**SHOP MANAGEMENT SYSTEM**

**ARRAFI AHSAN SOMUDRO**

**Contents:**

1. Introduction……………………………………………………………3
2. Project Proposal……………………………………………………….3
3. Diagram
   1. Class Diagram………………………………………………….4-5
   2. Use Case Diagram……………………………………………...6
   3. Activity Diagram……………………………………………….7
4. User Interface………………………………………………………….8-9
5. Scenario Description………………………………………………….10
6. ER Diagram…………………………………………………………....11
7. Normalization………………………………………………………….12-23
8. Schema Diagram……………………………………………………….24
9. Table Creation…………………………………………………………25-28
10. Data insertion………………………………………………………....28-37
11. Query writing
    1. SQL…………………………………………………………….38-42
    2. PL/SQL………………………………………………………….43-59

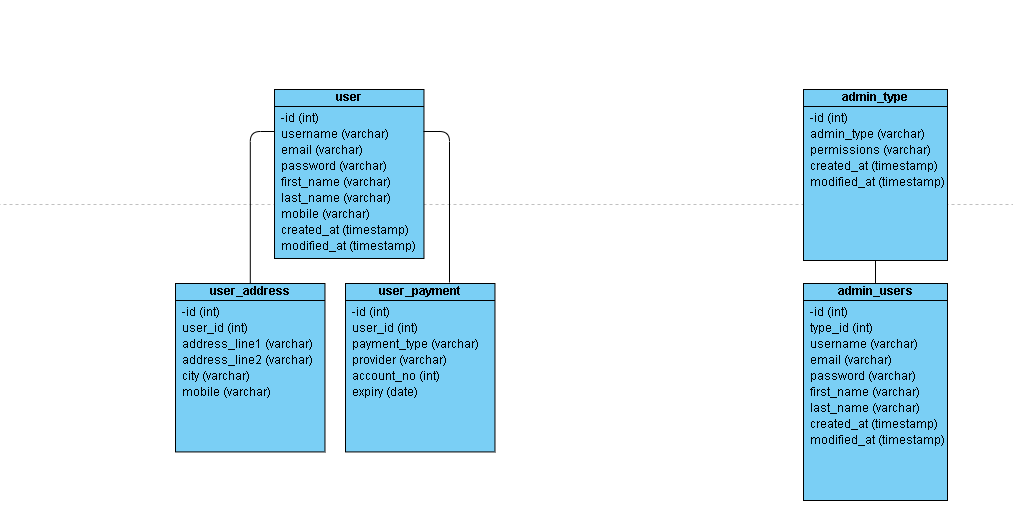
12.Conclusion …………………………………………………………….60

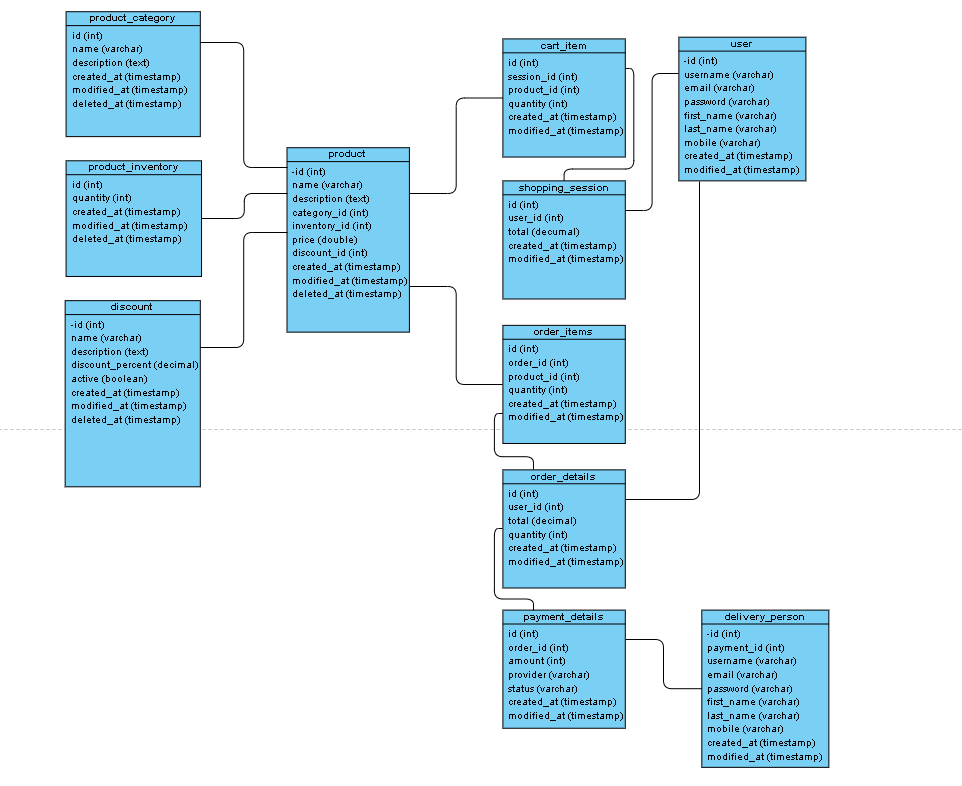
**1. Introduction:**

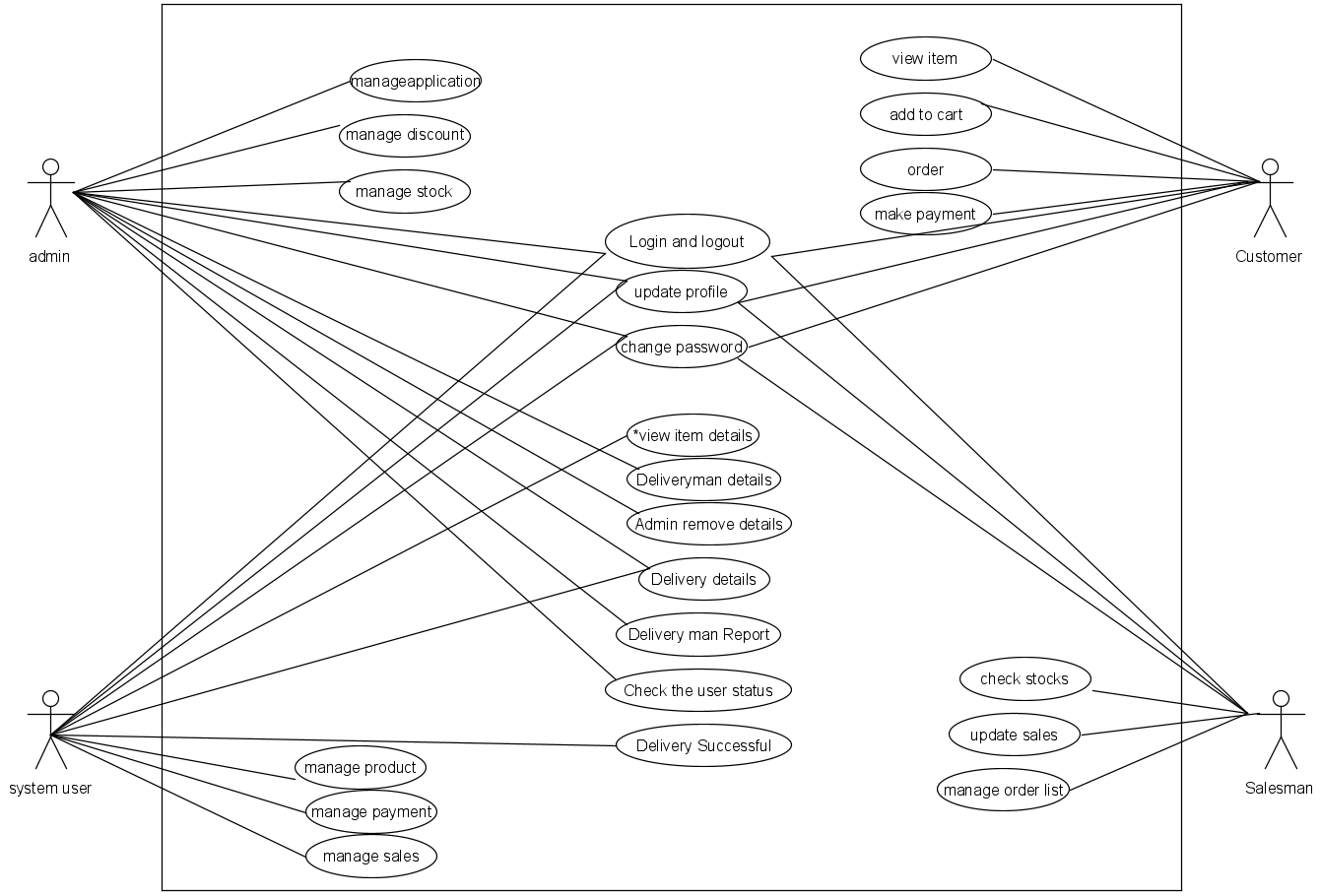
E-Commerce websites are online portals that facilitate online transactions of goods and services through means of the transfer of information and funds over the Internet. We all know the importance of e-commerce, especially in this pandemic situation. There are lots of e-commerce websites available such 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:**

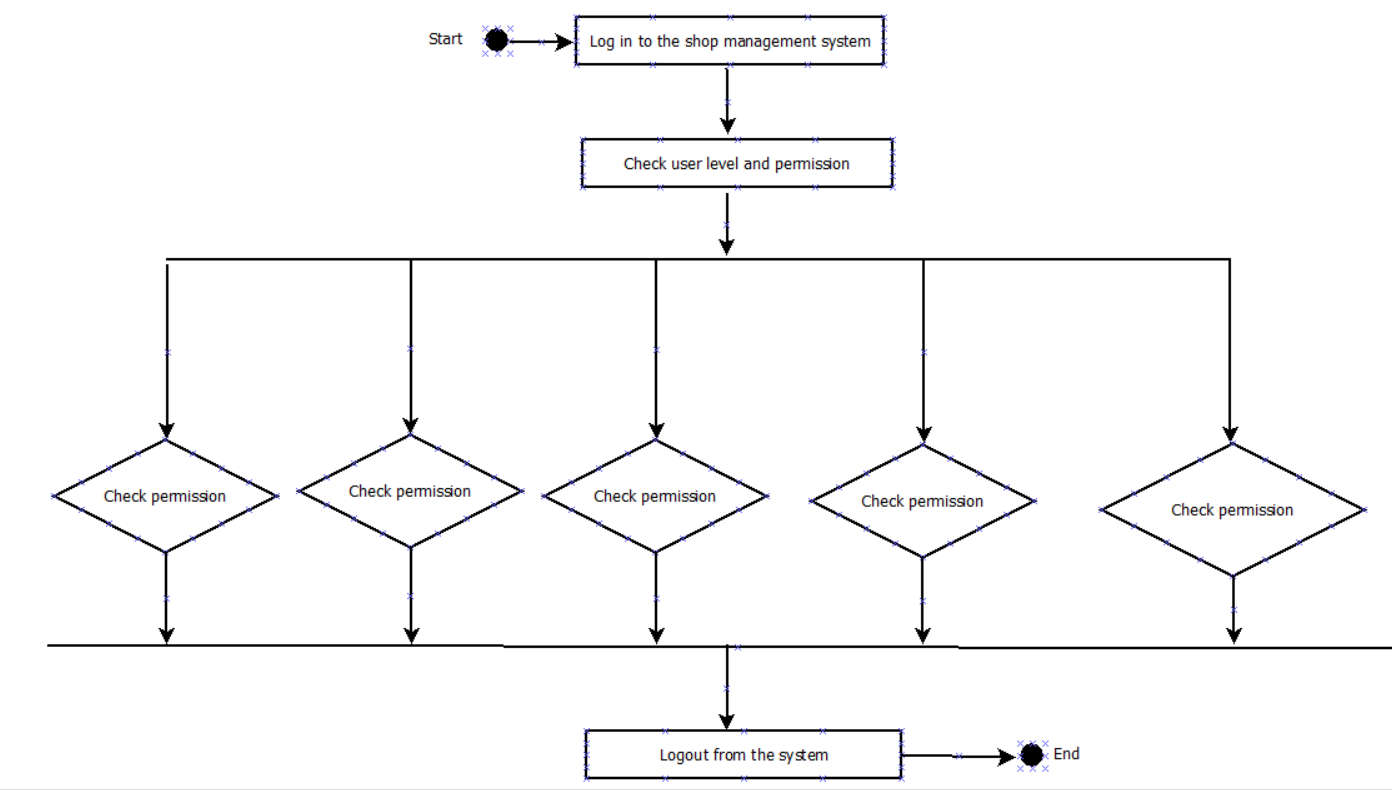


****b) **Use case Diagram:** This use case diagram there are four actor Admin, System user, Customer, Salesman.

**C) Activity Diagram:**

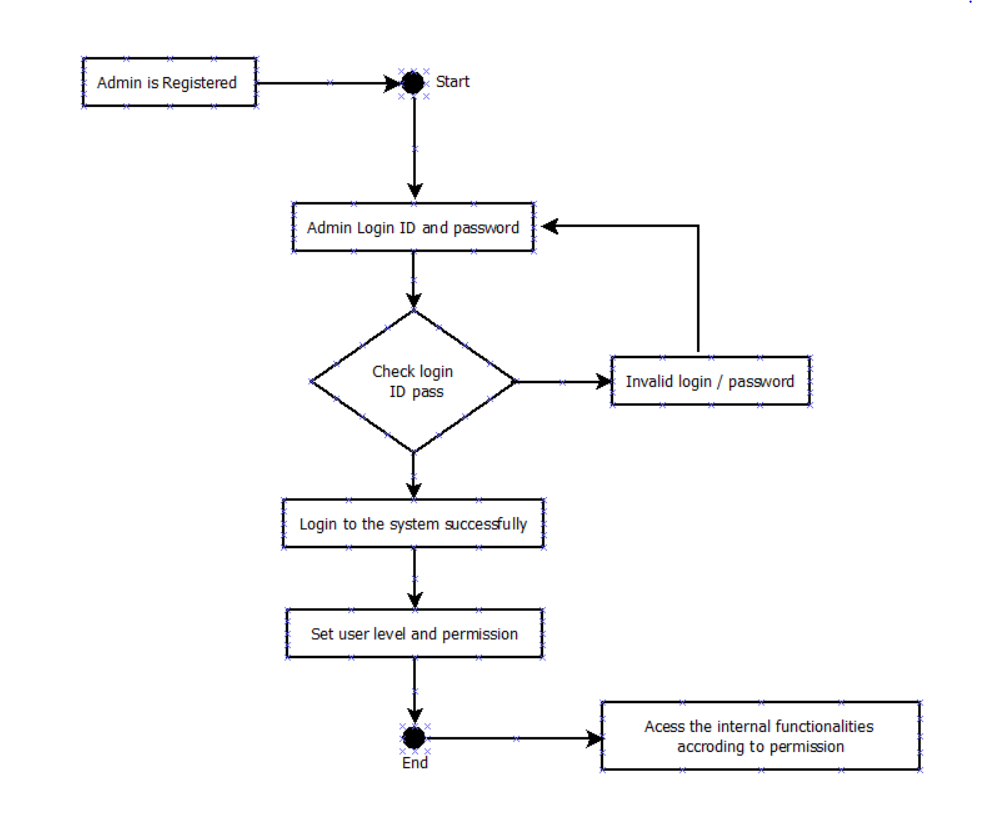
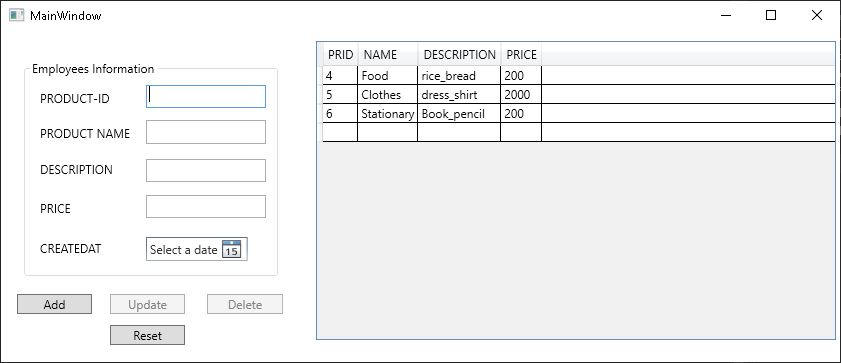
This is activity diagram of shop management system which shows the flow between the activity of sales, stock inventory, payments, discounts, product. The main activity involved in this UMI Activity Diagram of shop management system are as follows:

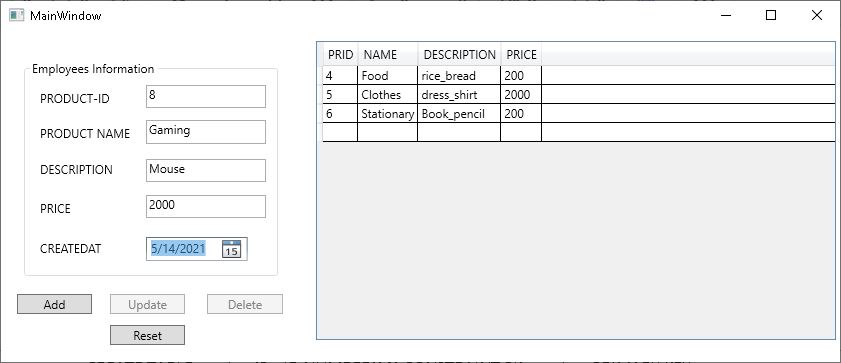
1. Sales Activity  
2.Stock Inventory Activity  
3.Payments Activity  
4.Discount Activity  
5.Product Activity

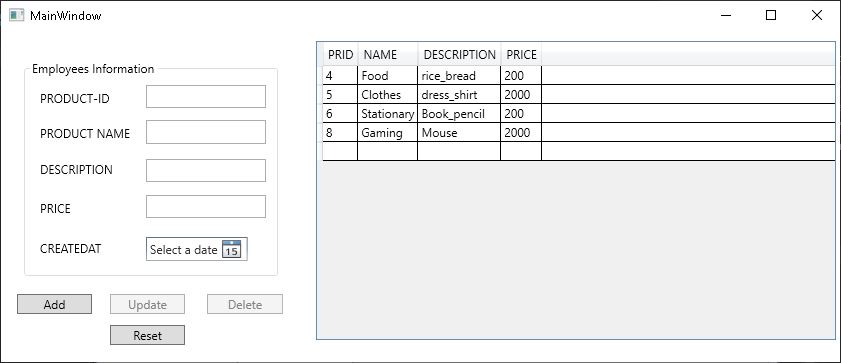
  
**Feature Of the Activity UML Diagram Of shop management system**   
1. Admin user can search sales, view description of a selected sales, add sales, update sales and delete sales  
2. It shows the activity flow of editing, adding and updating of stock inventory  
3. User will be able to search and generate 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

****

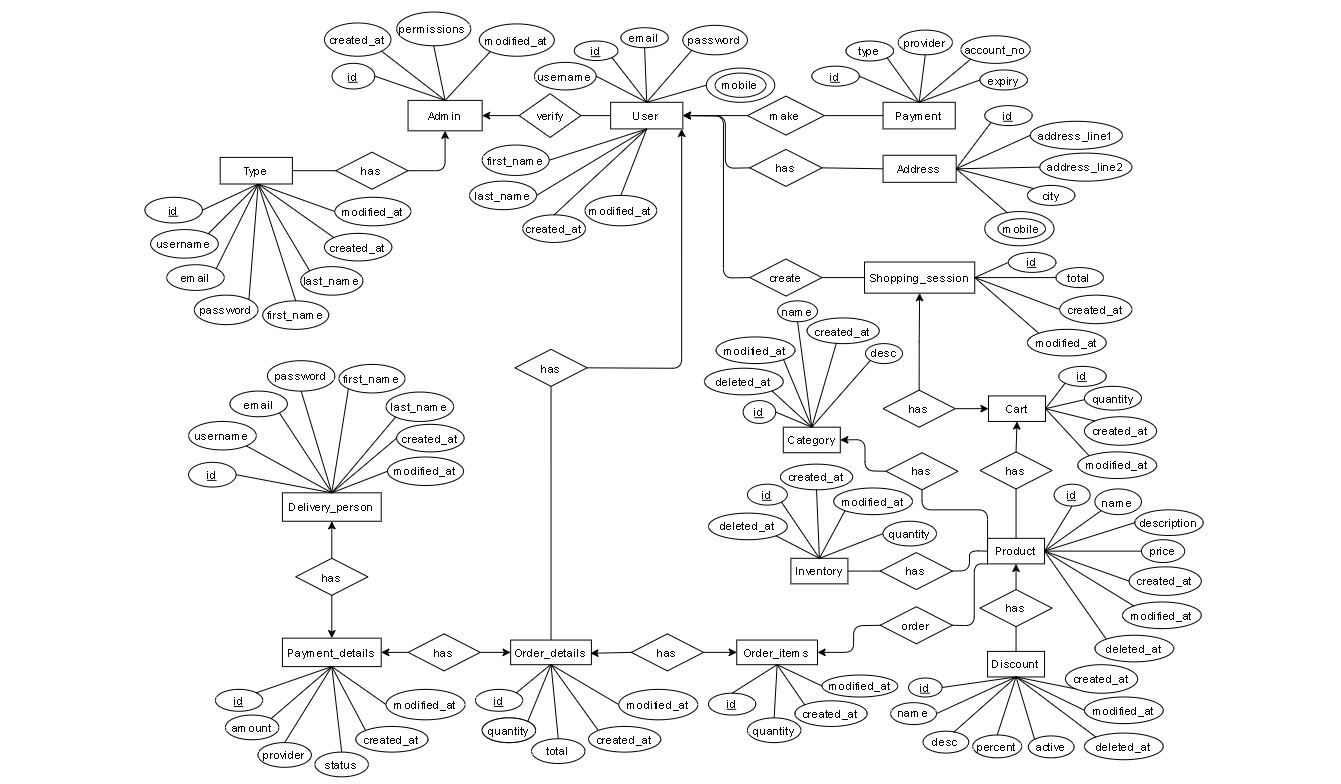




5. **Scenario Description:**

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:**

****

**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

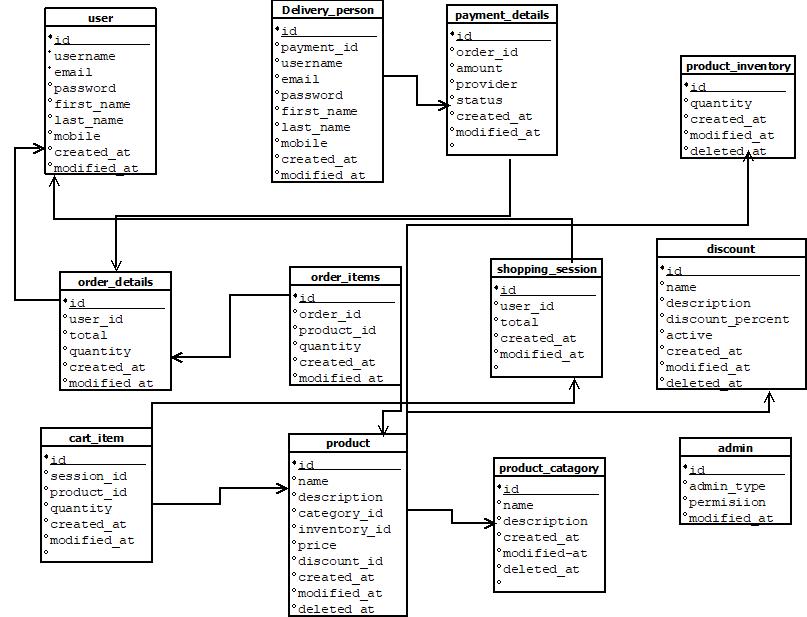
27.Ipr\_id**(**primary key**)** , I\_id**(**primary key**)** , **Pr\_id(**foreign key**)**

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.**

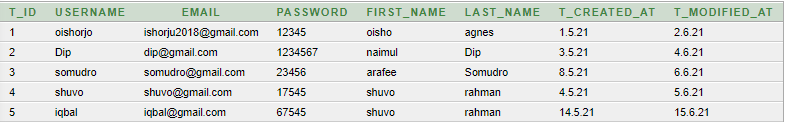
INSERT INTO type(T\_ID, username , email, password , first\_name , Last\_name , T\_created\_at , T\_modified\_at) VALUES(type\_T\_ID\_seq.NEXTVAL,'oishorjo','ishorju2018@gmail.com','12345', 'oisho' , 'agnes' , '1.5.21' ,'2.6.21');

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');



**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');



**3.**

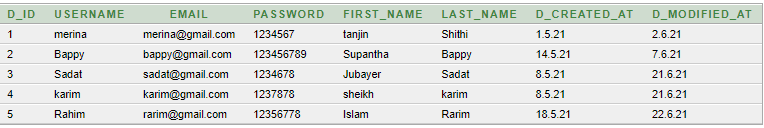
INSERT INTO deliveryperson(D\_ID, username , email, password , first\_name , Last\_name , D\_created\_at , D\_modified\_at) VALUES(deliveryperson\_D\_ID\_seq.NEXTVAL,'merina','merina@gmail.com','1234567', 'tanjin' , 'Shithi' , '1.5.21' ,'2.6.21');

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');



**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

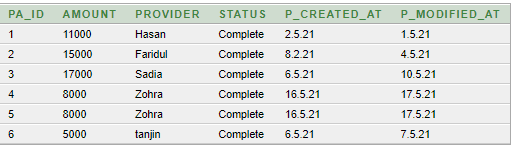
) VALUES(paymentdetails\_Pa\_ID\_seq.NEXTVAL,'7000', 'Sadia' ,'Complete' , '6.5.21', '10.5.21' );

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' );



**5.**

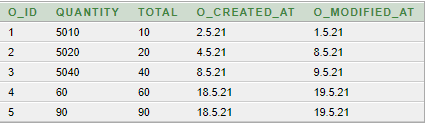
INSERT INTO orderdetails(O\_ID, quantity , total , O\_created\_at , O\_modified\_at) VALUES(orderdetails\_O\_ID\_seq.NEXTVAL,'10' , '10' ,'2.5.21', '1.5.21' );

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' );



**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');



**7.**

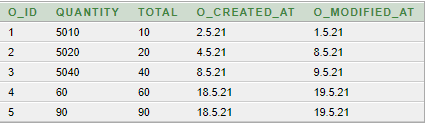
INSERT INTO orderitems(Oi\_ID,oi\_quantity , Oi\_created\_at , Oi\_modified\_at ) VALUES(orderitems\_Oi\_ID\_seq.NEXTVAL,'10' ,'2.5.21', '1.5.21');

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');



**8.**

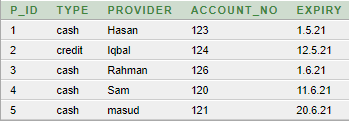
INSERT INTO payment(P\_ID , type , provider , account\_no , expiry ) VALUES(payment\_P\_ID\_seq.NEXTVAL,'cash' ,'Hasan', '123','1.5.21');

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');



**9.**

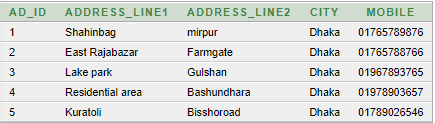
INSERT INTO address(Ad\_ID, address\_line1 , Address\_line2 , city , mobile ) VALUES(address\_Ad\_ID\_seq.NEXTVAL,'Shahinbag','mirpur','Dhaka', '01765789876' );

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' );



**10.**

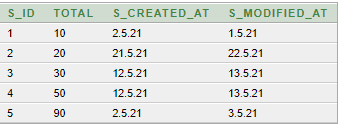
INSERT INTO shoppingsession(S\_ID, total , S\_created\_at , S\_modified\_at) VALUES(shoppingsession\_S\_ID\_seq.NEXTVAL,'10' ,'2.5.21', '1.5.21');

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');



**11.**

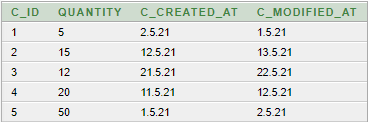
INSERT INTO cart(C\_ID, quantity , C\_created\_at , C\_modified\_at) VALUES(cart\_C\_ID\_seq.NEXTVAL,'5' ,'2.5.21', '1.5.21');

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');



**12.**

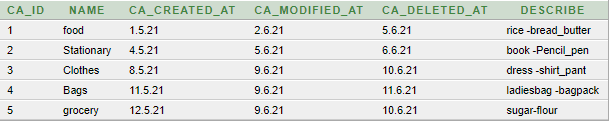
INSERT INTO catagory(Ca\_ID, Ca\_created\_at , Ca\_modified\_at, name , Ca\_deleted\_at , describe ) VALUES(catagory\_Ca\_ID\_seq.NEXTVAL, '1.5.21' ,'2.6.21' , ' food' , '5.6.21' , 'rice -bread\_butter');

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’);



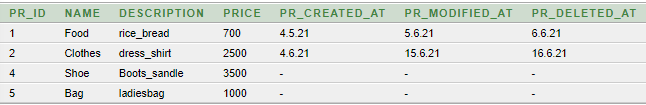
**13.**

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');



**14.**

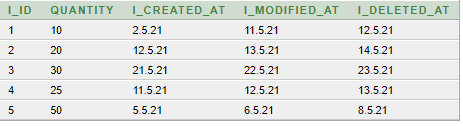
INSERT INTO inventory(I\_ID, quantity , I\_created\_at , I\_modified\_at , I\_deleted\_at) VALUES(inventory\_I\_ID\_seq.NEXTVAL,'10' ,'2.5.21', '11.5.21' , '12.5.21' );

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' );



**15.**

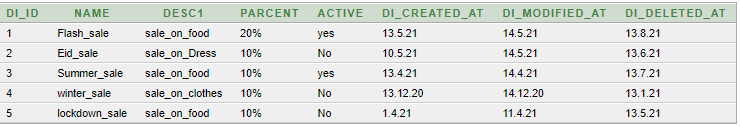
INSERT INTO discount(Di\_ID, name , desc1 , parcent , active , Di\_created\_at , Di\_modified\_at, Di\_deleted\_at ) VALUES(discount\_Di\_ID\_seq.NEXTVAL,'Flash\_sale' ,'sale\_on\_food', '20%' , 'yes' ,'13.5.21' , '14.5.21' ,'13.8.21');

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');



**11.Query writing:**

**SQL**

**Single Row:**

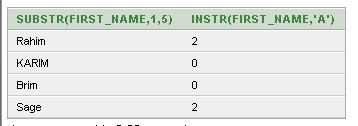
SELECT UPPER (first\_name), INITCAP (last\_name) FROM type WHERE rownum < 5;



SELECT CONCAT (u\_first\_name, u\_last\_name) FROM user\_customer WHERE rownum < 5;



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



**Group Function:**

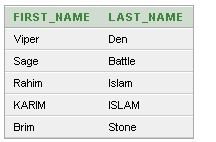
SELECT COUNT(\*) Count FROM type;



SELECT COUNT (\*) FROM type WHERE first\_name = 'Rahim';



SELECT first\_name, last\_name FROM type ORDER BY first\_name DESC;

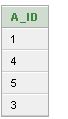


**Sub Query:**

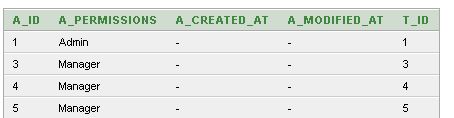
SELECT a\_id FROM admin WHERE t\_id IN (1, 3) ORDER BY a\_permissions;



SELECT a\_id FROM admin WHERE t\_id NOT IN (SELECT t\_id FROM type WHERE t\_email = '12345') ORDER BY a\_permissions;



DELETE FROM admin WHERE t\_id=ANY( SELECT t\_id FROM type WHERE t\_email='12345');



**Joining:**

SELECT A\_ID, A\_PERMISSIONS FROM Admin, type WHERE admin.a\_id = type.T\_Id;



SELECT A\_ID, A\_PERMISSIONS FROM ADMIN INNER JOIN type ON admin.a\_id= type.t\_id;

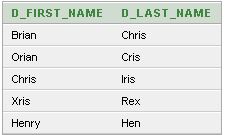


SELECT A\_ID, A\_PERMISSIONS FROM ADMIN FULL JOIN TYPE ON admin.a\_id = type.T\_Id;

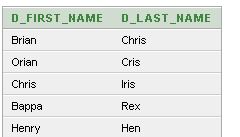


**View:**

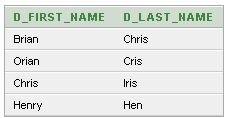
CREATE VIEW DELIVERY\_VIEW AS SELECT d\_first\_name, d\_last\_name FROM delivery;



UPDATE DELIVERY\_VIEW SET D\_FIRST\_NAME = 'Bappa' WHERE d\_last\_name = 'Rex';



DELETE FROM delivery\_VIEW WHERE d\_first\_name = 'Bappa';



**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

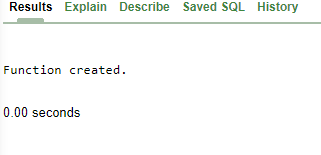
c number(2);

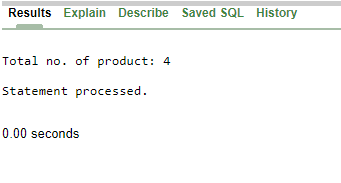
BEGIN

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

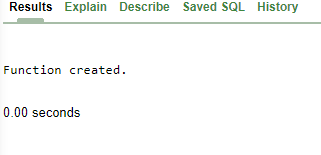
c number(2);

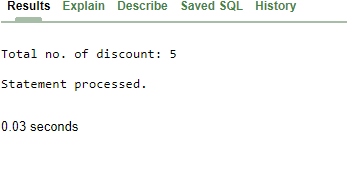
BEGIN

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

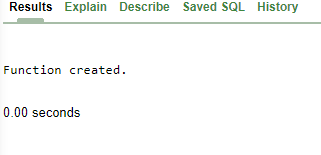
c number(2);

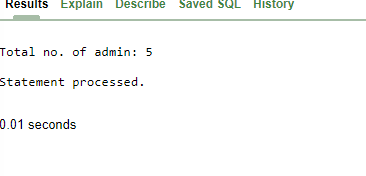
BEGIN

c := totaladmin();

dbms\_output.put\_line('Total no. of admin: ' || c);

END;





**2.Procedure**

**1.**

create or replace procedure "Main"

(U\_Id IN NUMBER,

email IN VARCHAR2)

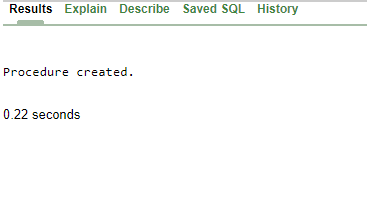
is

begin

insert into user1 values(U\_Id,email);

end;

/



**2.**

CREATE OR REPLACE PROCEDURE updateuser1(

p\_U\_Id IN user1.U\_ID%TYPE,

p\_username IN user1.USERNAME%TYPE)

IS

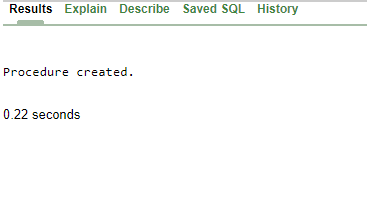
BEGIN

UPDATE user1 SET USERNAME = p\_username where U\_ID = p\_u\_id;

COMMIT;

END;

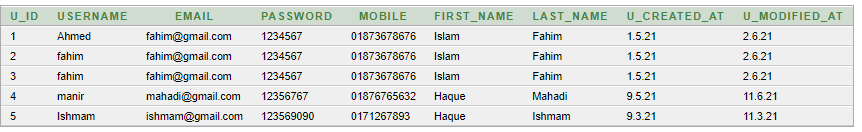
/



BEGIN

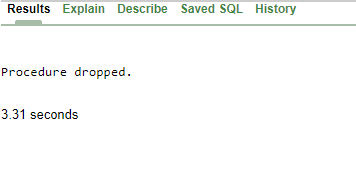
updateuser1(4,'manir');

END;



**3**.

Drop procedure updateuser1



**Record:**

**1.**

DECLARE

user1\_rec user1%rowtype;

BEGIN

SELECT \* into user1\_rec

FROM user1

WHERE U\_Id = 4;

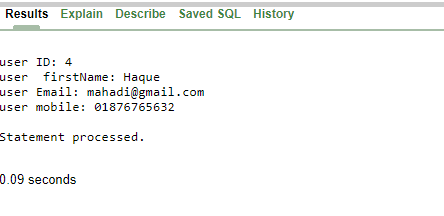
dbms\_output.put\_line('user ID: ' || user1\_rec.U\_Id);

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;



**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

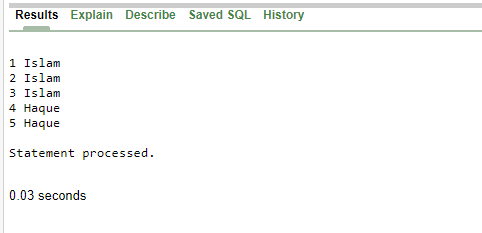
FETCH user1\_cur into user1\_rec;

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

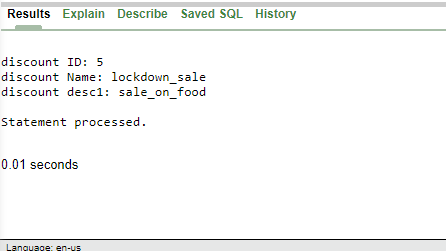
WHERE Di\_Id = 5;

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;

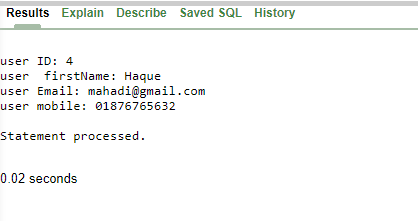
dbms\_output.put\_line('user ID: ' || user1\_rec.U\_Id);

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

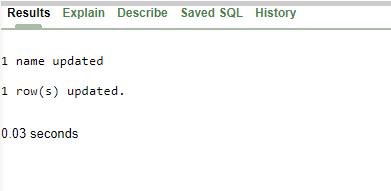
total\_count := sql%rowcount;

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

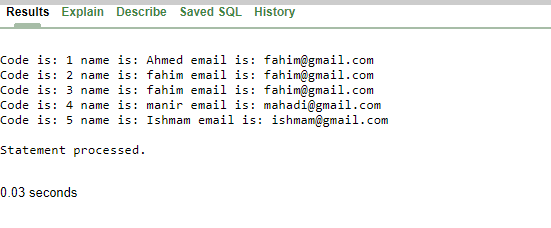
dbms\_output.put\_line('Code is: ' || U\_Id || ' ' || 'name is: ' || username || ' email is: ' || email);

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

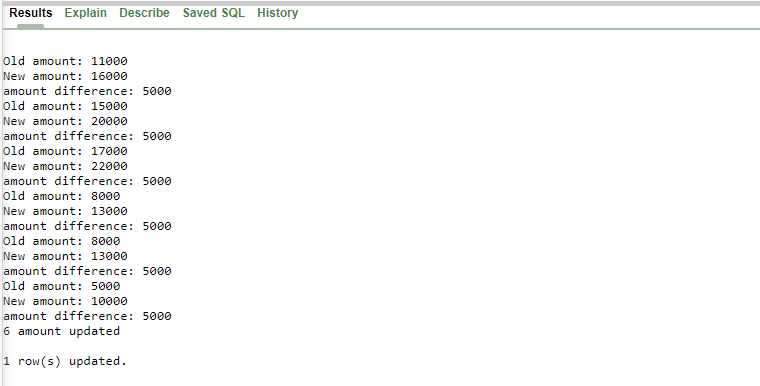
total\_rows := sql%rowcount;

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;

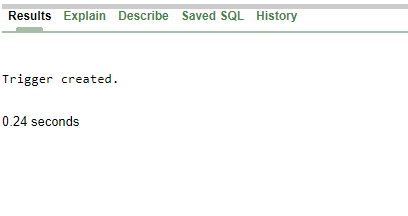
dbms\_output.put\_line('Old quantity: ' || :OLD.quantity);

dbms\_output.put\_line('New quantity: ' || :NEW.quantity);

dbms\_output.put\_line('quantity difference: ' || quantity\_diff);

END;

/



**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

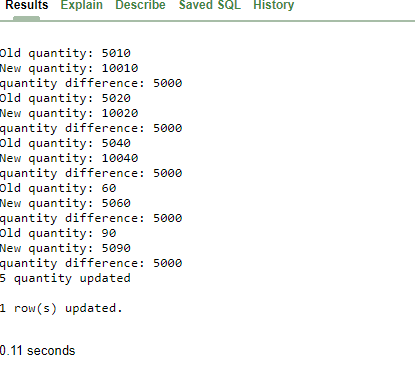
total\_rows := sql%rowcount;

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

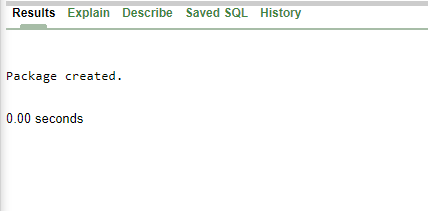
PROCEDURE delproduct(c\_id product.Pr\_id%TYPE);

--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;

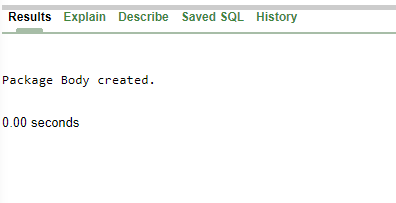
dbms\_output.put\_line('Product(' ||counter|| ')'||name\_list(counter));

END LOOP;

END listproduct;

END c\_package;

/



**3.**

DECLARE

code product.Pr\_id%type:= 3;

BEGIN

c\_package.addproduct(6, 'Shoe','Boots\_sandle', '3500');

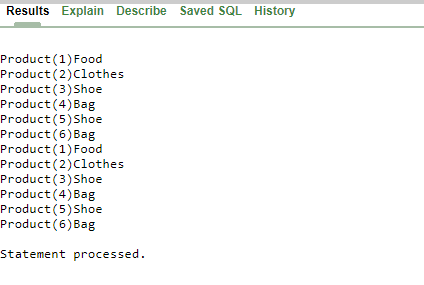
c\_package.addproduct(7, 'Bag', 'ladiesbag', '1000');

c\_package.listproduct;

c\_package.delproduct(code);

c\_package.listproduct;

END;



1. **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.