PASTRY SHOP MANAGEMENT SYSTEM

SL NO.	NAME	STUDENT ID
01	MD. TANVIR SIDDIQUE PRAMANIK	20-43286-1
02	TANZID	20-42057-1
03	ABDULLAH AL MAHADI	20-42044-1
04	SYED MUHAMMAD HASNAT JAMEE	20-42918-1

Introduction to Database (Section-M)

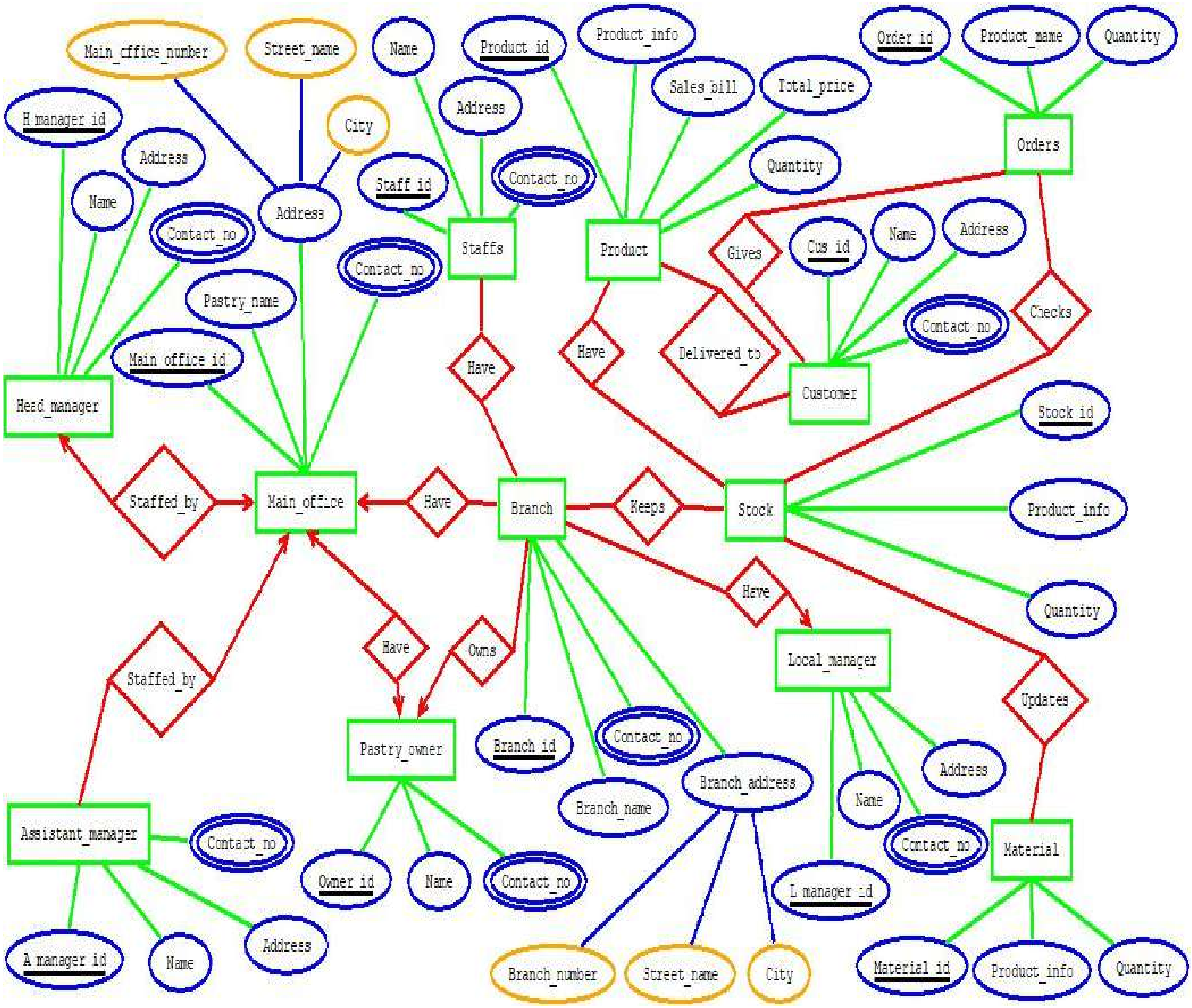
INTRODUCTION:

A database management system (DBMS) is system application for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. In our project we are going to build closer to perfect Pastry Shop Management System. So, first question can be asked, why we need a pastry management system? Well, There's a lot of competition in food businesses today. One wrong ingredient and you can end up making a sugary mess. Here comes our Pastry Shop Management System. Everything from order management to delivery is all taken care of by a Pastry Shop Management System. Operations become more organized and less time-consuming. In this system there are couple of management will be highly focused. Such as, Customer relationship management, Loyalty management, Stuff management, Pastry data reports Order Details, Stock Details, Billing details, Delivery Details, Order Details etc. Here the user can modify and create any type of changes in this shop from anywhere and manage things easily.

SCENARIO:

In a pastry shop management system, a main office may have many branches.A branch may have exactly one main office.A main office is identified by a main office id.The system stores main office id, pastry name, address and contact no of main office.There may be multiple contact no.A main office address is composed of main office number, street name and city.A main office have one pastry owner.A pastry owner have one main office.A pastry owner is identified by a owner id. The system stores owner id, name and contact no of pastry owner.There may be multiple contact no.A main office is staffed by one head manager. A head manager is staffed by one main office.A head manager is identified by a head manager id.Head manager id, Name, address and multiple contact no of head manager are stored by the system.A main office is staffed by many assistant managers. A assistant manager is staffed by one main office.A assistant manager is identified by a assistant manager id. Assistant manager id, Name, address and multiple contact no of assistant managers are stored by the system.A pastry owner owns many branches.Many branches are owned by exactly one pastry owner.A branch is identified by a branch id.The system also stores branch name,branch address and contact no.There may be multiple contact no.A branch address is composed of branch number, street name and city.A branch has one local manager.A local manager have in many branches.A local manager is identified by a local manager id. Local manager id, Name, address and multiple contact no of local managers are stored by the system.A branch have many staffs.A staff have in many branches.A staff is identified by a staff id.Staff id, Name, address and multiple contact no of staffs are stored by the system.A branch may keeps many stocks.A stock may be kept in by many branches.A stock is identified by a stock id. The system stores stock id, quantity and product info of stocks.A stock may have many products. A product may have many stocks.A product is identified by a product id.The system stores product id, sales bill, total price, quantity and product info of products.A product is delivered to many customers.A customer can wants to be delivered to him/her many products.A customer is identified by a customer id.Customer id, name, address and multiple contact no of customer is stored by the system. A customer may gives many orders.A order can be given by many customers.A order is identified by a order id.The system stores order id, product name and quantity of orders.A order checks many stocks.A stock can be checked by many orders.A stock can updates many materials.A material can be updated by many stocks.A material is identified by a material id. The system stores material id, product info and quantity of materials.

ER DIAGRAM:


Main_office_number
Street_name
Name
Product_id
Product_info
Order_id
Product_name
Quantity
H_manager_id
City
Address
Sales_bill
Total_price
Address
Orders
Name
Quantity
Address
Staff_id
Contact_no
Contact_no
Staffs
Product
Gives
Cus_id
Name
Address
Contact_no
Contact_no
Pastry_name
Checks
Main_office_id
Have
Have
Delivered_to
Contact_no
Head_manager
Customer
Stock_id
Staffed_by
Main_office
Have
Branch
Keeps
Stock
Product_info
Have
Quantity
Have
Owns
Local_manager
Staffed_by
Updates
Pastry_owner
Branch_id
Contact_no
Address
Branch_address
Name
Branch_name
Assistant_manager
Contact_no
Contact_no
Material
Owner_id
Name
Contact_no
L_manager_id
Address
A_manager_id
Name
Material_id
Product_info
Quantity
Branch_number
Street_name
City

Normalization:

Have

UNF

Have(Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no, Branch_id, Branch_name, Contact_no, Branch_address, Branch_number, Street_name, City)

1NF

Contact_no is a multi valued attribute.

1. Main_office_id, Pastry_name, Address, Main_office number, Street_name, City, Contact_no, Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

2NF

1. Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

2. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

3NF

1. Main_office_id, Pastry_name, Main_office number, Contact_no

2. Street_name, City

3. Branch_id, Branch_name, Contact_no, Branch_number

4. Street_name, City

Table Creation

1. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**

2. M_O_Add_id, Street_name, City

3. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, Main_office_id**

4. B_Add_id, Street_name, City

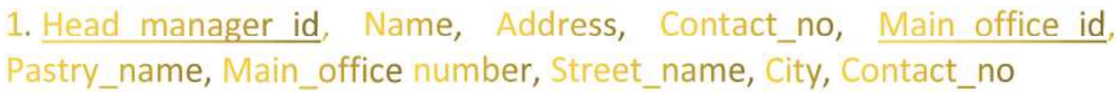
Staffed by

UNF

Staffed_by(Head_manager_id, Name, Address, Contact_no, Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no)

1NF

Contact_no is a multi valued attribute.



1. Head_manager_id, Name, Address, Contact_no, Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

2NF

1. Head_manager_id, Name, Address, Contact_no

2. Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

3NF

1. Head_manager_id, Name, Address, Contact_no

2. Main_office_id, Pastry_name, Main_office number, Contact_no

3. Street_name, City

Table Creation

1. Head_manager_id, Name, Address, Contact_no, **Main_office_id**

2. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**

3. M_O_Add_id**,** Street_name, City

Staffed by

UNF

Staffed_by(A_manager_id, Name, Address, Contact_no, Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no)

1NF

Contact_no is a multi valued attribute.

1. A_manager_id, Name, Address, Contact_no, Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

2NF

1. A_manager_id, Name, Address, Contact_no

2. Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

3NF

1. A_manager_id, Name, Address, Contact_no

2. Main_office_id, Pastry_name, Main_office number, Contact_no

3. Street_name, City

Table Creation

1. A_manager_id, Name, Address, Contact_no, **Main_office_id**

2. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**

3. M_O_Add_id, Street_name, City

Have

UNF

Have(Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no, Owner_id, Name, Contact_no)

1NF

Contact_no is a multi valued attribute.

1. Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no, Owner_id, name, Contact_no

2NF

1. Main_office_id, Pastry_name, Main_office number, Street_name, City, Contact_no

2. Owner_id, name, Contact_no

3NF

1. Main_office_id, Pastry_name, Main_office number, Contact_no
2. Street_name, City

3. Owner_id, name, Contact_no

Table Creation

1. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**

2. M_O_Add_id**,** Street_name, City

3. Owner_id, name, Contact_no, **Main_office_id**

Owns

UNF

Owns(Owner_id, Name, Contact_no, Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City)

1NF

Contact_no is a multi valued attribute.

1 .Owner_id, Name, Contact_no, Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

2NF

1. Owner_id, Name, Contact_no

2. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

3NF

1. Owner_id, Name, Contact_no

2. Branch_id, Branch_name, Contact_no, Branch_number

3. Street_name, City

Table Creation

1. Owner_id, Name, Contact_no

2. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, Owner_id**

3. B_Add_id, Street_name, City

Have

UNF

Have(Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, Staff_id , Name, Address, Contact_no)

1NF

Contact_no is a multi valued attribute.

1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, Staff_id , Name, Address, Contact_no

2NF

1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

2. Staff_id , Name, Address, Contact_no

3NF

1. Branch_id, Branch_name, Contact_no, Branch_number

2. Street_name, City

3. Staff_id , Name, Address, Contact_no

Table Creation

1. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id**

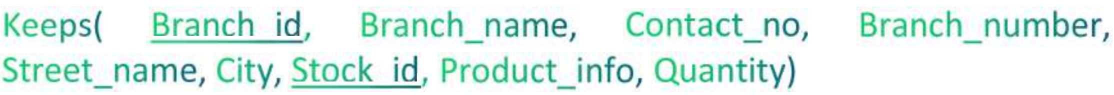
2. B_Add_id , Street_name, City

3. Staff_id , Name, Address, Contact_no

4. **Branch_id , Staff_id**

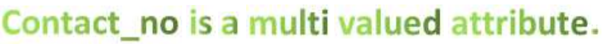
Keeps

UNF

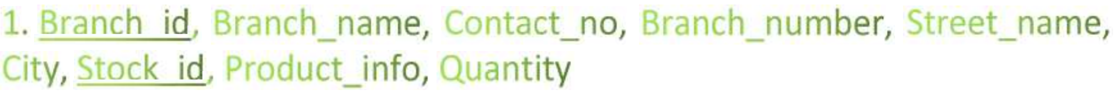


Keeps(Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, Stock_id, Product_info, Quantity)

1NF



Contact_no is a multi valued attribute.



1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, Stock_id, Product_info, Quantity

2NF

1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

2. Stock_id, Product_info, Quantity

3NF

1. Branch_id, Branch_name, Contact_no, Branch_number

2. Street_name, City

3. Stock_id, Product_info, Quantity

Table Creation

1. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id**

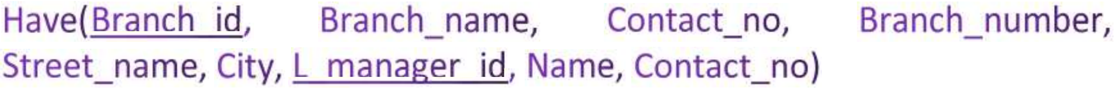
2. B_Add_id , Street_name, City

3. Stock_id, Product_info, Quantity

4. **Branch_id , Stock_id**

Have

UNF



Have(Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, L_manager_id, Name, Contact_no)

1NF

Contact_no is a multi valued attribute.

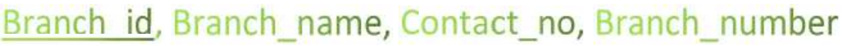
1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City, L_manager_id, Name, Contact_no

2NF

1. Branch_id, Branch_name, Contact_no, Branch_number, Street_name, City

2. L_manager_id, Name, Contact_no

3NF

1. 

Branch_id, Branch_name, Contact_no, Branch_number

2. Street_name, City

3. L_manager_id, Name, Contact_no

Table Creation

1. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, L_manager_id**

2. B_Add_id , Street_name, City

3. L_manager_id, Name, Contact_no

Have

UNF

Have(Product_id, Product_info, Sales_bill, Total_price, Quantity, Stock_id, Product_info, Quantity)

1NF

There is no multi valued attribute. Relation already in 1NF.

1. Product_id, Product_info, Sales_bill, Total_price, Quantity, Stock_id, Product_info, Quantity

2NF

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Stock_id, Product_info, Quantity

3NF

There is no transitive dependency. Relation already in 3NF.

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Stock_id, Product_info, Quantity

Table Creation

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Stock_id, Product_info, Quantity, **Product_id**

Updates

UNF

Updates(Stock_id, Product_info, Quantity, Material_id, Product_info, Quantity)

1NF

There is no multi valued attribute. Relation already in 1NF.

1. Stock_id, Product_info, Quantity, Material_id, Product_info, Quantity

2NF

1. Stock_id, Product_info, Quantity

2. Material_id, Product_info, Quantity

3NF

There is no transitive dependency. Relation already in 3NF.

1. Stock_id, Product_info, Quantity

2. Material_id, Product_info, Quantity

Table Creation

1. Stock_id, Product_info, Quantity

2. Material_id, Product_info, Quantity

3. **Stock_id**, **Material_id**

Checks

UNF

Checks(Stock_id, Product_info, Quantity, Order_id, Product_name, Quantity)

1NF

There is no multi valued attribute. Relation already in 1NF.

1. Stock_id, Product_info, Quantity, Order_id, Product_name, Quantity

2NF

1. Stock_id, Product_info, Quantity

2. Order_id, Product_name, Quantity

3NF

There is no transitive dependency. Relation already in 3NF.

1. Stock_id, Product_info, Quantity

2. Order_id, Product_name, Quantity

Table Creation

1. Stock_id, Product_info, Quantity

2. Order_id, Product_name, Quantity

3. **Stock_id** , **Order_id**

Gives

UNF

Gives(Cus_id, Name, Address, Contact_no, Order_id, Product_ name, Quantity)

1NF

Contact_no is a multi valued attribute.

1. Cus_id, Name, Address, Contact_no, Order_id, Product_ name, Quantity

2NF

1. Cus_id, Name, Address, Contact_no

2. Order_id, Product_ name, Quantity

3NF

There is no transitive dependency. Relation already in 3NF.

1. Cus_id, Name, Address, Contact_no

2. Order_id, Product_ name, Quantity

Table Creation

1. Cus_id, Name, Address, Contact_no

2. Order_id, Product_ name, Quantity

3. **Cus_id , Order_id**

Delivered to

UNF

Delivered_to(Product_id, Product_info, Sales_bill, Total_price, Quantity, Cus_id, Name, Address, Contact_no)

1NF

Contact_no is a multi valued attribute.

1. Product_id, Product_info, Sales_bill, Total_price, Quantity, Cus_id, Name, Address, Contact_no

2NF

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Cus_id, Name, Address, Contact_no

3NF

There is no transitive dependency. Relation already in 3NF.

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Cus_id, Name, Address, Contact_no

Table Creation

1. Product_id, Product_info, Sales_bill, Total_price, Quantity

2. Cus_id, Name, Address, Contact_no

3. **Product_id , Cus_id**

Temporary Tables

~~1. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**~~

~~2. M_O_Add_id, Street_name, City~~

3. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, Main_office_id**

~~4. B_Add_id, Street_name, City~~

5. Head_manager_id, Name, Address, Contact_no, **Main_office_id**

~~6. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**~~

~~7. M_O_Add_id, Street_name, City~~

8. A_manager_id, Name, Address, Contact_no, **Main_office_id**

~~9. Main_office_id, Pastry_name, Main_office number, Contact_no, **M_O_Add_id**~~

~~10. M_O_Add_id, Street_name, City~~

11. Main_office_id, Pastry_name, Main_office_number, Contact_no, **M_O_Add_id**

12. M_O_Add_id**,** Street_name, City

13. Owner_id, name, Contact_no, **Main_office_id**

~~14. Owner_id, Name, Contact_no~~

15. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, Owner_id**

~~16. B_Add_id, Street_name, City~~

~~17. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id**~~

~~18. B_Add_id , Street_name, City~~

19. Staff_id , Name, Address, Contact_no

20. **Branch_id , Staff_id**

~~21. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id**~~

~~22. B_Add_id , Street_name, City~~

23. Stock_id, Product_info, Quantity

24. **Branch_id , Stock_id**

25. Branch_id, Branch_name, Contact_no, Branch_number, **B_Add_id, L_manager_id**

26. B_Add_id , Street_name, City

27. L_manager_id, Name, Address, Contact_no

~~28. Product_id, Product_info, Sales_bill, Total_price, Quantity~~

29. Stock_id, Product_info, Quantity, **Product_id**

~~30. Stock_id, Product_info, Quantity~~

31. Material_id, Product_info, Quantity

32. **Stock_id, Material_id**

~~33. Stock_id, Product_info, Quantity~~

~~34. Order_id, Product_name, Quantity~~

35. **Stock_id** , **Order_id**

~~36. Cus_id, Name, Address, Contact_no~~

37. Order_id, Product_ name, Quantity

38. **Cus_id , Order_id**

39. Product_id, Product_info, Sales_bill, Total_price, Quantity

40. Cus_id, Name, Address, Contact_no

41. **Product_id , Cus_id**

Final Tables

1. Branch_id, Branch_name, Contact_no1, Contact_no2 , Branch_number, **B_Add_id, Main_office_id**

2. Head_manager_id, Name, Address,Contact_no1, Contact_no2, **Main_office_id**

3. A_manager_id, Name, Address, Contact_no1, Contact_no2, **Main_office_id**

4. Main_office_id, Pastry_name, Main_office_number, Contact_no1, Contact_no2, **M_O_Add_id**

5. M_O_Add_id**,** Street_name, City

6. Owner_id, name, Contact_no1, Contact_no2, **Main_office_id**

7. Branch_id, Branch_name, Contact_no1, Contact_no2, Branch_number, **B_Add_id, Owner_id**

8. Staff_id , Name, Address, Contact_no1, Contact_no2

9. **Branch_id , Staff_id**

10. **Branch_id , Stock_id**

11. Branch_id, Branch_name, Contact_no1, Contact_no2, Branch_number, **B_Add_id, L_manager_id**

12. B_Add_id , Street_name, City

13. L_manager_id, Name, Address, Contact_no1, Contact_no2

14. Stock_id, Product_info, Quantity, **Product_id**

15. Material_id, Product_info, Quantity

16. **Stock_id, Material_id**

17. **Stock_id** , **Order_id**

18. Order_id, Product_ name, Quantity

19. **Cus_id , Order_id**

20. Product_id, Product_info, Sales_bill, Total_price, Quantity

21. Cus_id, Name, Address,Contact_no1, Contact_no2

22. **Product_id , Cus_id**

Schema Diagram :

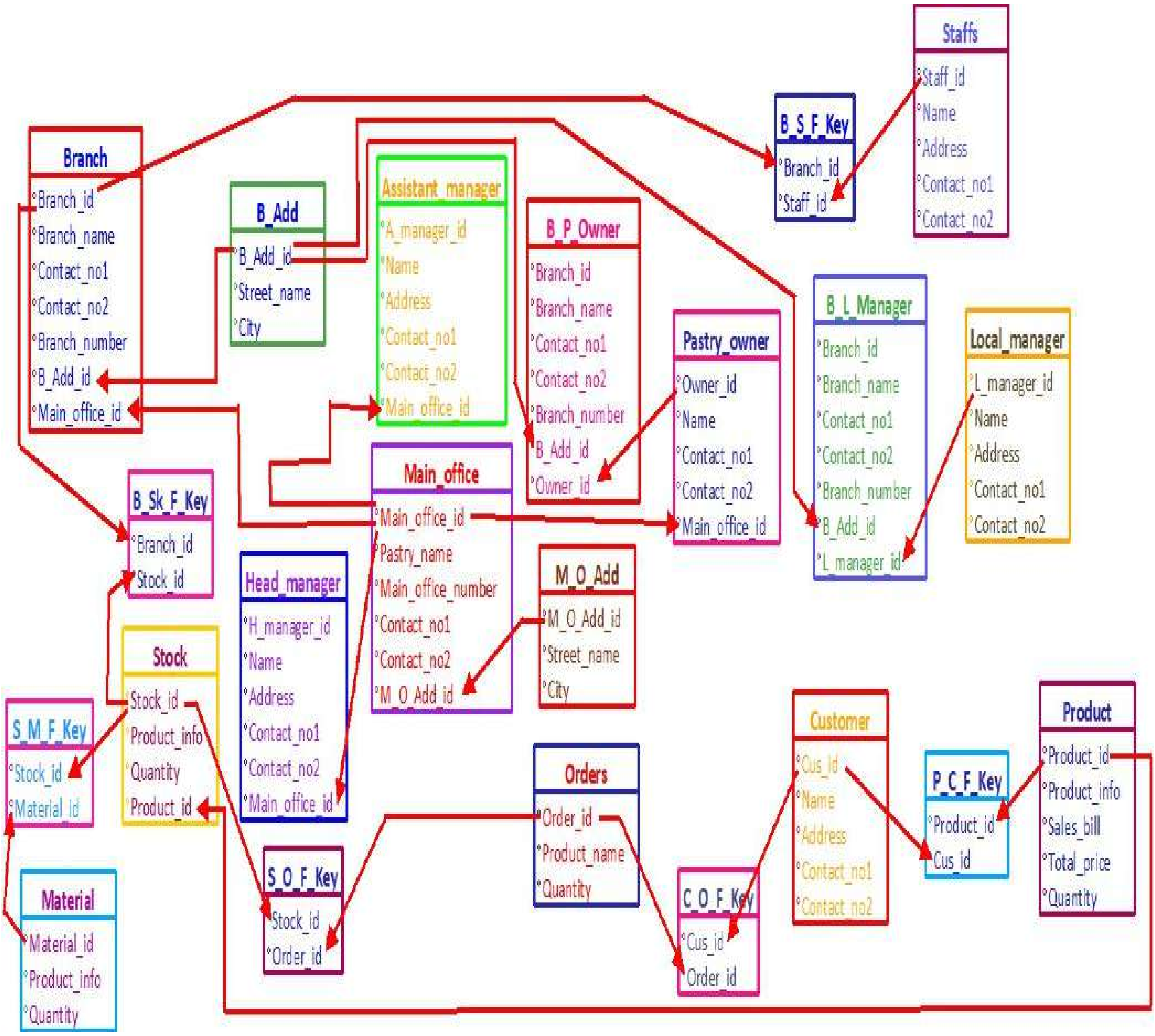


Table Creation:

1. CREATE TABLE Branch (Branch_id NUMBER(10) PRIMARY KEY, Branch_name VARCHAR2(20), Contact_no1 NUMBER(11), Contact_no2 NUMBER(11) , Branch_number NUMBER(10), B_Add_id NUMBER(10), Main_office_id NUMBER(10));

2. CREATE TABLE Head_manager (Head_manager_id NUMBER(10) PRIMARY KEY, Name VARCHAR2(20), Address VARCHAR2(30),Contact_no1 NUMBER(11), Contact_no2 NUMBER(11), Main_office_id NUMBER(10));

3. CREATE TABLE Assistant_manager (A_manager_id NUMBER(10) PRIMARY KEY, Name VARCHAR2(20), Address VARCHAR2(30),Contact_no1 NUMBER(11), Contact_no2 NUMBER(11), Main_office_id NUMBER(10));

4. CREATE TABLE Main_office(Main_office_id NUMBER(10) PRIMARY KEY, Pastry_name VARCHAR2(20), Main_office_number NUMBER(10), Contact_no1 NUMBER(11), Contact_no2 NUMBER(11), M_O_Add_id NUMBER(10));

5. CREATE TABLE M_O_Add(M_O_Add_id NUMBER(10) PRIMARY KEY, Street_name VARCHAR2(20), City VARCHAR(20));

6. CREATE TABLE Pastry_owner(Owner_id NUMBER(10), name VARCHAR2(20), Contact_no1 NUMBER(11), Contact_no2 NUMBER(11), Main_office_id NUMBER(10));

7. CREATE TABLE B_P_Owner (Branch_id NUMBER(10) PRIMARY KEY, Branch_name VARCHAR2(20), Contact_no1 NUMBER(11), Contact_no2 NUMBER(11) , Branch_number NUMBER(10), B_Add_id NUMBER(10), Owner_id NUMBER(10));

8. CREATE TABLE Staffs(Staff_id NUMBER(10) PRIMARY KEY, Name VARCHAR2(20), Address VARCHAR2(30),Contact_no1 NUMBER(11), Contact_no2 NUMBER(11));

9. CREATE TABLE B_S_F_Key(Branch_id NUMBER(10), Staff_id NUMBER(10));

10. CREATE TABLE B_Sk_F_Key(Branch_id NUMBER(10), Stock_id NUMBER(10));

11. CREATE TABLE B_L_Manager (Branch_id NUMBER(10) PRIMARY KEY, Branch_name VARCHAR2(20), Contact_no1 NUMBER(11), Contact_no2 NUMBER(11) , Branch_number NUMBER(10), B_Add_id NUMBER(10), L_manager_id NUMBER(10));

12. CREATE TABLE B_Add(B_Add_id NUMBER(10) PRIMARY KEY , Street_name VARCHAR2(20), City VARCHAR2(20));

13. CREATE TABLE Local_manager (L_manager_id NUMBER(10) PRIMARY KEY, Name VARCHAR2(20), Address VARCHAR2(30),Contact_no1 NUMBER(11), Contact_no2 NUMBER(11));

14. CREATE TABLE Stock(Stock_id NUMBER(10) PRIMARY KEY,Product_info VARCHAR2(50), Quantity VARCHAR2(10), Product_id NUMBER(10));

15. CREATE TABLE Material(Material_id NUMBER(10) PRIMARY KEY,Product_info VARCHAR2(50), Quantity VARCHAR2(10));

16. CREATE TABLE S_M_F_Key(Stock_id NUMBER(10), Material_id NUMBER(10));

17. CREATE TABLE S_O_F_Key(Stock_id NUMBER(10), Order_id NUMBER(10));

18. CREATE TABLE Orders(Order_id NUMBER(10) PRIMARY KEY,Product_name VARCHAR2(50), Quantity VARCHAR2(10));

19. CREATE TABLE C_O_F_Key(Cus_id NUMBER(10), Order_id NUMBER(10));

20. CREATE TABLE Product(Product_id NUMBER(10) PRIMARY KEY, Product_info VARCHAR2(50), Sales_bill NUMBER (6,4), Total_price NUMBER(9,5), Quantity VARCHAR2(10));

21. CREATE TABLE Customer (Cus_id NUMBER(10) PRIMARY KEY, Name VARCHAR2(20), Address VARCHAR2(30),Contact_no1 NUMBER(11), Contact_no2 NUMBER(11));

22. CREATE TABLE P_C_F_Key(Product_id NUMBER(10), Cus_id NUMBER(10));

1. ALTER TABLE Branch ADD CONSTRAINT fk1 FOREIGN KEY (B_Add_id) REFERENCES B_Add(B_Add_id);

2. ALTER TABLE Branch ADD CONSTRAINT fk2 FOREIGN KEY (Main_office_id) REFERENCES Main_office(Main_office_id);

3. ALTER TABLE Head_manager ADD CONSTRAINT fk3 FOREIGN KEY (Main_office_id) REFERENCES Main_office(Main_office_id);

4. ALTER TABLE Assistant_manager ADD CONSTRAINT fk4 FOREIGN KEY (Main_office_id) REFERENCES Main_office(Main_office_id);

5. ALTER TABLE Main_office ADD CONSTRAINT fk5 FOREIGN KEY (M_O_Add_id) REFERENCES M_O_Add(M_O_Add_id);

6. ALTER TABLE Pastry_owner ADD CONSTRAINT fk6 FOREIGN KEY (Main_office_id) REFERENCES Main_office(Main_office_id);

7. ALTER TABLE B_P_owner ADD CONSTRAINT fk7 FOREIGN KEY (B_Add_id) REFERENCES B_Add(B_Add_id);

8. ALTER TABLE B_S_F_Key ADD CONSTRAINT prim1 PRIMARY KEY (Branch_id,Staff_id);

9. ALTER TABLE B_S_F_Key ADD CONSTRAINT fk8 FOREIGN KEY (Branch_id) REFERENCES Branch(Branch_id);

10. ALTER TABLE B_S_F_Key ADD CONSTRAINT fk9 FOREIGN KEY (Staff_id) REFERENCES Staffs(Staff_id);

11. ALTER TABLE B_Sk_F_Key ADD CONSTRAINT prim2 PRIMARY KEY (Branch_id,Stock_id);

12. ALTER TABLE B_Sk_F_Key ADD CONSTRAINT fk10 FOREIGN KEY (Branch_id) REFERENCES Branch(Branch_id);

13. ALTER TABLE B_Sk_F_Key ADD CONSTRAINT fk11 FOREIGN KEY (Stock_id) REFERENCES Stock(Stock_id);

14. ALTER TABLE B_L_Manager ADD CONSTRAINT fk12 FOREIGN KEY (B_Add_id) REFERENCES B_Add(B_Add_id);

15. ALTER TABLE B_L_Manager ADD CONSTRAINT fk13 FOREIGN KEY (L_manager_id) REFERENCES Local_manager(L_manager_id);

16. ALTER TABLE Stock ADD CONSTRAINT fk14 FOREIGN KEY (Product_id) REFERENCES Product(Product_id);

17. ALTER TABLE S_M_F_Key ADD CONSTRAINT prim3 PRIMARY KEY (Stock_id,Material_id);

18. ALTER TABLE S_M_F_Key ADD CONSTRAINT fk15 FOREIGN KEY (Stock_id) REFERENCES Stock(Stock_id);

19. ALTER TABLE S_M_F_Key ADD CONSTRAINT fk16 FOREIGN KEY (Material_id) REFERENCES Material(Material_id);

20. ALTER TABLE S_O_F_Key ADD CONSTRAINT prim4 PRIMARY KEY (Stock_id,Order_id);

21. ALTER TABLE S_O_F_Key ADD CONSTRAINT fk17 FOREIGN KEY (Stock_id) REFERENCES Stock(Stock_id);

22. ALTER TABLE S_O_F_Key ADD CONSTRAINT fk18 FOREIGN KEY (Order_id) REFERENCES Orders(Order_id);

23. ALTER TABLE C_O_F_Key ADD CONSTRAINT prim5 PRIMARY KEY (Cus_id,Order_id);

24. ALTER TABLE C_O_F_Key ADD CONSTRAINT fk19 FOREIGN KEY (Cus_id) REFERENCES Customer(Cus_id);

25. ALTER TABLE C_O_F_Key ADD CONSTRAINT fk20 FOREIGN KEY (Order_id) REFERENCES Orders(Order_id);

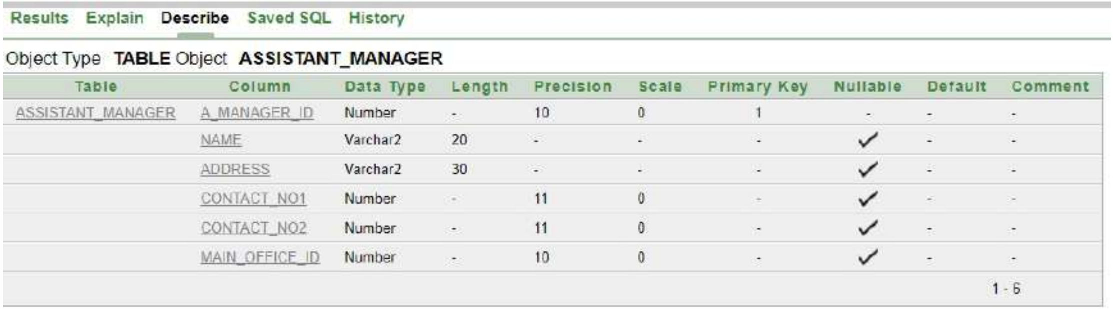
26. ALTER TABLE P_C_F_Key ADD CONSTRAINT prim6 PRIMARY KEY (Product_id,Cus_id);

27. ALTER TABLE P_C_F_Key ADD CONSTRAINT fk21 FOREIGN KEY (Product_id) REFERENCES Product(Product_id);

28. ALTER TABLE P_C_F_Key ADD CONSTRAINT fk22 FOREIGN KEY (Cus_id) REFERENCES Customer(Cus_id);

Screenshots of created table using describe command :

1. Assistant_manager table:

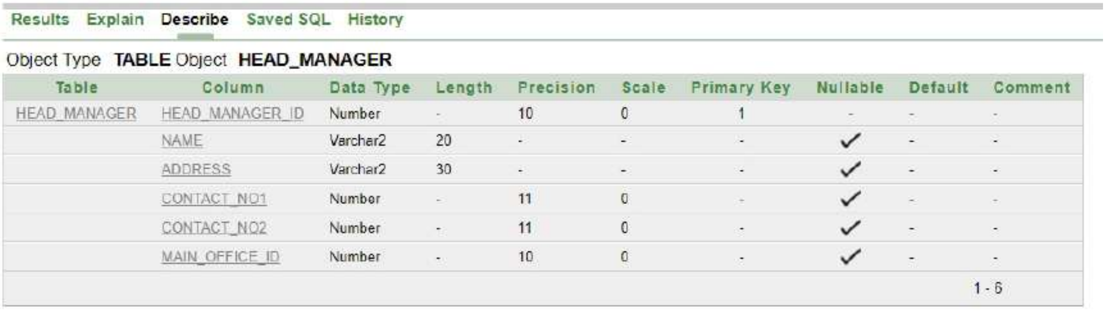


Object Type TABLE Object ASSISTANT_MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ASSISTANT_MANAGER	A_MANAGER_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	MAIN_OFFICE_ID	Number	-	10	0	-	✓	-	-

1 - 6

2. Head_manager table:

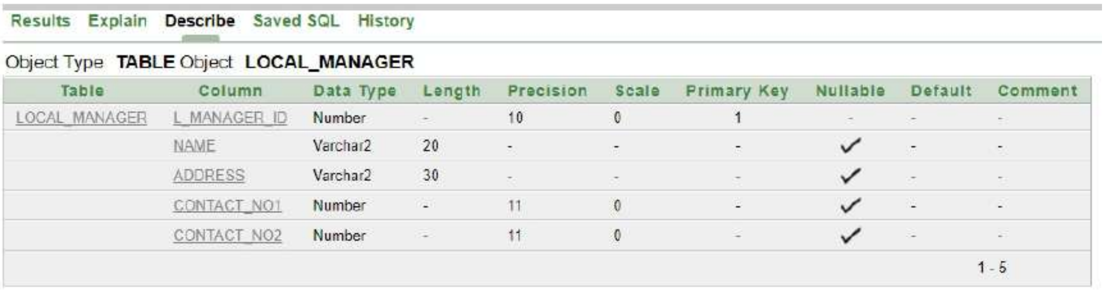


Object Type TABLE Object HEAD_MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HEAD_MANAGER	HEAD_MANAGER_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	MAIN_OFFICE_ID	Number	-	10	0	-	✓	-	-

1 - 6

3. Local_manager table:

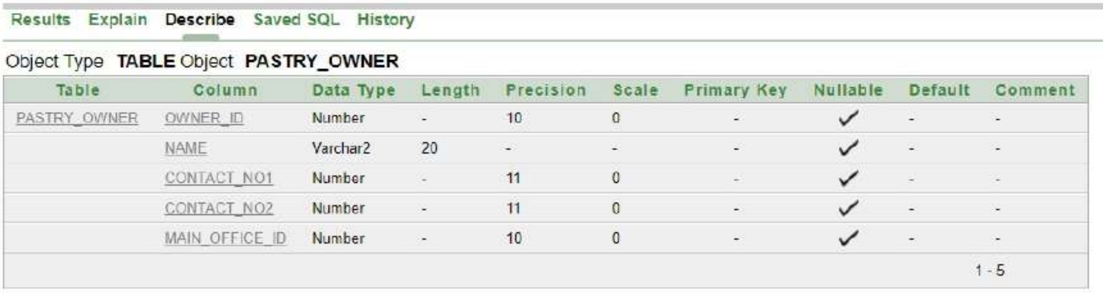


Object Type TABLE Object LOCAL_MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOCAL_MANAGER	L_MANAGER_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-

1 - 5

4. Pastry_owner table:

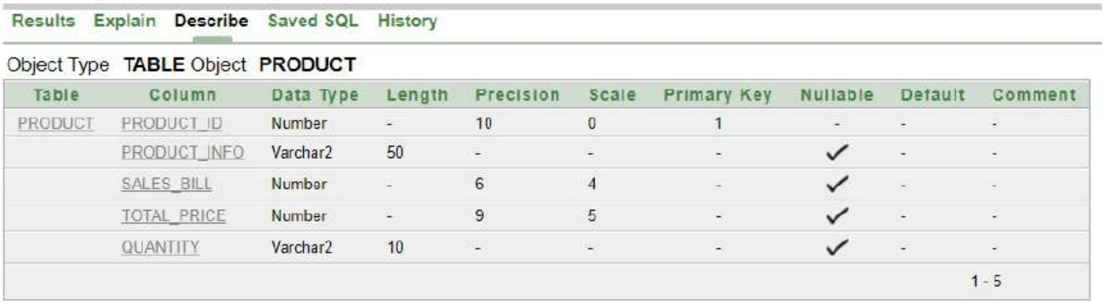


Object Type TABLE Object PASTRY_OWNER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASTRY_OWNER	OWNER_ID	Number	-	10	0	-	✓	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	MAIN_OFFICE_ID	Number	-	10	0	-	✓	-	-

1 - 5

5. Product table:

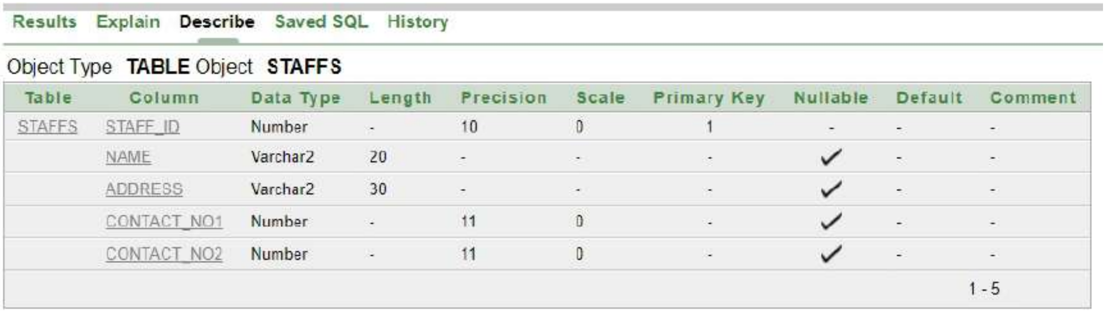


Object Type TABLE Object PRODUCT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODUCT_ID	Number	-	10	0	1	-	-	-
	PRODUCT_INFO	Varchar2	50	-	-	-	✓	-	-
	SALES_BILL	Number	-	6	4	-	✓	-	-
	TOTAL_PRICE	Number	-	9	5	-	✓	-	-
	QUANTITY	Varchar2	10	-	-	-	✓	-	-

1 - 5

6. Staffs table:

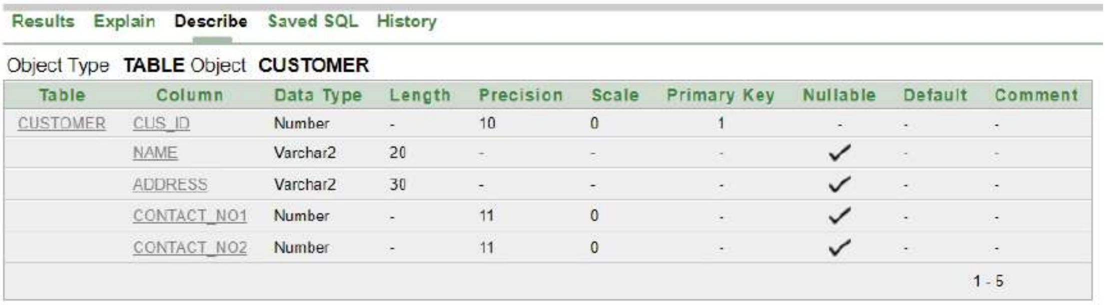


Object Type TABLE Object STAFFS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFFS	STAFF_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-

1 - 5

7. Customer table:

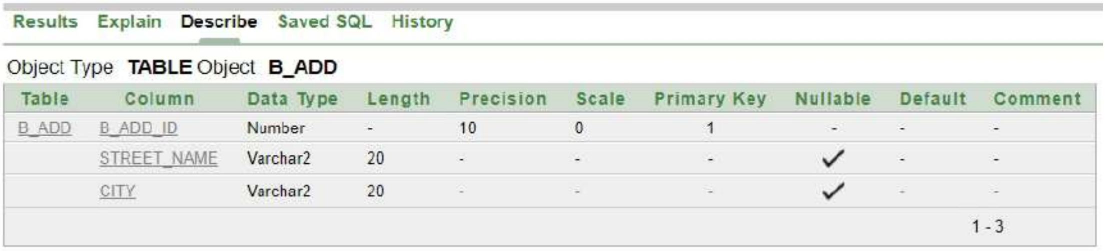


Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUS_ID	Number	-	10	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-

1 - 5

8. B_Add table:

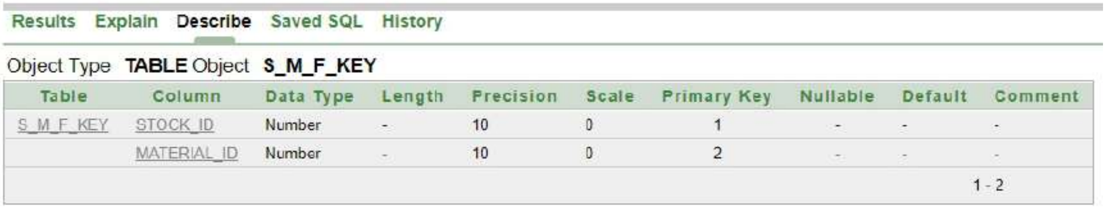


Object Type TABLE Object B_ADD

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_ADD	B_ADD_ID	Number	-	10	0	1	-	-	-
	STREET_NAME	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 3

9. S_M_F_Key table:

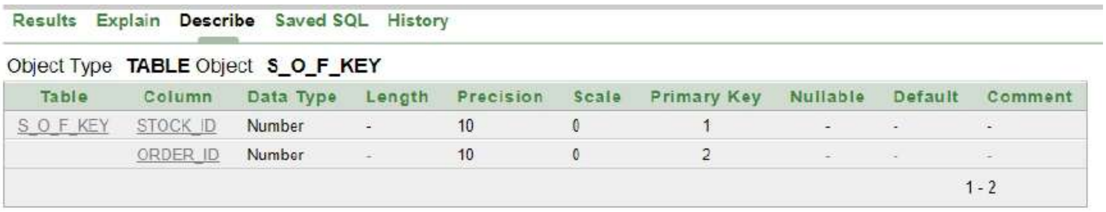


Object Type TABLE Object S_M_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
S_M_F_KEY	STOCK_ID	Number	-	10	0	1	-	-	-
	MATERIAL_ID	Number	-	10	0	2	-	-	-

1 - 2

10. S_O_F_Key table:

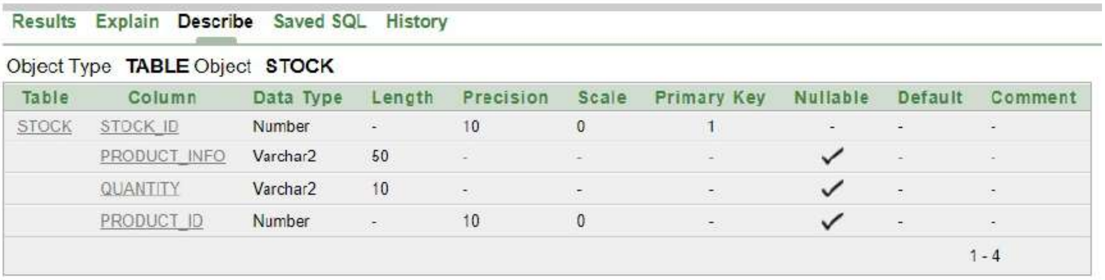


Object Type TABLE Object S_O_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
S_O_F_KEY	STOCK_ID	Number	-	10	0	1	-	-	-
	ORDER_ID	Number	-	10	0	2	-	-	-

1 - 2

11. Stock table:

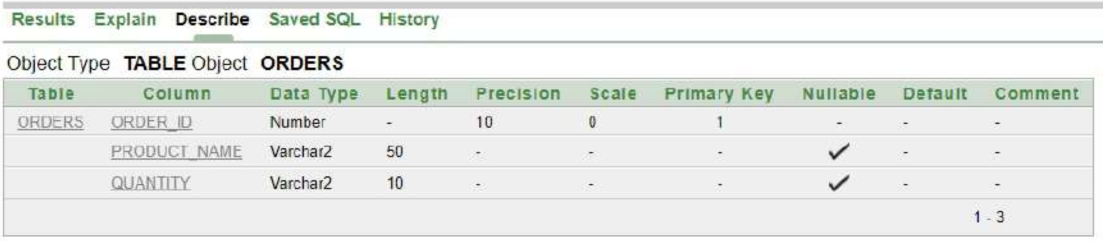


Object Type **TABLE** Object **STOCK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STOCK	STOCK_ID	Number	-	10	0	1	-	-	-
	PRODUCT_INFO	Varchar2	50	-	-	-	✓	-	-
	QUANTITY	Varchar2	10	-	-	-	✓	-	-
	PRODUCT_ID	Number	-	10	0	-	✓	-	-

1 - 4

12. Orders table:

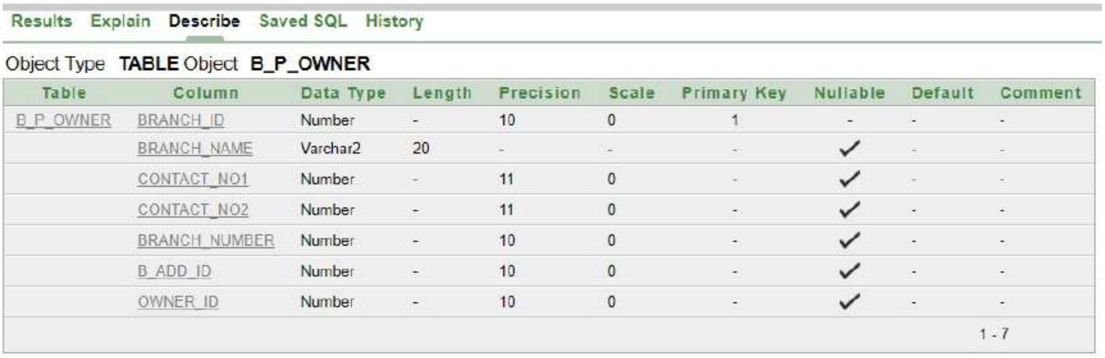


Object Type **TABLE** Object **ORDERS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERS	ORDER_ID	Number	-	10	0	1	-	-	-
	PRODUCT_NAME	Varchar2	50	-	-	-	✓	-	-
	QUANTITY	Varchar2	10	-	-	-	✓	-	-

1 - 3

13. B_P_Owner table:

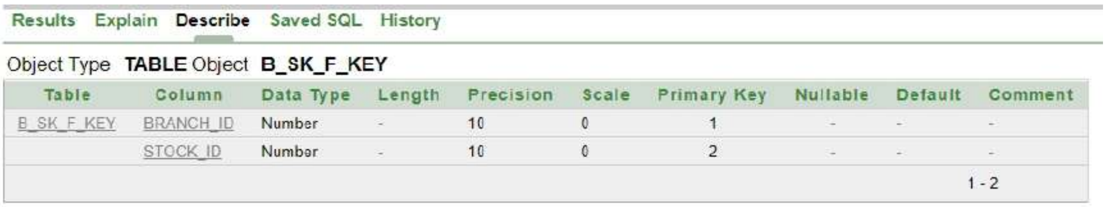


Object Type **TABLE** Object **B_P_OWNER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_P_OWNER	BRANCH_ID	Number	-	10	0	1	-	-	-
	BRANCH_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	BRANCH_NUMBER	Number	-	10	0	-	✓	-	-
	B_ADD_ID	Number	-	10	0	-	✓	-	-
	OWNER_ID	Number	-	10	0	-	✓	-	-

1 - 7

14. B_Sk_F_Key table:

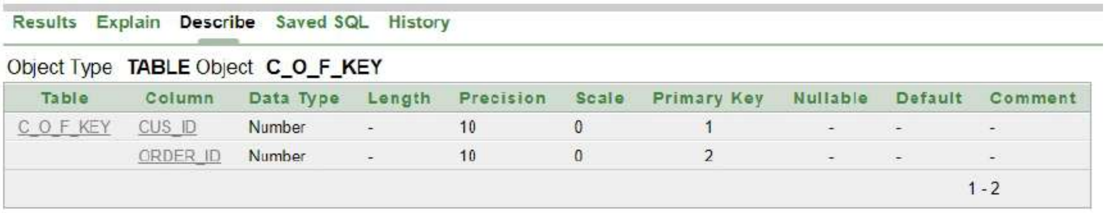


Object Type **TABLE** Object **B_SK_F_KEY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_SK_F_KEY	BRANCH_ID	Number	-	10	0	1	-	-	-
	STOCK_ID	Number	-	10	0	2	-	-	-

1 - 2

15. C_O_F_Key table:

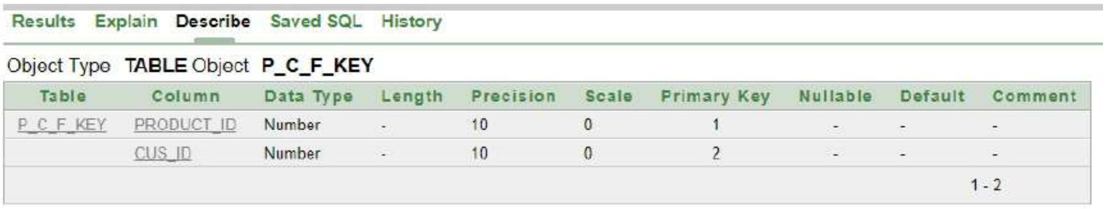


Object Type TABLE Object C_O_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
C_O_F_KEY	CUS_ID	Number	-	10	0	1	-	-	-
	ORDER_ID	Number	-	10	0	2	-	-	-

1 - 2

16. P_C_F_Key table:

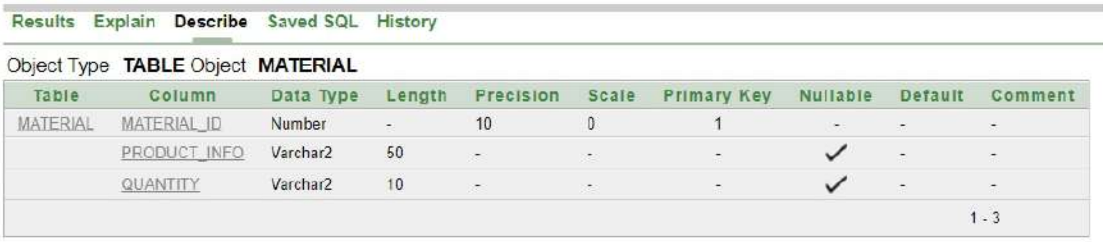


Object Type TABLE Object P_C_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
P_C_F_KEY	PRODUCT_ID	Number	-	10	0	1	-	-	-
	CUS_ID	Number	-	10	0	2	-	-	-

1 - 2

17. Material table:

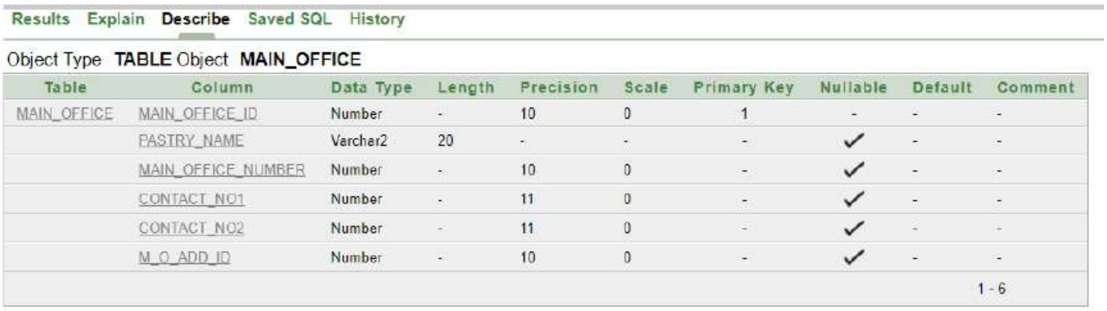


Object Type TABLE Object MATERIAL

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MATERIAL	MATERIAL_ID	Number	-	10	0	1	-	-	-
	PRODUCT_INFO	Varchar2	50	-	-	-	✓	-	-
	QUANTITY	Varchar2	10	-	-	-	✓	-	-

1 - 3

18. Main_office table:

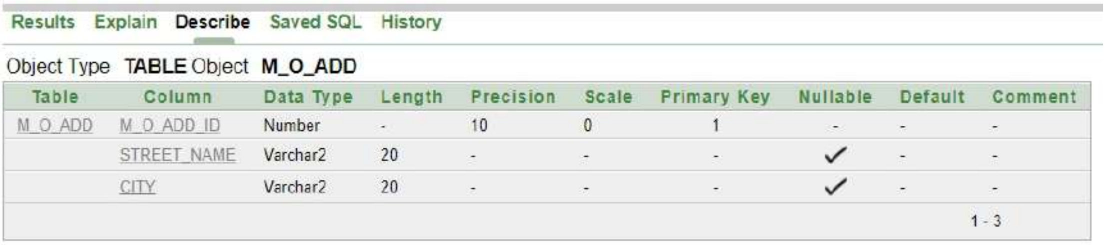


Object Type TABLE Object MAIN_OFFICE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAIN_OFFICE	MAIN_OFFICE_ID	Number	-	10	0	1	-	-	-
	PASTRY_NAME	Varchar2	20	-	-	-	✓	-	-
	MAIN_OFFICE_NUMBER	Number	-	10	0	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	M_O_ADD_ID	Number	-	10	0	-	✓	-	-

1 - 6

19. M_O_Add table:

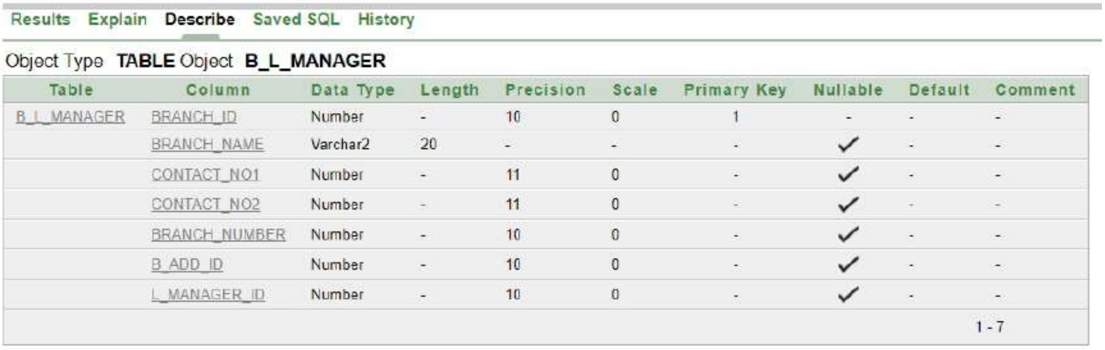


Object Type TABLE Object M_O_ADD

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
M_O_ADD	M_O_ADD_ID	Number	-	10	0	1	-	-	-
	STREET_NAME	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 3

20. B_L_Manager table:

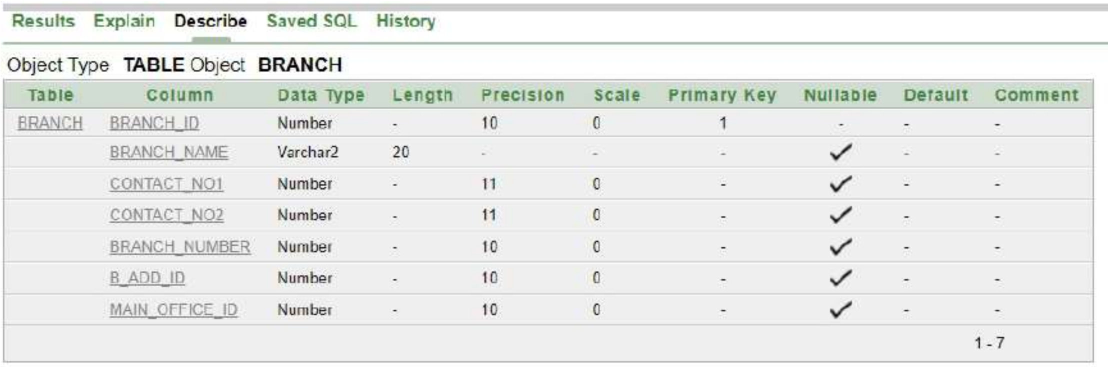


Object Type TABLE Object B_L_MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_L_MANAGER	BRANCH_ID	Number	-	10	0	1	-	-	-
	BRANCH_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	BRANCH_NUMBER	Number	-	10	0	-	✓	-	-
	B_ADD_ID	Number	-	10	0	-	✓	-	-
	L_MANAGER_ID	Number	-	10	0	-	✓	-	-

1 - 7

21. Branch table:

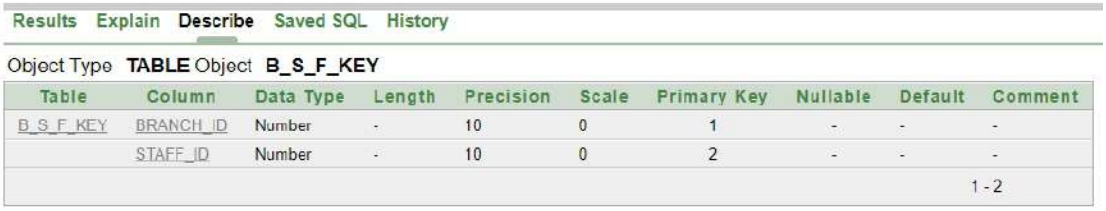


Object Type TABLE Object BRANCH

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRANCH	BRANCH_ID	Number	-	10	0	1	-	-	-
	BRANCH_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_NO1	Number	-	11	0	-	✓	-	-
	CONTACT_NO2	Number	-	11	0	-	✓	-	-
	BRANCH_NUMBER	Number	-	10	0	-	✓	-	-
	B_ADD_ID	Number	-	10	0	-	✓	-	-
	MAIN_OFFICE_ID	Number	-	10	0	-	✓	-	-

1 - 7

22. B_S_F_Key table:



Object Type TABLE Object B_S_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_S_F_KEY	BRANCH_ID	Number	-	10	0	1	-	-	-
	STAFF_ID	Number	-	10	0	2	-	-	-

1 - 2

create sequence:

1.
   ```
   CREATE SEQUENCE Seq
   INCREMENT BY 1
   START WITH 1
   NOCACHE
   NOCYCLE;
   ```

Create users, assign roles and grant privileges :

```
1.  CREATE USER Tanvir
     IDENTIFIED BY tanvir123;
     GRANT UNLIMITED TABLESPACE TO Tanvir;

2. CREATE USER Tanzid
    IDENTIFIED BY tanzid123;
    GRANT UNLIMITED TABLESPACE TO Tanzid;

3. CREATE USER Mahadi
    IDENTIFIED BY mahadi123;
    GRANT UNLIMITED TABLESPACE TO Mahadi;

4. CREATE USER Sakib
    IDENTIFIED BY sakib123;
    GRANT UNLIMITED TABLESPACE TO Sakib;

5. CREATE USER Rakib
    IDENTIFIED BY rakib123;
    GRANT UNLIMITED TABLESPACE TO Rakib;

6. CREATE USER Sadia
    IDENTIFIED BY sadia123;
    GRANT UNLIMITED TABLESPACE TO Sadia;

7. CREATE USER Nishat
    IDENTIFIED BY nishat123;
    GRANT UNLIMITED TABLESPACE TO Nishat;

8. CREATE USER Jamee
    IDENTIFIED BY jamee123;
    GRANT UNLIMITED TABLESPACE TO Jamee;

9. CREATE USER Mamun
    IDENTIFIED BY mamun123;
    GRANT UNLIMITED TABLESPACE TO Mamun;
```

```
10. CREATE USER Rafa
       IDENTIFIED BY rafa123;
      GRANT UNLIMITED TABLESPACE TO Rafa;

11. CREATE USER Rafsan
       IDENTIFIED BY rafsan123;
      GRANT UNLIMITED TABLESPACE TO Rafsan;

12. CREATE USER Nadim
       IDENTIFIED BY nadim123;
       GRANT UNLIMITED TABLESPACE TO Nadim;
```

```
1. CREATE ROLE Pastry_owner;
   GRANT create table, create view, create sequence, create user, create
role, create session, create procedure TO Pastry_owner;

2. CREATE ROLE Head_manager;
   GRANT create table, create view, create sequence, create session,
create procedure TO Head_manager;

3. CREATE ROLE Assistant_managers;
   GRANT create table, create view, create sequence, create session TO
Assistant_managers;

4. CREATE ROLE Local_managers;
   GRANT create view TO Local_managers;

1. GRANT Pastry_owner TO Tanvir;

2. GRANT Head_manager TO Tanzid;

3. GRANT Assistant_managers TO Mahadi, Sakib, Rakib, Sadia, Nishat;

4. GRANT Local_managers TO Jamee, Mamun, Rafa, Rafsan, Nadim;
```

Data Insertion :

```
1. INSERT INTO M_O_Add VALUES(seq.NEXTVAL, '9/D,SM Road,Mirpur-1', 'Dhaka');
```

```
2. INSERT INTO B_Add VALUES(seq.NEXTVAL, '1/L,Jia,Mirpur-1', 'Dhaka');
INSERT INTO B_Add VALUES(seq.NEXTVAL, '1/L,Jia,Mirpur-1', 'Dhaka');
INSERT INTO B_Add VALUES(seq.NEXTVAL, 'H#1,R#2,S#5,Uttara', 'Dhaka');
INSERT INTO B_Add VALUES(seq.NEXTVAL, 'Kazi road,Bamapara', 'Rangpur');
INSERT INTO B_Add VALUES(seq.NEXTVAL, 'City road,Sornogachi', 'Chittagong')
```

```
3. INSERT INTO Local_manager VALUES (seq.NEXTVAL, 'Jamee', 'Mirpur,Dhaka', '011299','229911');
INSERT INTO Local_manager VALUES (seq.NEXTVAL, 'Mamun', 'Uttara,Dhaka', '01133','2212211');
INSERT INTO Local_manager VALUES (seq.NEXTVAL, 'Rafa', 'Kazipara,Rangpur', '013199','221111');
INSERT INTO Local_manager VALUES (seq.NEXTVAL, 'Rafsan', 'Ullapara,Sylhet', '011399','29111');
INSERT INTO Local_manager VALUES (seq.NEXTVAL, 'Nadim', 'Sornogachi,Chittagong', '044199','02111');
```

```
4. INSERT INTO Orders VALUES (seq.NEXTVAL, 'Rosmonjoli', '800 gm');
INSERT INTO Orders VALUES (seq.NEXTVAL, 'Cup cake','500 gm');
INSERT INTO Orders VALUES (seq.NEXTVAL, 'Cup cream','900 gm');
INSERT INTO Orders VALUES (seq.NEXTVAL, 'Misti doi', '1 kg');
INSERT INTO Orders VALUES (seq.NEXTVAL, 'Choco cake', '1.5 kg');
```

```
5. INSERT INTO  Material VALUES (seq.NEXTVAL, 'Pastry maker with extra sugar', '10kg');
INSERT INTO Material VALUES (seq.NEXTVAL, 'Vanilla food flavor grade-6','10kg');
INSERT INTO Material VALUES (seq.NEXTVAL, 'Chocolate mix','6kg');
INSERT INTO Material VALUES (seq.NEXTVAL, 'Raw doi-less fat', '7kg');
INSERT INTO Material VALUES (seq.NEXTVAL, 'Wilton Gel Food Color Set, Primary', '6 kg');
```

```
6. INSERT INTO Staffs VALUES (seq.NEXTVAL, 'Nafin', 'Mirpur,Dhaka',
'011399','22991');
INSERT INTO Staffs VALUES (seq.NEXTVAL, 'Masud', 'Uttara,Dhaka',
'01033','2212111');
INSERT INTO Staffs VALUES (seq.NEXTVAL, 'Rafsan', 'Dhap,Rangpur',
'022199','201111');
INSERT INTO Staffs VALUES (seq.NEXTVAL, 'Masrafi', 'Ullapara,Sylhet',
'11299','2920201');
INSERT INTO Staffs VALUES (seq.NEXTVAL, 'Salam',
'Majhipara,Chittagong', '024199','02221');

7. INSERT INTO Main_office VALUES (seq.NEXTVAL, 'Pastry
Village®','011','781911','122211','6');

8. INSERT INTO Pastry_owner VALUES (seq.NEXTVAL, 'Tanvir',
'011399','22991','33');

9. INSERT INTO Assistant_manager VALUES (seq.NEXTVAL, 'Mahadi',
'Mirpur,Dhaka', '011299','229911','33');
INSERT INTO Assistant_manager VALUES (seq.NEXTVAL, 'Sakib',
'Uttara,Dhaka', '011219','229911','33');
INSERT INTO Assistant_manager VALUES (seq.NEXTVAL, 'Rakib',
'Sonargoan,Dhaka', '011299','229911','33');
INSERT INTO Assistant_manager VALUES (seq.NEXTVAL, 'Sadia',
'Mohammadpur,Dhaka', '011299','229911','33');
INSERT INTO Assistant_manager VALUES (seq.NEXTVAL, 'Nishat',
'Dhap,Rangpur', '011299','229911','33');

10. INSERT INTO Head_manager VALUES (seq.NEXTVAL, 'Tanzid',
'Rampura,Dhaka', '01999','023911','33');

11. INSERT INTO Product VALUES (seq.NEXTVAL,'Rosmonjoli-Small
size','10.56','10.56','800 gm');
INSERT INTO Product VALUES (seq.NEXTVAL,'Rosmonjoli-Regular
size','80.56','85.56','800 gm');
INSERT INTO Product VALUES (seq.NEXTVAL,'Cup
cake','95.56','99.56','500 gm');
INSERT INTO Product VALUES (seq.NEXTVAL,'Cup
cream','70.56','80.56','900 gm');
```

```
INSERT INTO Product VALUES (seq.NEXTVAL,'Misti doi','90.56','98.56','1 kg');

12. INSERT INTO Stock VALUES (seq.NEXTVAL,'Rosmonjoli-Small size','800 gm','50');
INSERT INTO Stock VALUES (seq.NEXTVAL,'Rosmonjoli-Regular size','800 gm','51');
INSERT INTO Stock VALUES (seq.NEXTVAL,'Cup cake','500 gm','52');
INSERT INTO Stock VALUES (seq.NEXTVAL,'Cup cream','900 gm','53');
INSERT INTO Stock VALUES (seq.NEXTVAL,'Misti doi','1 kg','54');

13. INSERT INTO Customer VALUES (seq.NEXTVAL, 'Nadia', 'Dhanmondi,Dhaka', '991399','2112991');
INSERT INTO Customer VALUES (seq.NEXTVAL, 'Borsha', 'Uttara,Dhaka', '0103213','912111');
INSERT INTO Customer VALUES (seq.NEXTVAL, 'Rafi', 'CO bazar,Rangpur', '9812199','10982111');
INSERT INTO Customer VALUES (seq.NEXTVAL, 'Salim', 'Puran dhaka, Dhaka', '8819099','28820201');
INSERT INTO Customer VALUES (seq.NEXTVAL, 'Samiya', 'Rajhandrapur,Chittagong', '02422199','02198221');

14. INSERT INTO Branch VALUES (seq.NEXTVAL, 'Pastry king®', '01987777', '10911111', '102','7', '33');
INSERT INTO Branch VALUES (seq.NEXTVAL, 'Pastry king®', '019811777', '1901111', '103','8', '33');
INSERT INTO Branch VALUES (seq.NEXTVAL, 'Pastry boy®', '01782277', '109865111', '106','9', '33');
INSERT INTO Branch VALUES (seq.NEXTVAL, 'Pastry village-2®', '01945777', '10771111', '100','10', '33');
INSERT INTO Branch VALUES (seq.NEXTVAL, 'Pastry house®', '01909177', '10551111', '111','11', '33');

15. INSERT INTO B_L_Manager VALUES (65, 'Pastry king®', '01987777', '10911111', '102','7', '12');
INSERT INTO B_L_Manager VALUES (66, 'Pastry king®', '019811777', '1901111', '103','8', '13');
INSERT INTO B_L_Manager VALUES (67, 'Pastry boy®', '01782277', '109865111', '106','9', '14');
INSERT INTO B_L_Manager VALUES (68, 'Pastry village-2®', '01945777',
'10771111', '100','10', '15');
INSERT INTO B_L_Manager VALUES (69, 'Pastry house®', '01909177',
'10551111', '111','11', '16');
```

```
16. INSERT INTO B_P_Owner VALUES (65, 'Pastry king®', '01987777',
'10911111', '102','7', '35');
INSERT INTO B_P_Owner VALUES (66, 'Pastry king®', '019811777',
'1901111', '103','8', '35');
INSERT INTO B_P_Owner VALUES (67, 'Pastry boy®', '01782277',
'109865111', '106','9', '35');
INSERT INTO B_P_Owner VALUES (68, 'Pastry village-2®', '01945777',
'10771111', '100','10', '35');
INSERT INTO B_P_Owner VALUES (69, 'Pastry house®', '01909177',
'10551111', '111','11', '35');
```

```
17. INSERT INTO S_M_F_Key VALUES (55,22);
INSERT INTO S_M_F_Key VALUES (56,23);
INSERT INTO S_M_F_Key VALUES (57,24);
INSERT INTO S_M_F_Key VALUES (58,25);
INSERT INTO S_M_F_Key VALUES (59,26);
```

```
18. INSERT INTO B_Sk_F_Key VALUES (65,55);
INSERT INTO B_Sk_F_Key VALUES (66,56);
INSERT INTO B_Sk_F_Key VALUES (67,57);
INSERT INTO B_Sk_F_Key VALUES (68,58);
INSERT INTO B_Sk_F_Key VALUES (69,59);
```

```
19. INSERT INTO C_O_F_Key VALUES (60,17);
INSERT INTO C_O_F_Key VALUES (61,18);
INSERT INTO C_O_F_Key VALUES (62,19);
INSERT INTO C_O_F_Key VALUES (63,20);
INSERT INTO C_O_F_Key VALUES (64,21);
```

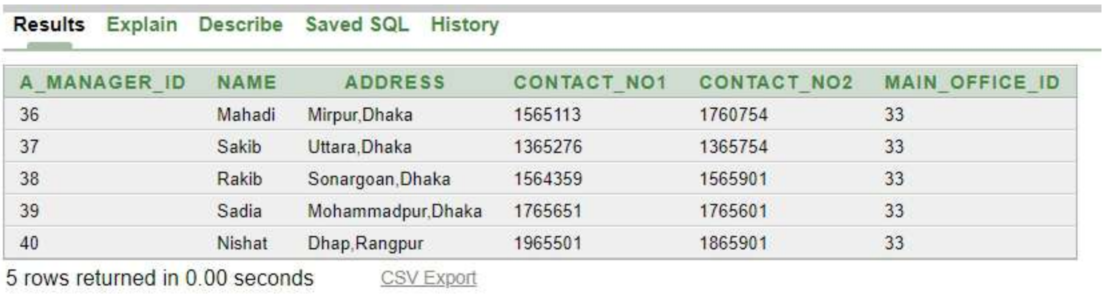
```
20. INSERT INTO B_S_F_Key VALUES (65,27);
INSERT INTO B_S_F_Key VALUES (66,28);
INSERT INTO B_S_F_Key VALUES (67,29);
INSERT INTO B_S_F_Key VALUES (68,30);
INSERT INTO B_S_F_Key VALUES (69,31);

21. INSERT INTO P_C_F_Key VALUES (50,60);
INSERT INTO P_C_F_Key VALUES (51,61);
INSERT INTO P_C_F_Key VALUES (52,62);
INSERT INTO P_C_F_Key VALUES (53,63);
INSERT INTO P_C_F_Key VALUES (54,64);

22. INSERT INTO S_O_F_Key VALUES (55,17);
INSERT INTO S_O_F_Key VALUES (56,18);
INSERT INTO S_O_F_Key VALUES (57,19);
INSERT INTO S_O_F_Key VALUES (58,20);
INSERT INTO S_O_F_Key VALUES (59,21);
```

screenshots of the tables after inserting data :

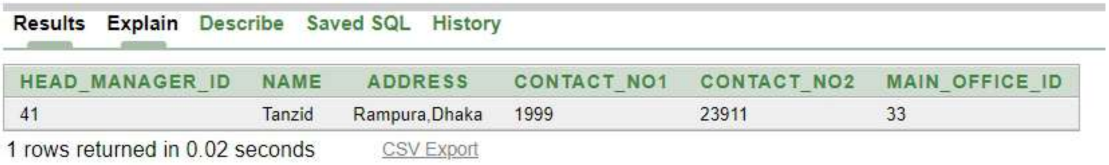
1. Assistant_manager table data:



A_MANAGER_ID	NAME	ADDRESS	CONTACT_NO1	CONTACT_NO2	MAIN_OFFICE_ID
36	Mahadi	Mirpur,Dhaka	1565113	1760754	33
37	Sakib	Uttara,Dhaka	1365276	1365754	33
38	Rakib	Sonargoan,Dhaka	1564359	1565901	33
39	Sadia	Mohammadpur,Dhaka	1765651	1765601	33
40	Nishat	Dhap,Rangpur	1965501	1865901	33

5 rows returned in 0.00 seconds

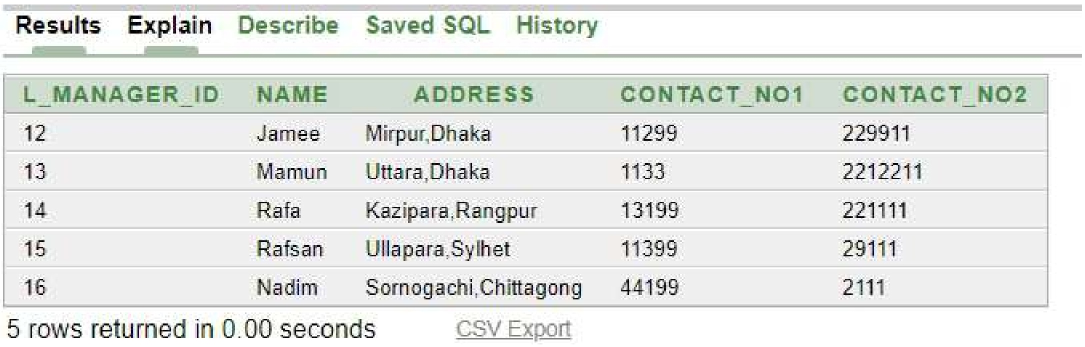
2. Head_manager table data:



HEAD_MANAGER_ID	NAME	ADDRESS	CONTACT_NO1	CONTACT_NO2	MAIN_OFFICE_ID
41	Tanzid	Rampura,Dhaka	1999	23911	33

1 rows returned in 0.02 seconds

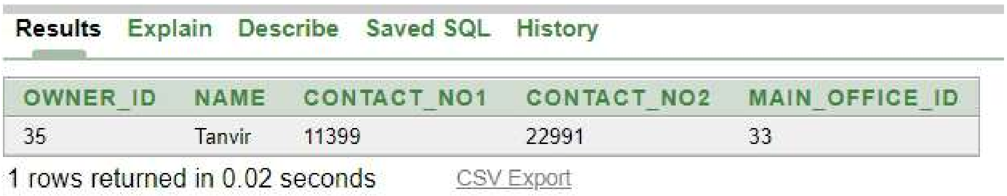
3. Local_manager table data:



L_MANAGER_ID	NAME	ADDRESS	CONTACT_NO1	CONTACT_NO2
12	Jamee	Mirpur,Dhaka	11299	229911
13	Mamun	Uttara,Dhaka	1133	2212211
14	Rafa	Kazipara,Rangpur	13199	221111
15	Rafsan	Ullapara,Sylhet	11399	29111
16	Nadim	Sornogachi,Chittagong	44199	2111

5 rows returned in 0.00 seconds CSV Export

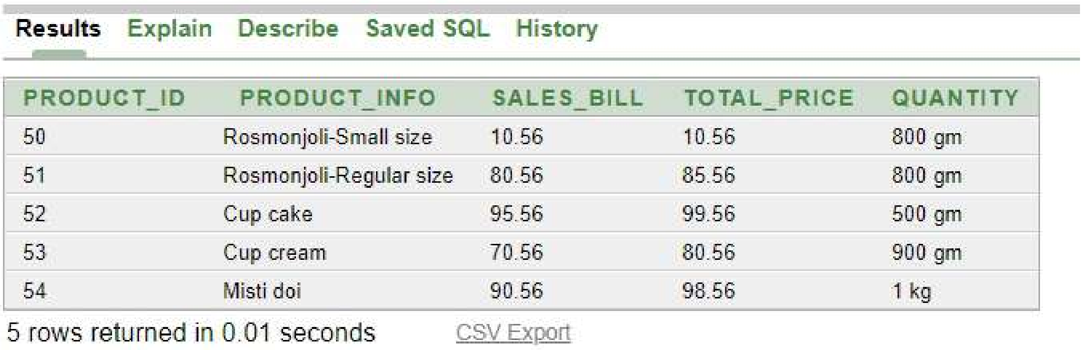
4. Pastry_owner table data:



OWNER_ID	NAME	CONTACT_NO1	CONTACT_NO2	MAIN_OFFICE_ID
35	Tanvir	11399	22991	33

1 rows returned in 0.02 seconds CSV Export

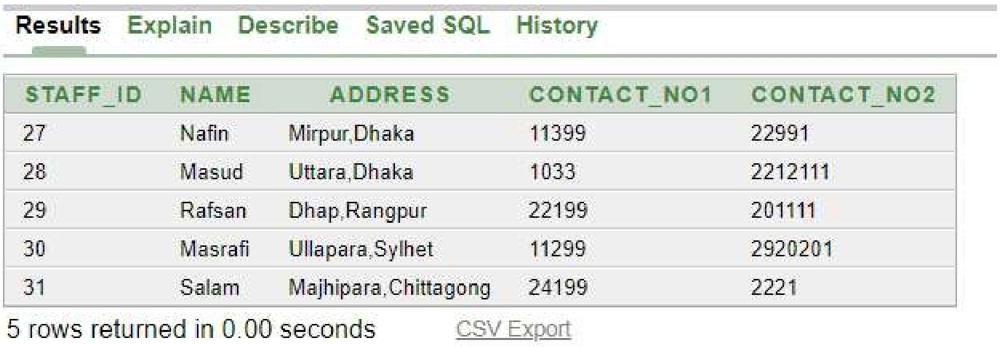
5. Product table data:



PRODUCT_ID	PRODUCT_INFO	SALES_BILL	TOTAL_PRICE	QUANTITY
50	Rosmonjoli-Small size	10.56	10.56	800 gm
51	Rosmonjoli-Regular size	80.56	85.56	800 gm
52	Cup cake	95.56	99.56	500 gm
53	Cup cream	70.56	80.56	900 gm
54	Misti doi	90.56	98.56	1 kg

5 rows returned in 0.01 seconds CSV Export

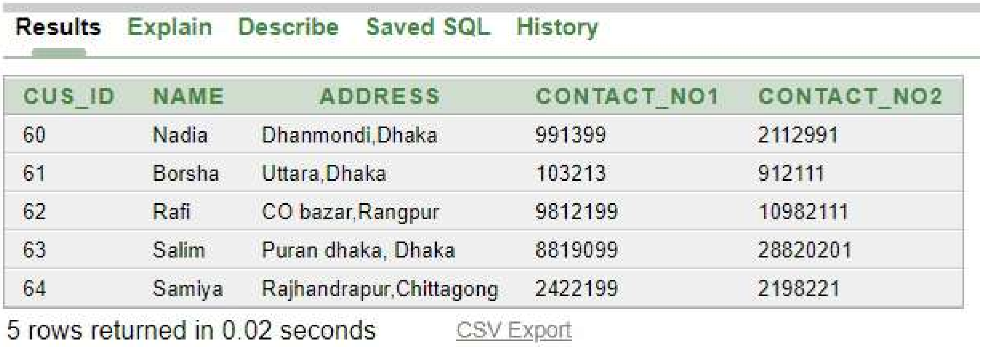
6. Staffs table:



STAFF_ID	NAME	ADDRESS	CONTACT_NO1	CONTACT_NO2
27	Nafin	Mirpur,Dhaka	11399	22991
28	Masud	Uttara,Dhaka	1033	2212111
29	Rafsan	Dhap,Rangpur	22199	201111
30	Masrafi	Ullapara,Sylhet	11299	2920201
31	Salam	Majhipara,Chittagong	24199	2221

5 rows returned in 0.00 seconds CSV Export

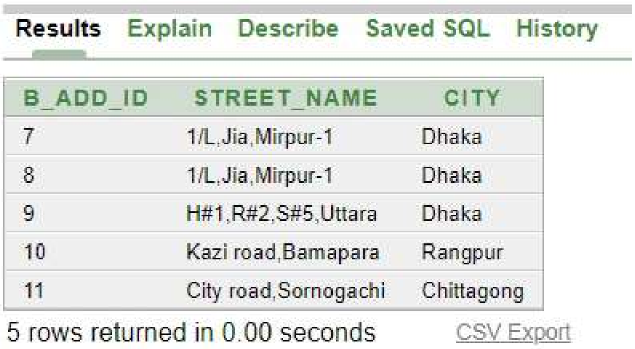
7. Customer table:



CUS_ID	NAME	ADDRESS	CONTACT_NO1	CONTACT_NO2
60	Nadia	Dhanmondi,Dhaka	991399	2112991
61	Borsha	Uttara,Dhaka	103213	912111
62	Rafi	CO bazar,Rangpur	9812199	10982111
63	Salim	Puran dhaka, Dhaka	8819099	28820201
64	Samiya	Rajhandrapur,Chittagong	2422199	2198221

5 rows returned in 0.02 seconds

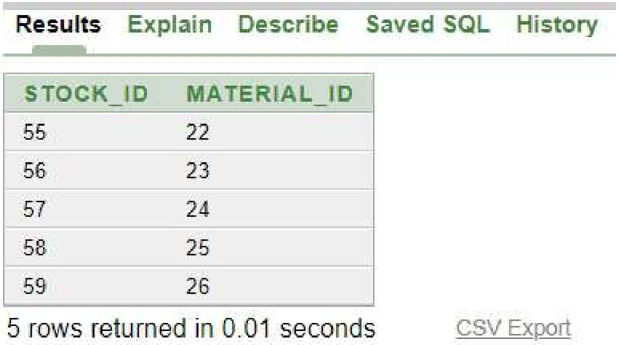
8. B_Add table data:



B_ADD_ID	STREET_NAME	CITY
7	1/L,Jia,Mirpur-1	Dhaka
8	1/L,Jia,Mirpur-1	Dhaka
9	H#1,R#2,S#5,Uttara	Dhaka
10	Kazi road,Bamapara	Rangpur
11	City road,Sornogachi	Chittagong

5 rows returned in 0.00 seconds

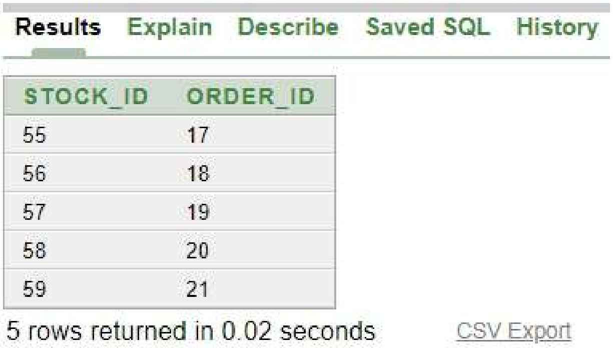
9. S_M_F_Key table data:



STOCK_ID	MATERIAL_ID
55	22
56	23
57	24
58	25
59	26

5 rows returned in 0.01 seconds

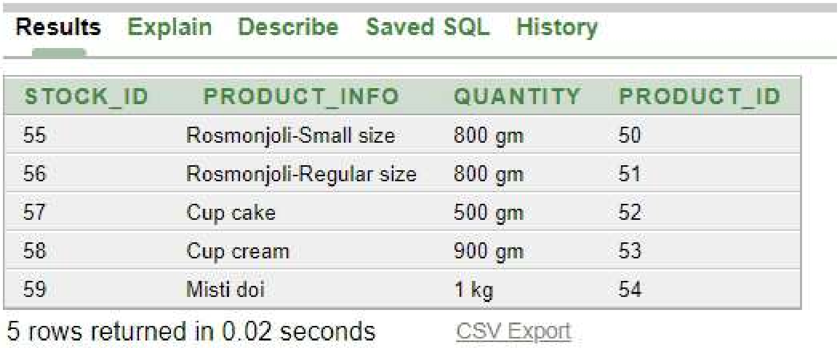
10. S_O_F_Key table data:



STOCK_ID	ORDER_ID
55	17
56	18
57	19
58	20
59	21

5 rows returned in 0.02 seconds

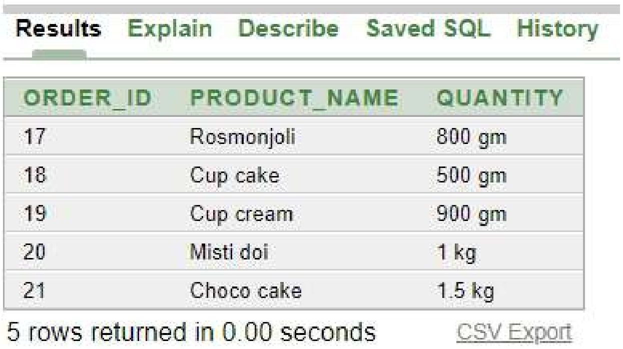
11. Stock table data:



STOCK_ID	PRODUCT_INFO	QUANTITY	PRODUCT_ID
55	Rosmonjoli-Small size	800 gm	50
56	Rosmonjoli-Regular size	800 gm	51
57	Cup cake	500 gm	52
58	Cup cream	900 gm	53
59	Misti doi	1 kg	54

5 rows returned in 0.02 seconds

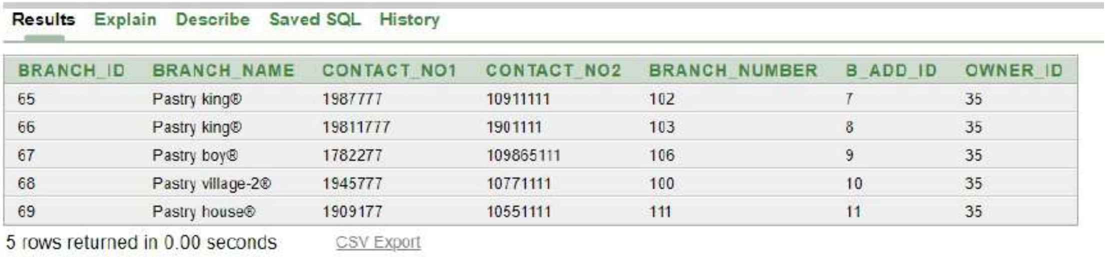
12. Orders table data :



ORDER_ID	PRODUCT_NAME	QUANTITY
17	Rosmonjoli	800 gm
18	Cup cake	500 gm
19	Cup cream	900 gm
20	Misti doi	1 kg
21	Choco cake	1.5 kg

5 rows returned in 0.00 seconds

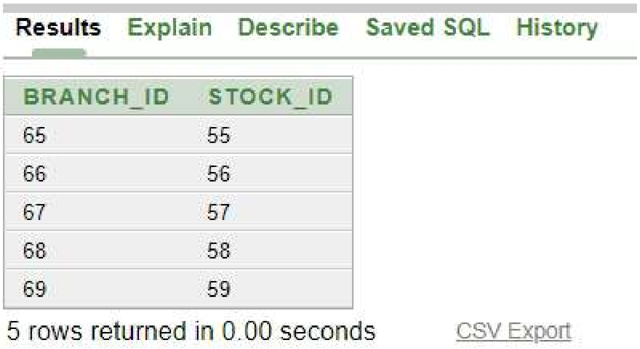
13. B_P_Owner table data:



BRANCH_ID	BRANCH_NAME	CONTACT_NO1	CONTACT_NO2	BRANCH_NUMBER	B_ADD_ID	OWNER_ID
65	Pastry king®	1987777	10911111	102	7	35
66	Pastry king®	19811777	1901111	103	8	35
67	Pastry boy®	1782277	109865111	106	9	35
68	Pastry village-2®	1945777	10771111	100	10	35
69	Pastry house®	1909177	10551111	111	11	35

5 rows returned in 0.00 seconds

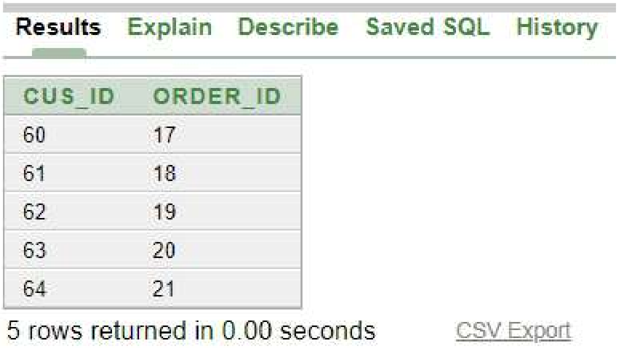
14. B_Sk_F_Key table:



BRANCH_ID	STOCK_ID
65	55
66	56
67	57
68	58
69	59

5 rows returned in 0.00 seconds

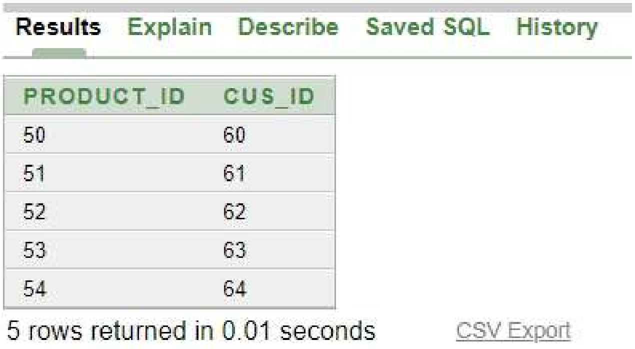
15. C_O_F_Key table



CUS_ID	ORDER_ID
60	17
61	18
62	19
63	20
64	21

5 rows returned in 0.00 seconds

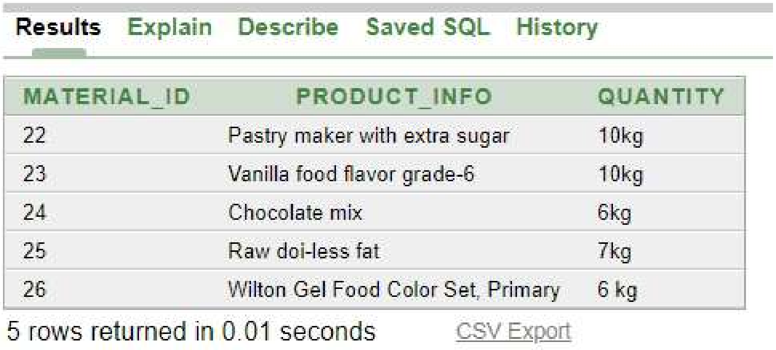
16. P_C_F_Key table data:



PRODUCT_ID	CUS_ID
50	60
51	61
52	62
53	63
54	64

5 rows returned in 0.01 seconds

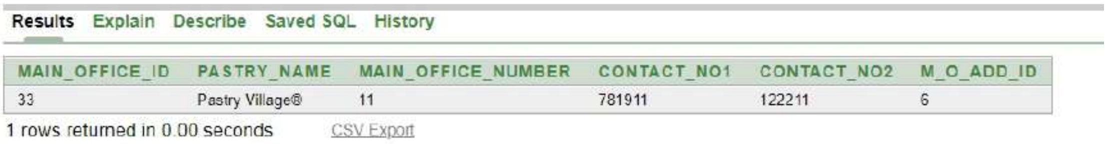
17. Material table data:



MATERIAL_ID	PRODUCT_INFO	QUANTITY
22	Pastry maker with extra sugar	10kg
23	Vanilla food flavor grade-6	10kg
24	Chocolate mix	6kg
25	Raw doi-less fat	7kg
26	Wilton Gel Food Color Set, Primary	6 kg

5 rows returned in 0.01 seconds

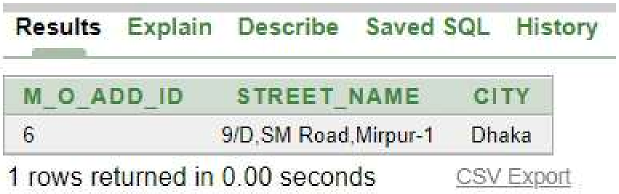
18. Main_office table data:



MAIN_OFFICE_ID	PASTRY_NAME	MAIN_OFFICE_NUMBER	CONTACT_NO1	CONTACT_NO2	M_O_ADD_ID
33	Pastry Village®	11	781911	122211	6

1 rows returned in 0.00 seconds

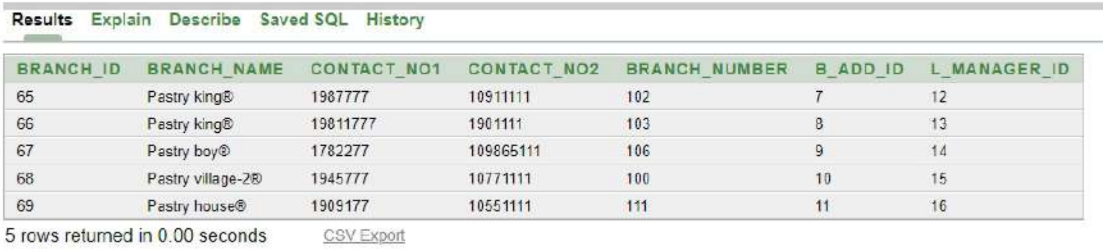
19. M_O_Add table data:



M_O_ADD_ID	STREET_NAME	CITY
6	9/D,SM Road,Mirpur-1	Dhaka

1 rows returned in 0.00 seconds

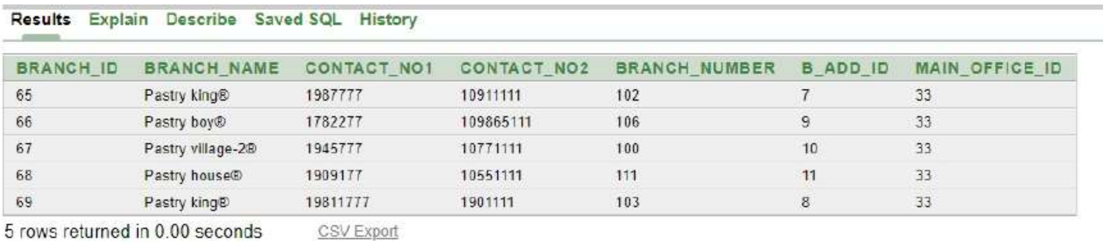
20. B_L_Manager table:



BRANCH_ID	BRANCH_NAME	CONTACT_NO1	CONTACT_NO2	BRANCH_NUMBER	B_ADD_ID	L_MANAGER_ID
65	Pastry king®	1987777	10911111	102	7	12
66	Pastry king®	19811777	1901111	103	8	13
67	Pastry boy®	1782277	109865111	106	9	14
68	Pastry village-2®	1945777	10771111	100	10	15
69	Pastry house®	1909177	10551111	111	11	16

5 rows returned in 0.00 seconds

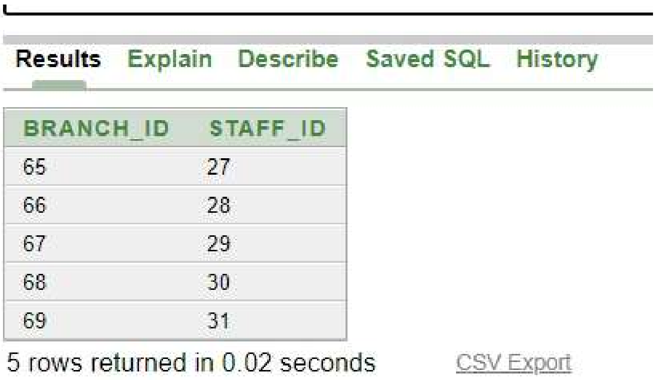
21. Branch table:



BRANCH_ID	BRANCH_NAME	CONTACT_NO1	CONTACT_NO2	BRANCH_NUMBER	B_ADD_ID	MAIN_OFFICE_ID
65	Pastry king®	1987777	10911111	102	7	33
66	Pastry boy®	1782277	109865111	106	9	33
67	Pastry village-2®	1945777	10771111	100	10	33
68	Pastry house®	1909177	10551111	111	11	33
69	Pastry king®	19811777	1901111	103	8	33

5 rows returned in 0.00 seconds CSV Export

22. B_S_F_Key table data:



BRANCH_ID	STAFF_ID
65	27
66	28
67	29
68	30
69	31

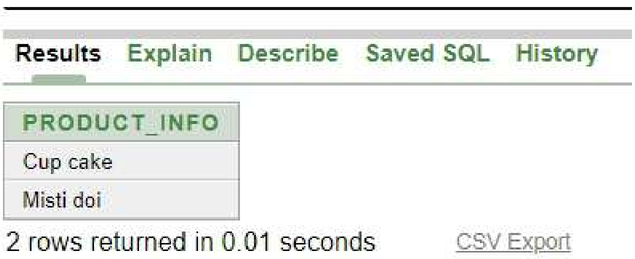
5 rows returned in 0.02 seconds CSV Export

Query Writing :

Subquery :

1. Which products total price are more than the the total price of Rosmonjoli-Regular size?

Ans:
```
SELECT product_info
FROM product
WHERE total_price>(SELECT total_price FROM product WHERE
product_info = 'Rosmonjoli-Regular size');
```



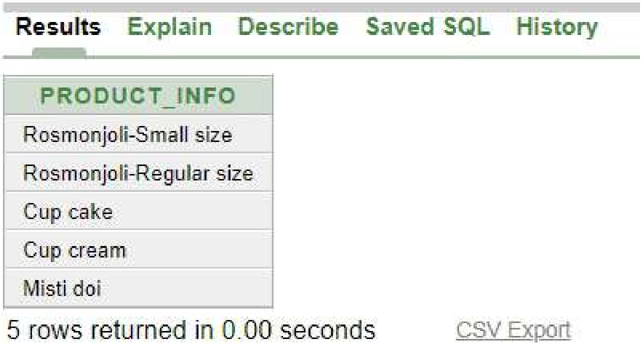
PRODUCT_INFO
Cup cake
Misti doi

2 rows returned in 0.01 seconds CSV Export

2. Which products sales bill are less than the total price of a product where product has a word called "doi" along with other words?

Ans:

```
SELECT Product_info
FROM product
WHERE sales_bill<(SELECT total_price FROM product WHERE
product_info LIKE '%doi%');
```



PRODUCT_INFO
Rosmonjoli-Small size
Rosmonjoli-Regular size
Cup cake
Cup cream
Misti doi

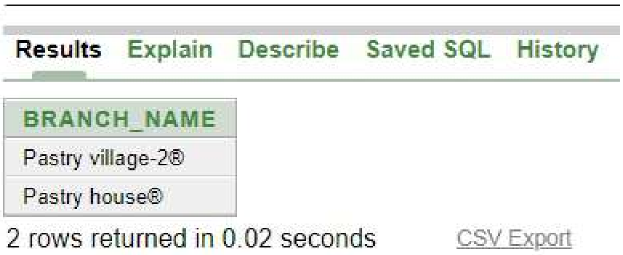
5 rows returned in 0.00 seconds

Joining:

1. Display the name of all the branches which are outside of Dhaka.

Ans:

```
SELECT bn.branch_name
FROM branch bn, b_add ba
WHERE bn.b_add_id=ba.b_add_id AND CITY NOT IN ('Dhaka');
```



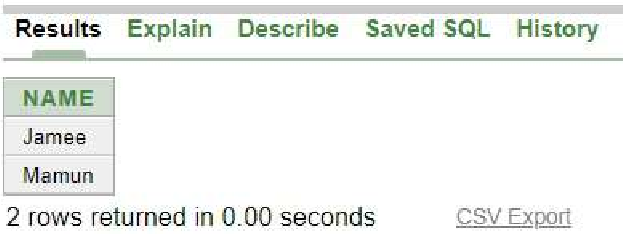
BRANCH_NAME
Pastry village-2®
Pastry house®

2 rows returned in 0.02 seconds

2.Display the name of the local managers of Pastry king®.

Ans:

```
SELECT l.name
FROM local_manager l, b_l_manager b
WHERE l.l_manager_id=b.l_manager_id AND
branch_name='Pastry king®';
```



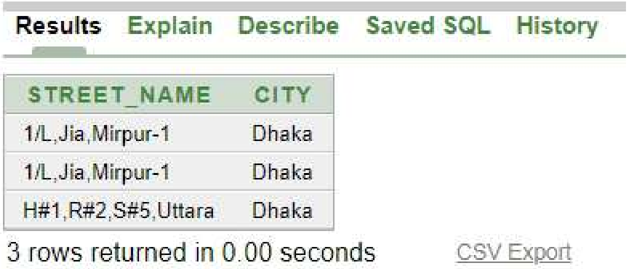
NAME
Jamee
Mamun

2 rows returned in 0.00 seconds

View :

1. Create a read only view called B_Add_View based on the Street_name and City from the B_Add table where city is Dhaka.

Ans:
```
CREATE VIEW B_Add_View
     AS SELECT street_name, city
     FROM B_Add
     WHERE city='Dhaka'
     WITH READ ONLY;
```

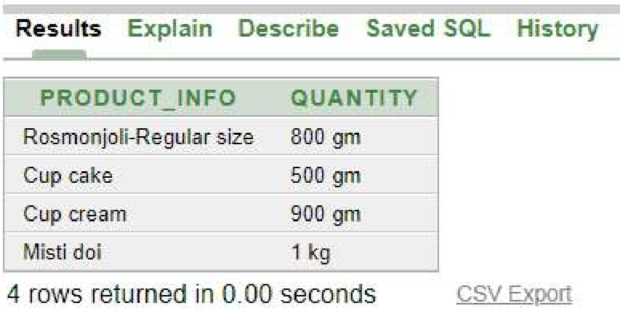


STREET_NAME	CITY
1/L,Jia,Mirpur-1	Dhaka
1/L,Jia,Mirpur-1	Dhaka
H#1,R#2,S#5,Uttara	Dhaka

3 rows returned in 0.00 seconds

2. Create a read only view called P_View based on the Product_info and Quantity from Product table where total price is more than 80.00 taka.

Ans:
```
CREATE VIEW P_View
     AS SELECT product_info, quantity
     FROM product
     WHERE total_price>80.00
     WITH READ ONLY;
```



PRODUCT_INFO	QUANTITY
Rosmonjoli-Regular size	800 gm
Cup cake	500 gm
Cup cream	900 gm
Misti doi	1 kg

4 rows returned in 0.00 seconds

Relational Algebra :

1. Find the name of the branch where branch mumber is 103.

Ans: $\pi_{branch\_name}\ (\sigma_{branch\_number="103"}(Branch))$

2. Find the sales bill of the product where product_id is 52.

Ans: $\pi_{sales\_bill}\ (\sigma_{product\_id="52"}(Product))$

3. Find the order id of the product where product name is Cup cake.

Ans: . $\pi_{order\_id}\ (\sigma_{product\_name="Cup\ cake"}(Orders))$

4. Find the address of the local manager where local manager id is 14.

Ans: $\pi_{address}\ (\sigma_{l\_manager\_id="14"}(Local\_manager))$

5. Find the staff id of the staff where name of the staff is Masud.

Ans: $\pi_{staff\_id}\ (\sigma_{name="Masud"}(Staffs))$

Conclusion:

As we have worked on pastry management system it will provide you with your shop data. Daily sales, staff, accounting, customer, delivery reports and more can be run on your system. it also keeps track on its employees. Our goal was to make a very simple yet effective database management system for this "pastry shop management system". And we have tried our best in a short time .If we get a chance to make it better it can even possible . We can add some more features on this project afterward. It can be modify able and can be better.