Pr_deleted_at) VALUES(product_Pr_ID_seq.NEXTVAL, 'Stationary' , 'Book_pencil' , '200' , '14.5.21' ,'15.5.21', '16.5.21');

INSERT INTO product(Pr_ID , name , description , price , Pr_created_at , Pr_modified_at , Pr_deleted_at) VALUES(product_Pr_ID_seq.NEXTVAL, 'grocery' , 'flour_sugar' , '300' , '14.5.21' ,'5.5.21', '6.5.21');

PR_ID	NAME	DESCRIPTION	PRICE	PR_CREATED_AT	PR_MODIFIED_AT	PR_DELETED_AT
1	Food	rice_bread	700	4.5.21	5.6.21	6.6.21
2	Clothes	dress_shirt	2500	4.6.21	15.6.21	16.6.21
4	Shoe	Boots_sandle	3500	-	-	-
5	Bag	ladiesbag	1000	-	-	-

14.

INSERT INTO inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at) VALUES(inventory_I_ID_seq.NEXTVAL,'10' ,'2.5.21', '11.5.21' , '12.5.21');

INSERT INTO inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at) VALUES(inventory_I_ID_seq.NEXTVAL,'20' ,'12.5.21', '13.5.21' , '14.5.21');

INSERT INTO inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at) VALUES(inventory_I_ID_seq.NEXTVAL,'30' ,'21.5.21', '22.5.21' , '23.5.21');

INSERT INTO inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at) VALUES(inventory_I_ID_seq.NEXTVAL,'25' ,'11.5.21', '12.5.21' , '13.5.21');

INSERT INTO inventory(I_ID, quantity , I_created_at , I_modified_at , I_deleted_at) VALUES(inventory_I_ID_seq.NEXTVAL,'50' ,'5.5.21', '6.5.21' , '8.5.21');

I_ID	QUANTITY	I_CREATED_AT	I_MODIFIED_AT	I_DELETED_AT
1	10	2.5.21	11.5.21	12.5.21
2	20	12.5.21	13.5.21	14.5.21
3	30	21.5.21	22.5.21	23.5.21
4	25	11.5.21	12.5.21	13.5.21
5	50	5.5.21	6.5.21	8.5.21

15.

INSERT INTO discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at) VALUES(discount_Di_ID_seq.NEXTVAL,'Flash_sale' ,'sale_on_food', '20%' , 'yes' ,'13.5.21' , '14.5.21' ,'13.8.21');

INSERT INTO discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at) VALUES(discount_Di_ID_seq.NEXTVAL,'Eid_sale' ,'sale_on_Dress', '10%' , 'No' ,'10.5.21' , '14.5.21' ,'13.6.21');

INSERT INTO discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at) VALUES(discount_Di_ID_seq.NEXTVAL,'Summer_sale' ,'sale_on_food', '10%' , 'yes' ,'13.4.21' , '14.4.21' ,'13.7.21');

INSERT INTO discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at) VALUES(discount_Di_ID_seq.NEXTVAL,'winter_sale' ,'sale_on_clothes', '10%' , 'No' ,'13.12.20' , '14.12.20' ,'13.1.21');

INSERT INTO discount(Di_ID, name , desc1 , parcent , active , Di_created_at , Di_modified_at, Di_deleted_at) VALUES(discount_Di_ID_seq.NEXTVAL,'lockdown_sale' ,'sale_on_food', '10%' , 'No' ,'1.4.21' , '11.4.21' ,'13.5.21');

DI_ID	NAME	DESC1	PARCENT	ACTIVE	DI_CREATED_AT	DI_MODIFIED_AT	DI_DELETED_AT
1	Flash_sale	sale_on_food	20%	yes	13.5.21	14.5.21	13.8.21
2	Eid_sale	sale_on_Dress	10%	No	10.5.21	14.5.21	13.6.21
3	Summer_sale	sale_on_food	10%	yes	13.4.21	14.4.21	13.7.21
4	winter_sale	sale_on_clothes	10%	No	13.12.20	14.12.20	13.1.21
5	lockdown_sale	sale_on_food	10%	No	1.4.21	11.4.21	13.5.21

11.Query writing:

SQL

Single Row:

SELECT UPPER (first_name), INITCAP (last_name) FROM type WHERE rownum < 5;

UPPER(FIRST_NAME)	INITCAP(LAST_NAME)
RAHIM	Islam
KARIM	Islam
BRIM	Stone
SAGE	Battle

SELECT CONCAT (u_first_name, u_last_name) FROM user_customer WHERE rownum < 5;

CONCAT(U_FIRST_NAME,U_LAST_NAME)
DeepFaz
BappyRahman
Nazmullslam
SupanthaBappi

SELECT SUBSTR (first_name, 1, 5), INSTR (first_name, 'a') FROM type WHERE rownum < 5;

SUBSTR(FIRST_NAME,1,5)	INSTR(FIRST_NAME,'A')
Rahim	2
KARIM	0
Brim	0
Sage	2

Group Function:

SELECT COUNT(*) Count FROM type;

COUNT
5

SELECT COUNT (*) FROM type WHERE first_name = 'Rahim';

COUNT(*)
1

SELECT first_name, last_name FROM type ORDER BY first_name DESC;

FIRST_NAME	LAST_NAME
Viper	Den
Sage	Battle
Rahim	Islam
KARIM	ISLAM
Brim	Stone

Sub Query:

SELECT a_id FROM admin WHERE t_id IN (1, 3) ORDER BY a_permissions;

A_ID
1
3

SELECT a_id FROM admin WHERE t_id NOT IN (SELECT t_id FROM type WHERE t_email = '12345') ORDER BY a_permissions;

A_ID
1
4
5
3

DELETE FROM admin WHERE t_id=ANY(SELECT t_id FROM type WHERE t_email='12345');

A_ID	A_PERMISSIONS	A_CREATED_AT	A_MODIFIED_AT	T_ID
1	Admin	-	-	1
3	Manager	-	-	3
4	Manager	-	-	4
5	Manager	-	-	5

Joining:

SELECT A_ID, A_PERMISSIONS FROM Admin, type WHERE admin.a_id = type.T_Id;

A_ID	A_PERMISSIONS
1	Admin
3	Manager
4	Manager
5	Manager

SELECT A_ID, A_PERMISSIONS FROM ADMIN INNER JOIN type ON admin.a_id= type.t_id;

A_ID	A_PERMISSIONS
1	Admin
3	Manager
4	Manager
5	Manager

SELECT A_ID, A_PERMISSIONS FROM ADMIN FULL JOIN TYPE ON admin.a_id = type.T_Id;

A_ID	A_PERMISSIONS
1	Admin
3	Manager
4	Manager
5	Manager
-	-

View:

CREATE VIEW DELIVERY_VIEW AS SELECT d_first_name, d_last_name FROM delivery;

D_FIRST_NAME	D_LAST_NAME
Brian	Chris
Orian	Cris
Chris	Iris
Xris	Rex
Henry	Hen

UPDATE DELIVERY_VIEW SET D_FIRST_NAME = 'Bappa' WHERE d_last_name = 'Rex';

D_FIRST_NAME	D_LAST_NAME
Brian	Chris
Orian	Cris
Chris	Iris
Bappa	Rex
Henry	Hen

DELETE FROM delivery_VIEW WHERE d_first_name = 'Bappa';

D_FIRST_NAME	D_LAST_NAME
Brian	Chris
Orian	Cris
Chris	Iris
Henry	Hen

Synonym:

CREATE SYNONYM deliveries FOR d_email.delivery;

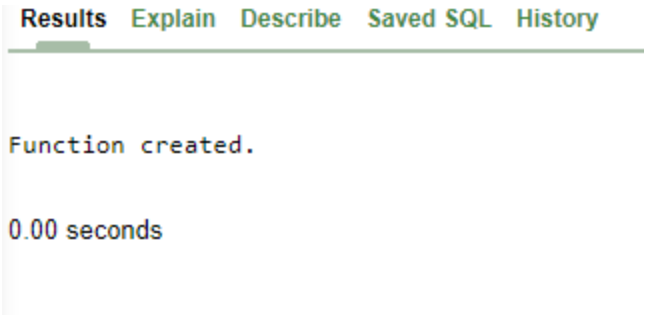
```
CREATE SYNONYM types FOR t_email.type;
CREATE SYNONYM user_customers FOR u_email.user_customer;
```

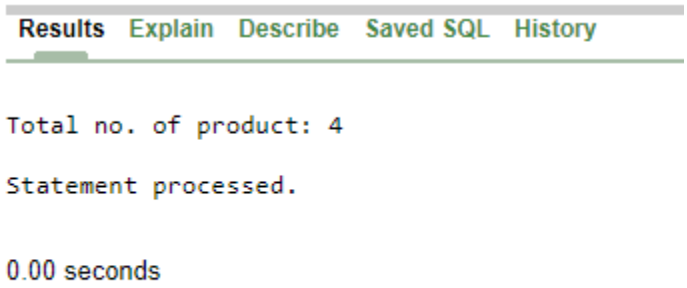
Pl/Sql

Functions

1.

```
 CREATE OR REPLACE FUNCTION totalproduct
RETURN number AS
   total number(2) := 0;
BEGIN
   SELECT count(*) into total
   FROM product;
   RETURN total;
END;

DECLARE
   c number(2);
BEGIN
   c := totalproduct();
   dbms_output.put_line('Total no. of product: ' || c);
END;
```



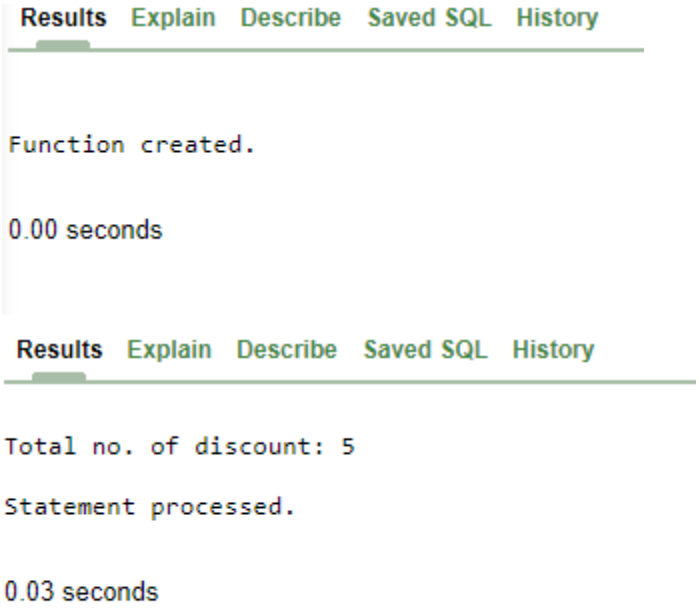


2.

```
CREATE OR REPLACE FUNCTION totaldiscount
RETURN number AS
  total number(2) := 0;
BEGIN
  SELECT count(*) into total
  FROM discount;
  RETURN total;
END;
```

```
DECLARE
  c number(2);
BEGIN
  c := totaldiscount();
```

```
  dbms_output.put_line('Total no. of discount: ' || c);
END;
```



3.

```
CREATE OR REPLACE FUNCTION totaladmin
RETURN number AS
  total number(2) := 0;
BEGIN
  SELECT count(*) into total
  FROM admin;
  RETURN total;
END;

DECLARE
  c number(2);
BEGIN
```

```
  c := totaladmin();
  dbms_output.put_line('Total no. of admin: ' || c);
END;
```


```
Function created.

0.00 seconds
```


```
Total no. of admin: 5

Statement processed.

0.01 seconds
```

2.Procedure

1.

```
  create or replace procedure "Main"
   (U_Id IN NUMBER,
   email IN VARCHAR2)
   is
   begin
   insert into user1 values(U_Id,email);
   end;
   /
```


```
Procedure created.
```

0.22 seconds

2.

```
CREATE OR REPLACE PROCEDURE updateuser1(
    p_U_Id IN user1.U_ID%TYPE,
    p_username IN user1.USERNAME%TYPE)
IS
BEGIN

  UPDATE user1 SET USERNAME = p_username where U_ID = p_u_id;

  COMMIT;

END;
/
```


```
Procedure created.
```

0.22 seconds

```
BEGIN
  updateuser1(4,'manir');
END;
```

U_ID	USERNAME	EMAIL	PASSWORD	MOBILE	FIRST_NAME	LAST_NAME	U_CREATED_AT	U_MODIFIED_AT
1	Ahmed	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
2	fahim	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
3	fahim	fahim@gmail.com	1234567	01873678676	Islam	Fahim	1.5.21	2.6.21
4	manir	mahadi@gmail.com	12356767	01876765632	Haque	Mahadi	9.5.21	11.6.21
5	Ishmam	ishmam@gmail.com	123569090	0171267893	Haque	Ishmam	9.3.21	11.3.21

3.

Drop procedure updateuser1


```
Procedure dropped.
```

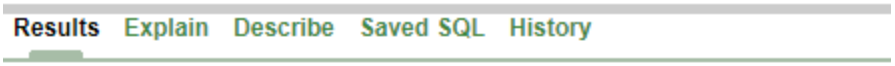
3.31 seconds

Record:

1.

```
DECLARE
  user1_rec user1%rowtype;
BEGIN
  SELECT * into user1_rec
  FROM user1
  WHERE U_Id = 4;
  dbms_output.put_line('user ID: ' || user1_rec.U_Id);
  dbms_output.put_line('user  firstName: ' || user1_rec.first_name);
  dbms_output.put_line('user Email: ' || user1_rec.email);
  dbms_output.put_line('user mobile: ' || user1_rec.mobile);
```

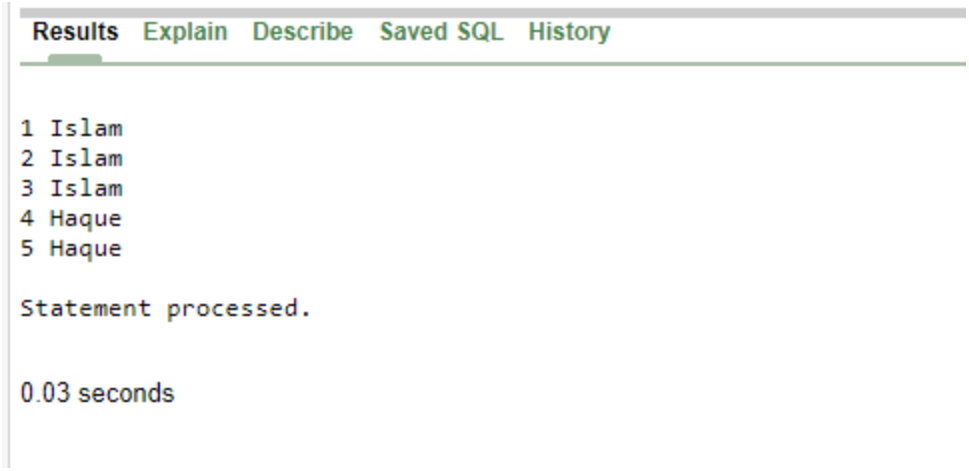
END;




```
user ID: 4
user  firstName: Haque
user Email: mahadi@gmail.com
user mobile: 01876765632

Statement processed.


0.09 seconds
```

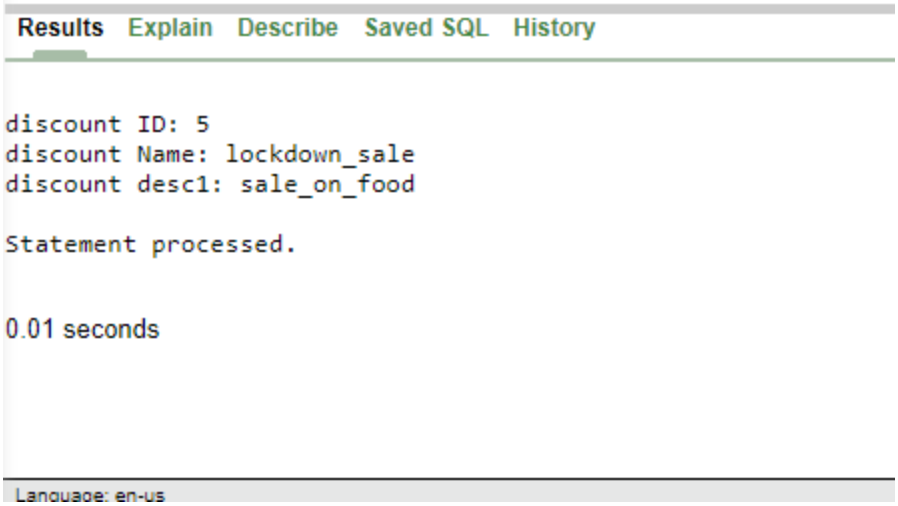
2.

```
 DECLARE
   CURSOR user1_cur is
      SELECT U_Id , first_name , last_name
      FROM user1;
   user1_rec user1_cur%rowtype;
BEGIN
   OPEN user1_cur;
   LOOP
      FETCH user1_cur into user1_rec;
      EXIT WHEN user1_cur%notfound;
      DBMS_OUTPUT.put_line(user1_rec.U_Id || ' ' || user1_rec.first_name);
   END LOOP;
END;
```



3.

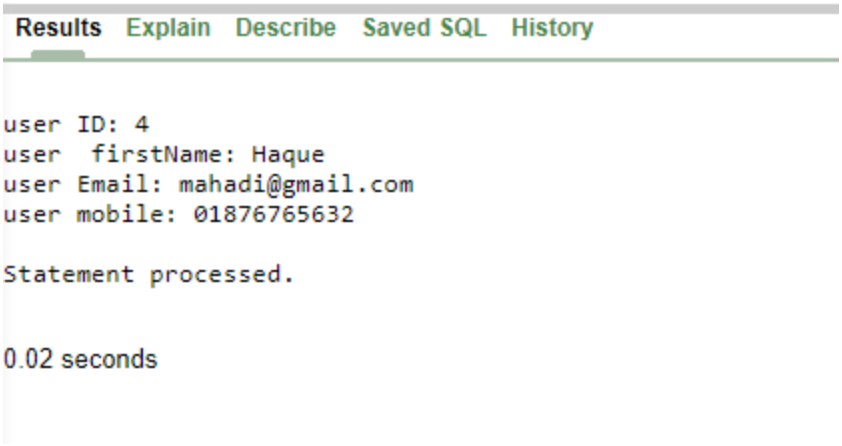
```
 DECLARE
   discount_rec discount%rowtype;
BEGIN
   SELECT * into discount_rec
   FROM discount
   WHERE Di_Id = 5;
   dbms_output.put_line('discount ID: ' || discount_rec.Di_Id);
   dbms_output.put_line('discount Name: ' || discount_rec.name);
   dbms_output.put_line('discount desc1: ' || discount_rec.desc1);

END;
```



Cursor

1.

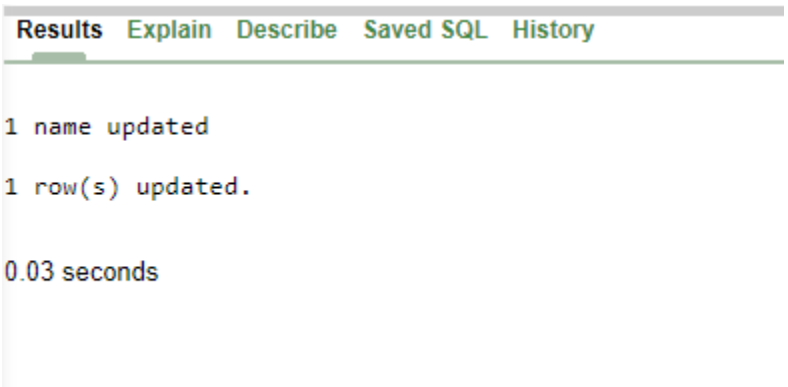
```
 DECLARE
   user1_rec user1%rowtype;
BEGIN
   SELECT * into user1_rec
   FROM user1
   WHERE U_Id = 4;
   dbms_output.put_line('user ID: ' || user1_rec.U_Id);
   dbms_output.put_line('user  firstName: ' || user1_rec.first_name);
   dbms_output.put_line('user Email: ' || user1_rec.email);
   dbms_output.put_line('user mobile: ' || user1_rec.mobile);
END;
```



2.

```
DECLARE
   total_count number(30);
BEGIN

   --updating a row
   UPDATE user1
   SET username= 'Ahmed' where U_Id = 1;

   -- result in boolean, true returned if no rows affected
   IF sql%notfound THEN
      dbms_output.put_line('no subjects fetched');


      -- result in boolean, true returned if any rows affected
      ELSIF sql%found THEN

      -- count the number of rows affected rows affected
      total_count := sql%rowcount;
```

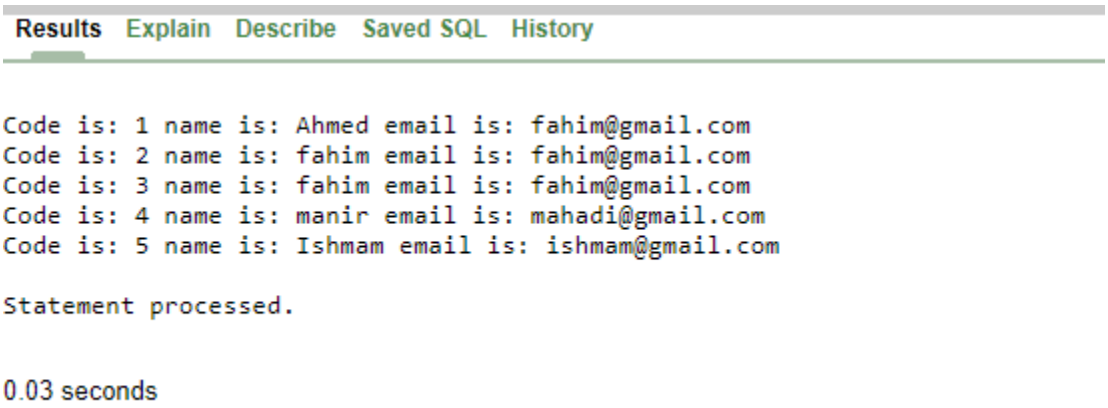
```
        dbms_output.put_line( total_count || ' name updated ');
    END IF;
END;
/
```



3.

```
DECLARE
   -- cursor declaration
CURSOR c2 is
SELECT U_Id , username , email  FROM user1;
U_Id user1.U_Id%type;
username user1.username%type;
 email user1.email%type;
BEGIN

   -- opening a cursor
   OPEN c2;
LOOP

    -- fetching values from cursor
    FETCH c2 into U_Id , username , email;
```

```
    EXIT WHEN c2%notfound;

    -- printing in console
    dbms_output.put_line('Code is: ' || U_Id || ' ' || 'name is: ' || username || ' email is: ' || email);
END LOOP;
CLOSE c2;
END;
/
```



Trigger

1.

```
DECLARE
    total_rows number(2);
  BEGIN
    UPDATE paymentdetails
    SET amount = amount + 5000;
    IF sql%notfound THEN
      dbms_output.put_line('no amount updated');
    ELSIF sql%found THEN
      total_rows := sql%rowcount;
      dbms_output.put_line( total_rows || ' amount updated ');
```

```
    END IF;
  END;
  /
```

2.

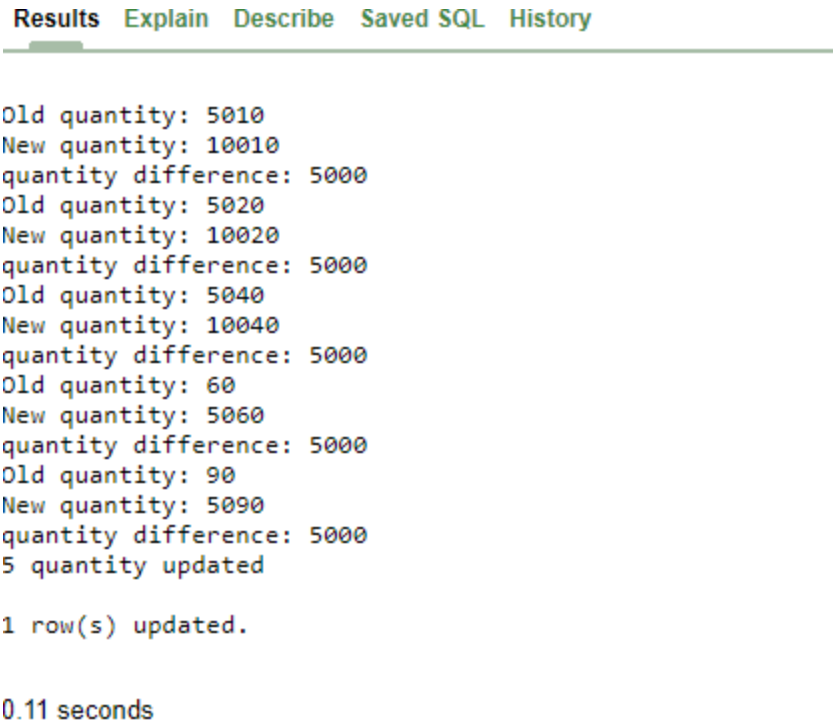
```
CREATE OR REPLACE TRIGGER display_orderdetails_changes
BEFORE DELETE OR INSERT OR UPDATE ON orderdetails
FOR EACH ROW
WHEN (NEW.O_ID > 0)
DECLARE
  quantity_diff number;
BEGIN
 quantity_diff := :NEW.quantity - :OLD.quantity;
  dbms_output.put_line('Old quantity: ' || :OLD.quantity);
  dbms_output.put_line('New quantity: ' || :NEW.quantity);
  dbms_output.put_line('quantity difference: ' || quantity_diff);
END;
```

/


```
Trigger created.

0.24 seconds
```

3.

```
DECLARE
    total_rows number(2);
  BEGIN
    UPDATE orderdetails
    SET quantity = quantity + 5000;
    IF sql%notfound THEN
      dbms_output.put_line('no quantity updated');
    ELSIF sql%found THEN
      total_rows := sql%rowcount;
      dbms_output.put_line( total_rows || ' quantity updated ');
    END IF;
  END;
  /
```

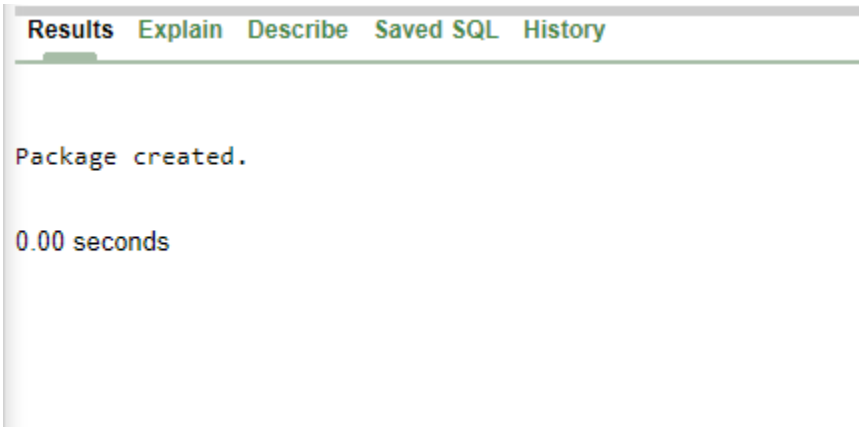


Package

1.

```
CREATE OR REPLACE PACKAGE c_package AS
  -- Adds a product
  PROCEDURE addproduct(c_id   product.Pr_id%type,
  c_name product.Name%type,
  c_description  product.description%type,

  c_price  product.price%type);

  -- Removes a product
  PROCEDURE delproduct(c_id  product.Pr_id%TYPE);
  --Lists all product
```

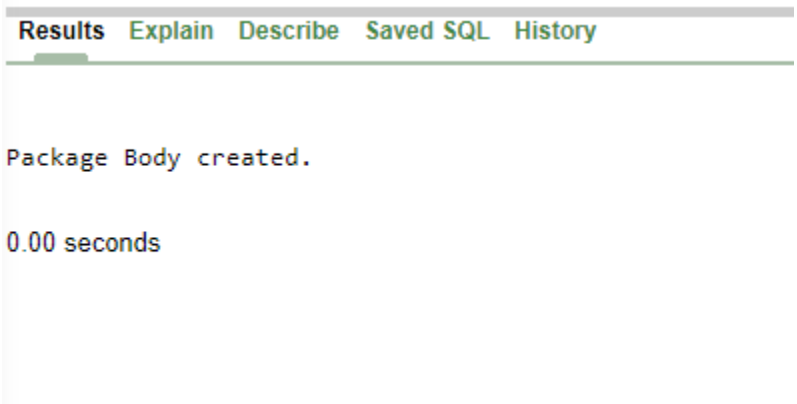
```
  PROCEDURE listproduct;

END c_package;
/
```



2.

```
CREATE OR REPLACE PACKAGE BODY c_package AS
  PROCEDURE addproduct(c_id  product.pr_id%type,
   c_name product.Name%type,
  c_description  product.description%type,

  c_price  product.price%type)

  IS
  BEGIN
    INSERT INTO product (Pr_id , name , description , price)
      VALUES(c_id, c_name, c_description, c_price);
  END addproduct;

  PROCEDURE delproduct(c_id   product.Pr_id%type) IS
  BEGIN
```

```
    DELETE FROM product
    WHERE Pr_id = c_id;
  END delproduct;

  PROCEDURE listproduct IS
  CURSOR c_product is
    SELECT  name FROM product;
  TYPE c_list is TABLE OF product.Name%type;
  name_list c_list := c_list();
  counter integer :=0;
  BEGIN
    FOR n IN c_product LOOP
    counter := counter +1;
    name_list.extend;
    name_list(counter) := n.name;
    dbms_output.put_line('Product(' ||counter|| ')'||name_list(counter));
    END LOOP;
  END listproduct;

END c_package;
/
```



3.

```
DECLARE
  code product.Pr_id%type:= 3;
BEGIN
  c_package.addproduct(6, 'Shoe','Boots_sandle', '3500');
  c_package.addproduct(7, 'Bag',  'ladiesbag', '1000');
  c_package.listproduct;
  c_package.delproduct(code);
  c_package.listproduct;
END;
```

Results Explain Describe Saved SQL History

```
Product(1)Food
Product(2)Clothes
Product(3)Shoe
Product(4)Bag
Product(5)Shoe
Product(6)Bag
Product(1)Food
Product(2)Clothes
Product(3)Shoe
Product(4)Bag
Product(5)Shoe
Product(6)Bag

Statement processed.
```

12. Conclusion: In this project report we are trying to design a shop management system database where user can buy a product from a sales man. There is an admin who has control each and every procedure of this shop management system. we created table for our shop management system by using normalization (up to 3NF). In this shop management system here lot of tables like customer, employee, product, admin, payment table created. Then we insert our data in this table after that we are writing query in our oracle software. In future we want to develop our database so that we can use our shop management system in larger scale.