AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Project Name: Pastry Shop Management System

Group Members:

Islam, Md. Nishadul (17-34724-2)
[Gave support on the project)]

Momu, Sharman Akter (18-36567-1)
[Provided the full information and structure of the ER diagram (100%) and drew it(100%).Did the table creation (100%), data insertion (100%), query writing (100%), and relational algebra (50%)]

Mojumder, MD. Fahim Montasir (19-41630-3)
[Took the screenshot of the created table (100%).Drew the Schema diagram (20%)]

Shathi, Sunjida Nourin (20-42597-1)
[Written the introduction part (30%). Provided group with necessary informationfrom project guideline (80%).Written the normalization (100%) and drew the schema diagram (80%)]

Toky, Golam Shahriar (20-42743-1)

[Made the full cover page and content page (100%). Wrote the scenario description(100%), Introduction part (70%), conclusion (100%)and the relational algebra (50%)]

INTRODUCTION TO DATABASE [E]

TABLE OF CONTENTS

Introduction

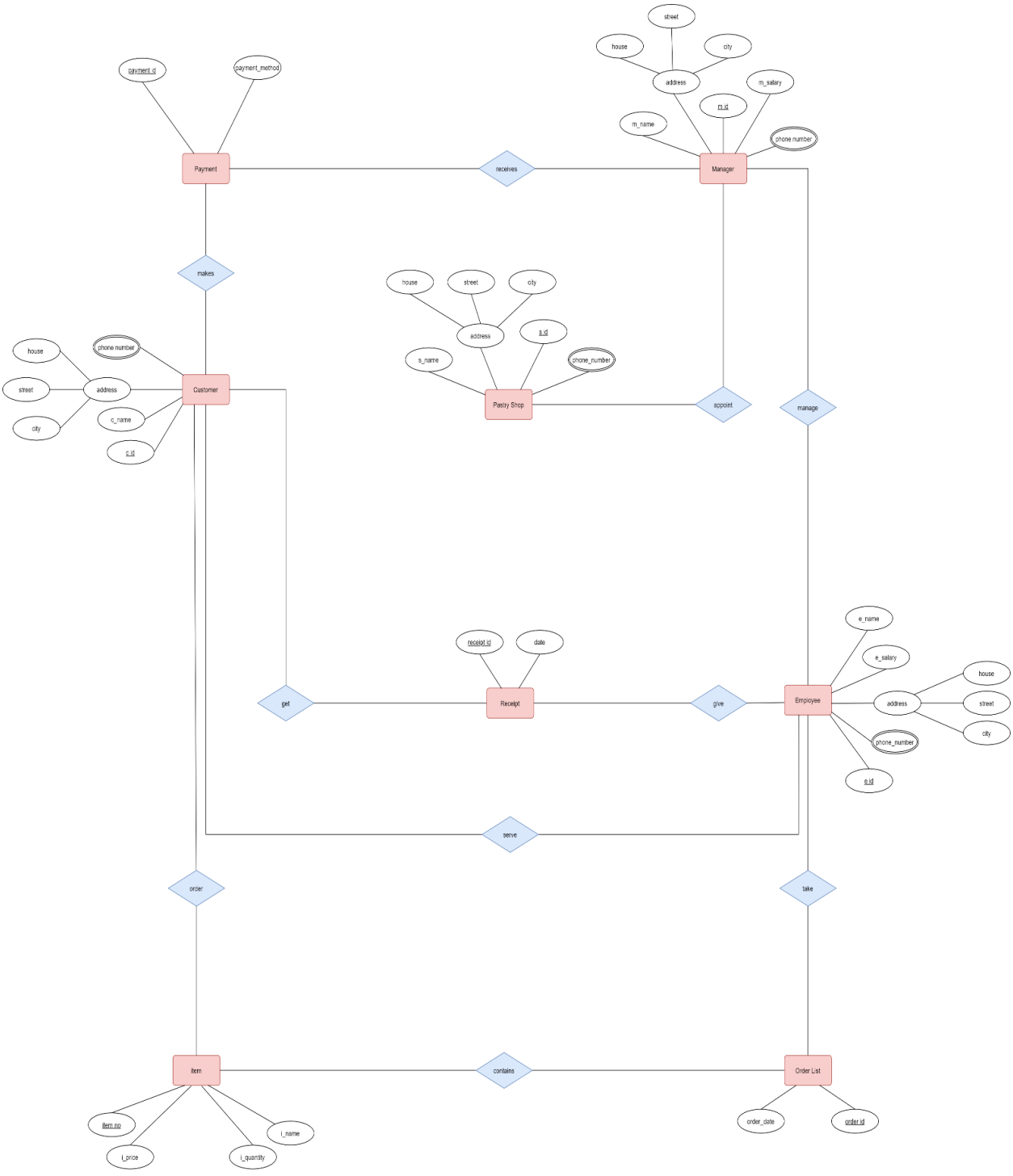
A pastry shop is the small kind of restaurant where mainly flour-based food is baked forthe customer. Foods that are served here are not meant to satisfy the greater hunger of the customers. Delicious foods mixed with the dazzling environment makes the pastry shop an ideal place for regular chit chat and casual hangout. The project “Pastry Shop Management System” is designed to record the information of the manager, employees and customers. It is built to facilitate and to make the best use of data by keeping the records in Database. Through this project, we can easily deal with the daily management system of a pastry shop. For this project we created eight tables. Showing the tables and their work as an overview we modeled an ER diagram. The basic aim of the project is to record, search, delete and insert data using SQL statements. The normalization and schema diagram are also the key elements for this project. Basically, the purpose of the project is to handle all the managements of a pastry shop.

SCENARIO DESCRIPTION

In this pastry shop management, the shop is recognized by its name, address, id and contact number. A manager holding a name, m_id, salary, address and phone number is being appointed to receive the payment from the customer and also to manage the other employees. On every address entity, attributes like house, street and city were added to make the address more specific. There a number of employees working under a manager so the relationship between employees and manager is many to one.The Employees working under the manager contain attributes like e_id, e_name, address, salary and phone number. Employee takes the order from the customer with an order List containing attributes like order_date and order id. One employee can take orders from many customers at a time. So, it is one to many relationships. Also, one customer can order many items. Item comprises four categories: item_no, i_price i_name, i_quantity. The customer has the attributes of name, customer_id, address and phone number. After being served from the employee, customer gets both the receipt and the food. The receipt includes receipt_id and date. Finally, the customer makes the payment. Payment contains information of payment method, payment_id. To make the payment many customers may approach to the manager making it a many to one relationship. Manager receiving the payment the customer may leave the shop.

ER Diagram

To have a clear view click here to see the web version.



NORMALIZATION

receives

UNF

receives (payment id, payment_method, m_id, m_name, house, street, city, m_salary, phone number)

1NF

Phone number is a multi-valued attribute.

1.payment id, payment_method, m_id, m_name, house, street, city, m_salary, phone number

2NF

1. payment id, payment_method
2. m_id, m_name, house, street, city, m_salary, phone number

3NF

1. payment id, payment_method
2. m_id, m_name, m_salary, phone number
3. house, street, city

Table Creation

1. payment id, payment_method
2. m_id, m_name, m_salary, phone number, **m_add_id**
3. m_add_id, house, street, city
4. **payment id**, **m_id**

makes

UNF

makes (payment id, payment_method, c_id, phone number, c_name, house, street, city)

1NF

Phone number is a multi-valued attribute.

1. payment id, payment_method, c_id, phone number, c_name, house, street, city

2NF

1. payment id, payment_method
2. c_id, phone number, c_name, house, street, city

3NF

1. payment id, payment_method
2. c_id, phone number, c_name
3. house, street, city

Table Creation

1. payment id, payment_method
2. c_id, phone number, c_name, **c_add_id**
3. c_add_id, house, street, city
4. **payment id**, **c_id**

appoint

UNF

appoint (m_id, m_name, house, street, city, m_salary, phone number, s_id, s_name, house, street, city, phone_number)

1NF

Phone number is a multi-valued attribute.

1. m_id, m_name, house, street, city, m_salary, phone number, s_id, s_name, house, street, city, phone_number

2NF

1. m_id, m_name, house, street, city, m_salary, phone number
2. s_id, s_name, house, street, city, phone_number

3NF

1. m_id, m_name, m_salary, phone number
2. house, street, city
3. s_id, s_name, phone _number
4. house, street, city

Table Creation

1. m_id, m_name, m_salary, phone number, **m_add_id**
2. m_add_id, house, street, city
3. s_id, s_name, phone_number, **s_add_id**
4. s_add_id, house, street, city
5. **m_id**, **s_id**

manage

UNF

manage (m_id, m_name, house, street, city, m_salary, phone number, e_id, e_name, e_salary, house, street, city, phone number)

1NF

Phone number is a multi-valued attribute.

1. m_id, m_name, house, street, city, m_salary, phone number, e_id, e_name, e_salary, house, street, city, phone number

2NF

1. m_id, m_name, house, street, city, m_salary, phone number
2. e_id, e_name, e_salary, house, street, city, phone number

3NF

1. m_id, m_name, m_salary, phone number
2. house, street, city
3. e_id, e_name, e_salary, phone number
4. house, street, city

Table Creation

1. m_id, m_name, m_salary, phone number, **m_add_id**
2. m_add_id, house, street, city
3. e_id, e_name, e_salary, phone number, **e_add_id**
4. e_add_id, house, street, city
5. **m_id**, **e_id**

get

UNF

get (c_id, c_name, house, street, city, phone number, receipt_id, date)

1NF

Phone_number is a multi-valued attribute.

1. c_id, c_name, house, street, city, phone number, receipt_id, date

2NF

1. c_id, c_name, house, street, city, phone number
2. receipt_id, date

3NF

1. c_id, c_name, phone number
2. house, street, city
3. receipt_id, date

Table Creation

1. c_id, c_name, phone number, **c_add_id**
2. c_add_id, house, street, city
3. receipt_id, date
4. **c_id**, **receipt_id**

give

UNF

give (receipt_id, date, e_id, e_name, e_salary, house, street, city, phone number)

1NF

phone number is a multi-valued attribute.

1. receipt_id, date, e_id, e_name, e_salary, house, street, city, phone number

2NF

1. receipt_id, date

2. e_id, e_name, e_salary, house, street, city, phone number

3NF

1. receipt_id, date
2. e_id, e_name, e_salary, phone number
3. house, street, city

Table Creation

1. receipt_id, date
2. e_id, e_name, e_salary, phone number, **e_add_id**
3. e_add_id, house, street, city
4. **receipt_id**, **e_id**

serve

UNF

serve (c_id, c_name, house, street, city, phone number, e_id, e_name, e_salary, house, street, city, phone number)

1NF

Phone number is a multi-valued attribute.

1. c_id, c_name, house, street, city, phone number, e_id, e_name, e_salary, house, street, city, phone number

2NF

1. c_id, c_name, house, street, city, phone number
2. e_id, e_name, e_salary, house, street, city, phone number

3NF

1. c_id, c_name, phone number
2. house, street, city
3. e_id, e_name, e_salary, phone number
4. house, street, city

Table Creation

1. c_id, c_name, phone number, **c_add_id**
2. c_add_id, house, street, city
3. e_id, e_name, e_salary, phone number, **e_add_id**
4. e_add_id, house, street, city
5. **c_id**, **e_id**

order

UNF

order (c_id, c_name, house, street, city, phone number, item_no, i_price, i_quantity, i_name)

1NF

Phone number is a multi-valued attribute.

1. c_id, c_name, house, street, city, phone number, item_no, i_price, i_quantity, i_name

2NF

1. c_id, c_name, house, street, city, phone number
2. item_no, i_price, i_quantity, i_name

3NF

1. c_id, c_name, phone number
2. house, street, city
3. item_no, i_price, i_quantity, i_name

Table Creation

1. c_id, c_name, phone number, **c_add_id**
2. c_add_id, house, street, city
3. item_no, i_price, i_quantity, i_name
4. **c_id**, **item_no**

take

UNF

take (e_id, e_name, e_salary, house, street, city, phone number, order_id, order_date)

1NF

Phone number is a multi-valued attribute.

1. e_id, e_name, e_salary, house, street, city, phone number, order_id, order_date

2NF

1. e_id, e_name, e_salary, house, street, city, phone number
2. order_id, order_date

3NF

1. e_id, e_name, e_salary, phone number
2. house, street, city
3. order_id, order_date

Table Creation

1. e_id, e_name, e_salary, phone number, **e_add_id**
2. e_add_id, house, street, city
3. order_id, order_date

4. **e_id**, **order_id**

contains

UNF

contains (item no, i_price, i_name, i_quantity, order id, order_date)

1NF

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

1. item no, i_price, i_name, i_quantity, order id, order_date

2NF

1. item no, i_price, i_name, i_quantity
2. order id, order_date

3NF

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

1. item no, i_price, i_name, i_quantity
2. order id, order_date

Table Creation

1. item no, i_price, i_name, i_quantity
2. order id, order_date
3. **item no**, **order id**

Temporary Tables

1. payment id, payment_method
2. ~~m_id, m_name, m_salary, phone number, **m_add_id**~~
3. ~~m_add_id, house, street, city~~
4. **payment id**, **m_id**
5. ~~payment id, payment_method~~
6. ~~c_id, phone number, c_name, **c_add_id**~~
7. c_add_id, house, street, city
8. **payment id**, **c_id**
9. ~~m_id, m_name, m_salary, phone number, **m_add_id**~~
10. ~~m_add_id, house, street, city~~
11. s_id, s_name, phone_number, **s_add_id**
12. s_add_id, house, street, city
13. **m_id**, **s_id**
14. m_id, m_name, m_salary, phone number, **m_add_id**
15. m_add_id, house, street, city
16. ~~e_id, e_name, e_salary, phone number, **e_add_id**~~
17. ~~e_add_id, house, street, city~~
18. **m_id**, **e_id**
19. ~~c_id, c_name, phone number, **c_add_id**~~
20. ~~c_add_id, house, street, city~~
21. ~~receipt_id, date~~
22. **c_id**, **receipt_id**
23. receipt_id, date
24. ~~e_id, e_name, e_salary, phone number, **e_add_id**~~
25. ~~e_add_id, house, street, city~~
26. **receipt_id**, **e_id**
27. ~~c_id, c_name, phone number, **c_add_id**~~
28. ~~c_add_id, house, street, city~~
29. ~~e_id, e_name, e_salary, phone number, **e_add_id**~~

30. ~~e_add_id, house, street, city~~
31. **c_id**, **e_id**
32. c_id, c_name, phone number, **c_add_id**
33. ~~c_add_id, house, street, city~~
34. ~~item_no, i_price, i_quantity, i_name~~
35. **c_id**, **item_no**
36. e_id, e_name, e_salary, phone number, **e_add_id**
37. e_add_id, house, street, city
38. ~~order_id, order_date~~
39. **e_id**, **order_id**
40. item no, i_price, i_name, i_quantity
41. order id, order_date
42. **item no**, **order id**

Final Tables

1. payment id, payment_method
2. **payment id**, **m_id**
3. c_add_id, house, street, city
4. **payment id**, **c_id**
5. s_id, s_name, phone_number1, phone_number2, **s_add_id**
6. s_add_id, house, street, city
7. **m_id**, **s_id**
8. m_id, m_name, m_salary, phone number1, phone_number2, phone_number3, **m_add_id**
9. m_add_id, house, street, city
10. **m_id**, **e_id**
11. **c_id**, **receipt_id**
12. receipt_id, date
13. **receipt_id**, **e_id**
14. **c_id**, **e_id**
15. c_id, c_name, phone number1, phone_number2, **c_add_id**

16. **c_id**, **item_no**
17. e_id, e_name, e_salary, phone_number1, phone_number2, **e_add_id**
18. e_add_id, house, street, city
19. **e_id**, **order_id**
20. item no, i_price, i_name, i_quantity
21. order id, order_date
22. **item no**, **order id**

Schema Diagram

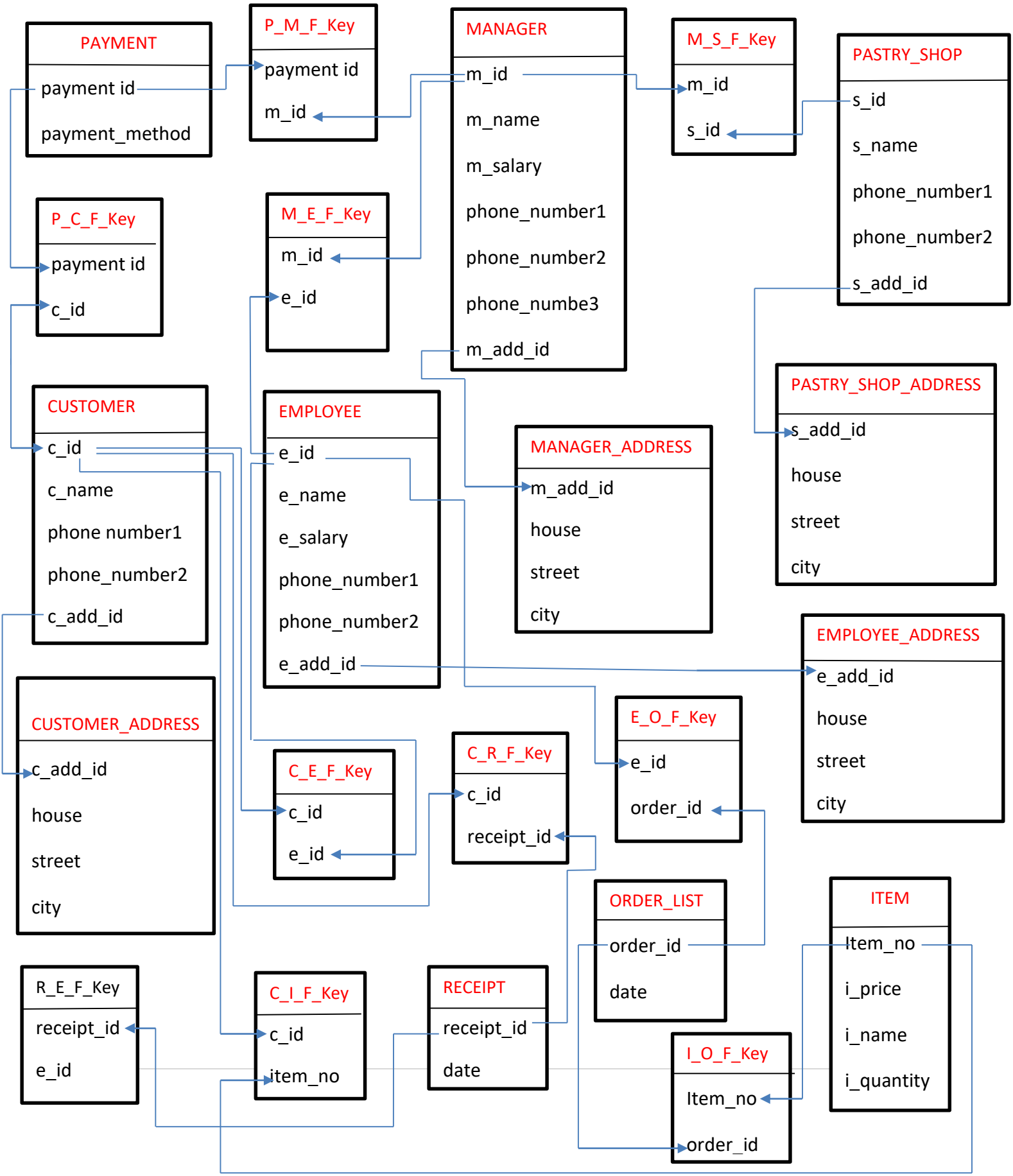


Table Creation

1. CREATE TABLE PAYMENT(PAYMENT_id NUMBER(10) PRIMARY KEY,
   PAYMENT_METHOD VARCHAR2(20));
2. CREATE TABLE P_M_F_Key(PAYMENT_id NUMBER(20), MANAGER_id
   NUMBER(10));
3. CREATE TABLE CUSTOMER_ADDRESS(C_Add_id NUMBER(10) PRIMARY KEY
   ,HOUSE NUMBER(10),Street VARCHAR2(20), City VARCHAR2(20));
4. CREATE TABLE P_C_F_Key(PAYMENT_id NUMBER(20), CUSTOMER_id
   NUMBER(10));
5. CREATE TABLE PASTRY_SHOP (PASTRY_SHOP_id NUMBER(10) PRIMARY KEY,
   S_NAME VARCHAR2(20), PHONE_NUMBER1 NUMBER(11), PHONE_NUMBER2
   NUMBER(11), S_ADD_id NUMBER(10));
6. CREATE TABLE PASTRY_SHOP_ADDRESS(S_Add_id NUMBER(10) PRIMARY KEY
   ,HOUSE NUMBER(10),Street VARCHAR2(20), City VARCHAR2(20));
7. CREATE TABLE M_S_F_Key(MANAGER_id NUMBER(20), PASTRY_SHOP_id
   NUMBER(10));
8. CREATE TABLE MANAGER (MANAGER_id NUMBER(10) PRIMARY KEY,
   M_NAME VARCHAR2(20),M_SALARY NUMBER(10), PHONE_NUMBER1

NUMBER(11), PHONE_NUMBER2 NUMBER(11),PHONE_NUMBER3 NUMBER(11), M_ADD_id NUMBER(10));

9. CREATE TABLE MANAGER_ADDRESS(M_Add_id NUMBER(10) PRIMARY KEY ,HOUSE NUMBER(10),Street VARCHAR2(20), City VARCHAR2(20));
10. CREATE TABLE M_E_F_Key(MANAGER_id NUMBER(20), EMPLOYEE_id NUMBER(10));
11. CREATE TABLE C_R_F_Key(CUSTOMER_id NUMBER(20), RECEIPT_id NUMBER(10));
12. CREATE TABLE RECEIPT(RECEIPT_id NUMBER(10) PRIMARY KEY,DT DATE);
13. CREATE TABLE R_E_F_Key(RECEIPT_id NUMBER(20), EMPLOYEE_id NUMBER(10));
14. CREATE TABLE C_E_F_Key(CUSTOMER_id NUMBER(20), EMPLOYEE_id NUMBER(10));
15. CREATE TABLE CUSTOMER (CUSTOMER_id NUMBER(10) PRIMARY KEY, C_NAME VARCHAR2(20), PHONE_NUMBER1 NUMBER(11), PHONE_NUMBER2 NUMBER(11), C_ADD_id NUMBER(10));
16. CREATE TABLE C_I_F_Key(CUSTOMER_id NUMBER(20), ITEM_NO NUMBER(10));

17. CREATE TABLE EMPLOYEE (EMPLOYEE_id NUMBER(10) PRIMARY KEY,E_NAME VARCHAR2(20),E_SALARY NUMBER(10), PHONE_NUMBER1 NUMBER(11), PHONE_NUMBER2 NUMBER(11),E_ADD_id NUMBER(10));

18. CREATE TABLE EMPLOYEE_ADDRESS(E_Add_id NUMBER(10) PRIMARY KEY ,HOUSE NUMBER(10),Street VARCHAR2(20), City VARCHAR2(20));

19. CREATE TABLE E_O_F_Key(EMPLOYEE_id NUMBER(20), ORDER_id NUMBER(10));

20. CREATE TABLE ITEM(ITEM_NO NUMBER(10) PRIMARY KEY, I_NAME VARCHAR2(20),I_PRICE NUMBER(10),I_QUANTITY NUMBER(20));

21. CREATE TABLE ORDER_LIST(ORDER_id NUMBER(10) PRIMARY KEY,DT DATE);

22. CREATE TABLE I_O_F_Key(ITEM_NO NUMBER(20), ORDER_id NUMBER(10));

1. ALTER TABLE PASTRY_SHOP ADD CONSTRAINT FK1 FOREIGN KEY (S_Add_id) REFERENCES PASTRY_SHOP_ADDRESS(S_Add_id);

2. ALTER TABLE MANAGER ADD CONSTRAINT FK2 FOREIGN KEY (M_Add_id) REFERENCES MANAGER_ADDRESS(M_Add_id);

3. ALTER TABLE CUSTOMER ADD CONSTRAINT FK3 FOREIGN KEY (C_Add_id)

REFERENCES CUSTOMER_ADDRESS(C_Add_id);

4. ALTER TABLE EMPLOYEE ADD CONSTRAINT FK4 FOREIGN KEY (E_Add_id)

REFERENCES EMPLOYEE_ADDRESS(E_Add_id);

5. ALTER TABLE M_E_F_Key ADD CONSTRAINT PK1 PRIMARY KEY

(MANAGER_id,EMPLOYEE_id);

6. ALTER TABLE M_E_F_Key ADD CONSTRAINT FK5 FOREIGN KEY (MANAGER_id)

REFERENCES MANAGER(MANAGER_id);

7. ALTER TABLE M_E_F_Key ADD CONSTRAINT FK6 FOREIGN KEY (EMPLOYEE_id)

REFERENCES EMPLOYEE(EMPLOYEE_id);

8. ALTER TABLE C_E_F_Key ADD CONSTRAINT PK2 PRIMARY KEY

(CUSTOMER_id,EMPLOYEE_id);

9. ALTER TABLE C_E_F_Key ADD CONSTRAINT FK7 FOREIGN KEY (CUSTOMER_id)

REFERENCES CUSTOMER(CUSTOMER_id);

10. ALTER TABLE C_E_F_Key ADD CONSTRAINT FK8 FOREIGN KEY (EMPLOYEE_id)

REFERENCES EMPLOYEE(EMPLOYEE_id);

11. ALTER TABLE M_S_F_Key ADD CONSTRAINT PK3 PRIMARY KEY

(MANAGER_id,PASTRY_SHOP_id);

12. ALTER TABLE M_S_F_Key ADD CONSTRAINT FK9 FOREIGN KEY (MANAGER_id)

REFERENCES MANAGER(MANAGER_id);

13. ALTER TABLE M_S_F_Key ADD CONSTRAINT FK10 FOREIGN KEY

(PASTRY_SHOP_id) REFERENCES PASTRY_SHOP(PASTRY_SHOP_id);

14. ALTER TABLE P_M_F_Key ADD CONSTRAINT PK4 PRIMARY KEY

(PAYMENT_id,MANAGER_id);

15. ALTER TABLE P_M_F_Key ADD CONSTRAINT FK11 FOREIGN KEY (PAYMENT_id)

REFERENCES PAYMENT(PAYMENT_id);

16. ALTER TABLE P_M_F_Key ADD CONSTRAINT FK12 FOREIGN KEY

(MANAGER_id) REFERENCES MANAGER(MANAGER_id);

17. ALTER TABLE P_C_F_Key ADD CONSTRAINT PK5 PRIMARY KEY

(PAYMENT_id,CUSTOMER_id);

18. ALTER TABLE P_C_F_Key ADD CONSTRAINT FK13 FOREIGN KEY (PAYMENT_id)

REFERENCES PAYMENT(PAYMENT_id);

19. ALTER TABLE P_C_F_Key ADD CONSTRAINT FK14 FOREIGN KEY (CUSTOMER_id) REFERENCES CUSTOMER(CUSTOMER_id);

20. ALTER TABLE C_I_F_Key ADD CONSTRAINT PK6 PRIMARY KEY (CUSTOMER_id,ITEM_NO);

21. ALTER TABLE C_I_F_Key ADD CONSTRAINT FK15 FOREIGN KEY (CUSTOMER_id) REFERENCES CUSTOMER(CUSTOMER_id);

22. ALTER TABLE C_I_F_Key ADD CONSTRAINT FK16 FOREIGN KEY (ITEM_NO) REFERENCES ITEM(ITEM_NO);

23. ALTER TABLE E_O_F_Key ADD CONSTRAINT PK7 PRIMARY KEY (EMPLOYEE_id, ORDER_ID);

24. ALTER TABLE E_O_F_Key ADD CONSTRAINT FK17 FOREIGN KEY (EMPLOYEE_id) REFERENCES EMPLOYEE(EMPLOYEE_id);

25. ALTER TABLE E_O_F_Key ADD CONSTRAINT FK18 FOREIGN KEY (ORDER_id) REFERENCES ORDER_LIST(ORDER_id);

26. ALTER TABLE R_E_F_Key ADD CONSTRAINT PK8 PRIMARY KEY (RECEIPT_id, EMPLOYEE_ID);

27. ALTER TABLE R_E_F_Key ADD CONSTRAINT FK19 FOREIGN KEY (EMPLOYEE_id) REFERENCES EMPLOYEE(EMPLOYEE_id);

28. ALTER TABLE R_E_F_Key ADD CONSTRAINT FK20 FOREIGN KEY (RECEIPT_id) REFERENCES RECEIPT(RECEIPT_id);

29. ALTER TABLE I_O_F_Key ADD CONSTRAINT PK9 PRIMARY KEY (ITEM_NO, ORDER_ID);

30. ALTER TABLE I_O_F_Key ADD CONSTRAINT FK21 FOREIGN KEY (ORDER_id) REFERENCES ORDER_LIST(ORDER_id);

31. ALTER TABLE I_O_F_Key ADD CONSTRAINT FK22 FOREIGN KEY (ITEM_NO) REFERENCES ITEM(ITEM_NO);

Screenshots of created table after using describe command

1. PAYMENT Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object PAYMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	PAYMENT_ID	Number	-	10	0	1	-	-	-
	PAYMENT_METHOD	Varchar2	20	-	-	-	✓	-	-

1 - 2

2. P_M_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object P_M_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
P_M_F_KEY	PAYMENT_ID	Number	-	20	0	1	-	-	-
	MANAGER_ID	Number	-	10	0	2	-	-	-

1 - 2

3. CUSTOMER_ADDRESS Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMER_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_ADDRESS	C_ADD_ID	Number	-	10	0	1	-	-	-
	HOUSE	Number	-	10	0	-	✓	-	-
	STREET	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 4

4. P_C_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object P_C_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
P_C_F_KEY	PAYMENT_ID	Number	-	20	0	1	-	-	-
	CUSTOMER_ID	Number	-	10	0	2	-	-	-

1 - 2

5. PASTRY_SHOP Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object PASTRY_SHOP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASTRY_SHOP	PASTRY_SHOP_ID	Number	-	10	0	1	-	-	-
	S_NAME	Varchar2	20	-	-	-	✓	-	-
	PHONE_NUMBER1	Number	-	11	0	-	✓	-	-
	PHONE_NUMBER2	Number	-	11	0	-	✓	-	-
	S_ADD_ID	Number	-	10	0	-	✓	-	-

1 - 5

6. PASTRY_SHOP_ADDRESS Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object PASTRY_SHOP_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASTRY_SHOP_ADDRESS	S_ADD_ID	Number	-	10	0	1	-	-	-
	HOUSE	Number	-	10	0	-	✓	-	-
	STREET	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 4

7. M_S_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object M_S_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
M_S_F_KEY	MANAGER_ID	Number	-	20	0	1	-	-	-
	PASTRY_SHOP_ID	Number	-	10	0	2	-	-	-

1 - 2

8. MANAGER Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MANAGER_ID	Number	-	10	0	1	-	-	-
	M_NAME	Varchar2	20	-	-	-	✓	-	-
	M_SALARY	Number	-	10	0	-	✓	-	-
	PHONE_NUMBER1	Number	-	11	0	-	✓	-	-
	PHONE_NUMBER2	Number	-	11	0	-	✓	-	-
	PHONE_NUMBER3	Number	-	11	0	-	✓	-	-
	M_ADD_ID	Number	-	10	0	-	✓	-	-

1 - 7

9. MANAGER_ADDRESS Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object MANAGER_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER_ADDRESS	M_ADD_ID	Number	-	10	0	1	-	-	-
	HOUSE	Number	-	10	0	-	✓	-	-
	STREET	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 4

10. M_E_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object M_E_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
M_E_F_KEY	MANAGER_ID	Number	-	20	0	1	-	-	-
	EMPLOYEE_ID	Number	-	10	0	2	-	-	-

1 - 2

11. C_R_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object C_R_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
C_R_F_KEY	CUSTOMER_ID	Number	-	20	0	-	✓	-	-
	RECEIPT_ID	Number	-	10	0	-	✓	-	-

1 - 2

12. RECEIPT Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object RECEIPT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECEIPT	RECEIPT_ID	Number	-	10	0	1	-	-	-
	DT	Date	7	-	-	-	✓	-	-

1 - 2

13. R_E_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object R_E_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
R_E_F_KEY	RECEIPT_ID	Number	-	20	0	1	-	-	-
	EMPLOYEE_ID	Number	-	10	0	2	-	-	-

1 - 2

14. C_E_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object C_E_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
C_E_F_KEY	CUSTOMER_ID	Number	-	20	0	1	-	-	-
	EMPLOYEE_ID	Number	-	10	0	2	-	-	-

1 - 2

15. CUSTOMER Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Number	-	10	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	PHONE_NUMBER1	Number	-	11	0	-	✓	-	-
	PHONE_NUMBER2	Number	-	11	0	-	✓	-	-
	C_ADD_ID	Number	-	10	0	-	✓	-	-

1 - 5

16. C_I_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object C_I_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
C_I_F_KEY	CUSTOMER_ID	Number	-	20	0	1	-	-	-
	ITEM_NO	Number	-	10	0	2	-	-	-

1 - 2

17. EMPLOYEE Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMPLOYEE_ID	Number	-	10	0	1	-	-	-
	E_NAME	Varchar2	20	-	-	-	✓	-	-
	E_SALARY	Number	-	10	0	-	✓	-	-
	PHONE_NUMBER1	Number	-	11	0	-	✓	-	-
	PHONE_NUMBER2	Number	-	11	0	-	✓	-	-
	E_ADD_ID	Number	-	10	0	-	✓	-	-

1 - 6

18. EMPLOYEE_ADDRESS Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE_ADDRESS	E_ADD_ID	Number	-	10	0	1	-	-	-
	HOUSE	Number	-	10	0	-	✓	-	-
	STREET	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-

1 - 4

19. E_O_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object E_O_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
E_O_F_KEY	EMPLOYEE_ID	Number	-	20	0	1	-	-	-
	ORDER_ID	Number	-	10	0	2	-	-	-

1 - 2

20. ITEM Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object ITEM

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ITEM	ITEM_NO	Number	-	10	0	1	-	-	-
	I_NAME	Varchar2	20	-	-	-	✓	-	-
	I_PRICE	Number	-	10	0	-	✓	-	-
	I_QUANTITY	Number	-	20	0	-	✓	-	-

1 - 4

21. ORDER_LIST Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object ORDER_LIST

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_LIST	ORDER_ID	Number	-	10	0	1	-	-	-
	DT	Date	7	-	-	-	✓	-	-

1 - 2

22. I_O_F_Key Table :

Results Explain Describe Saved SQL History

Object Type TABLE Object I_O_F_KEY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
I_O_F_KEY	ITEM_NO	Number	-	20	0	1	-	-	-
	ORDER_ID	Number	-	10	0	2	-	-	-

1 - 2

Create Sequence

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

Create Users, assign roles & grant privileges

1. CREATE USER MUMU

```
IDENTIFIED BY MUMU123;
GRANT UNLIMITED TABLESPACE TO MUMU;
```

2. CREATE USER NISADUL

```
IDENTIFIED BY NIS123;
GRANT UNLIMITED TABLESPACE TO NISADUL;
```

3. CREATE USER TOKY

```
IDENTIFIED BY TOKY123;
GRANT UNLIMITED TABLESPACE TO TOKY;
```

4. CREATE USER SATHI

```
IDENTIFIED BY SATHI123;
GRANT UNLIMITED TABLESPACE TO SATHI;
```

5. CREATE USER FAHIM

IDENTIFIED BY FAHIM123;

GRANT UNLIMITED TABLESPACE TO FAHIM;

1.CREATE ROLE MANAGER;

GRANT create table, create view, create sequence, create user, create role, create session, create procedure TO MANAGER;

GRANT MANAGER TO NISADUL,SATHI ,MUMU,FAHIM,TOKY;

Data Insertion

1. PAYMENT TABLE

```
INSERT INTO PAYMENT VALUES (seq.NEXTVAL, 'CASH');

INSERT INTO PAYMENT  VALUES (seq.NEXTVAL, 'BKASH');

INSERT INTO PAYMENT VALUES (seq.NEXTVAL, 'CREDIT CARD');

INSERT INTO PAYMENT VALUES (seq.NEXTVAL, 'NAGAD');

INSERT INTO PAYMENT VALUES (seq.NEXTVAL, 'ROCKET');
```

2. MANAGER_ADDRESS TABLE

```
INSERT INTO MANAGER_ADDRESS VALUES (seq.NEXTVAL, '342',
'MIRPUR','DHAKA');

INSERT INTO MANAGER_ADDRESS  VALUES (seq.NEXTVAL, '343','Uttara','Dhaka');

INSERT INTO MANAGER_ADDRESS  VALUES
(seq.NEXTVAL,'453','Kazipara','Rangpur');

INSERT INTO MANAGER_ADDRESS  VALUES (seq.NEXTVAL,
'675','Ullapara','Sylhet');

INSERT INTO MANAGER_ADDRESS  VALUES (seq.NEXTVAL,
'897','Sornogachi','Chittagong');
```

3. MANAGER TABLE

```
INSERT INTO MANAGER VALUES (seq.NEXTVAL, 'MUMU',
'3500','017453','091563','012647','10');

INSERT INTO MANAGER  VALUES (seq.NEXTVAL,
'NISADUL','3600','018453','061563','072647','11');

INSERT INTO MANAGER  VALUES
(seq.NEXTVAL,'SATHI','3700','017653','091763','019647','12');

INSERT INTO MANAGER  VALUES (seq.NEXTVAL,
'TOKY','3600','017453','021563','016647','13');

INSERT INTO MANAGER  VALUES (seq.NEXTVAL,
'FAHIM','3700','017493','091863','012947','14');
```

4. P_M_F_KEY TABLE

```
INSERT INTO P_M_F_Key VALUES (1,15);

INSERT INTO P_M_F_Key VALUES (2,16);

INSERT INTO P_M_F_Key VALUES (3,17);

INSERT INTO P_M_F_Key VALUES (4,18);

INSERT INTO P_M_F_Key VALUES (5,19);
```

5. PASTRY_SHOP_ADDRESS TABLE

```
INSERT INTO PASTRY_SHOP_ADDRESS VALUES ('1243', '362','MIRPUR 2','DHAKA');
```

6. PASTRY_SHOP TABLE

```
INSERT INTO PASTRY_SHOP VALUES ('943', 'CAKE N
BAKE','017468','0173655','1243');
```

7. M_S_F_KEY TABLE

```
INSERT INTO M_S_F_KEY VALUES ('15','943' );

INSERT INTO M_S_F_KEY VALUES ('16','943' );

INSERT INTO M_S_F_KEY VALUES ('17','943' );

INSERT INTO M_S_F_KEY VALUES ('18','943' );

INSERT INTO M_S_F_KEY VALUES ('19','943' );
```

8. CUSTOMER_ADDRESS TABLE

```
INSERT INTO CUSTOMER_ADDRESS VALUES (seq.NEXTVAL, '098','MIRPUR
11','DHAKA');
```

```
INSERT INTO CUSTOMER_ADDRESS VALUES(seq.NEXTVAL,'087','MIRPUR
12','DHAKA' );

INSERT INTO CUSTOMER_ADDRESS VALUES(seq.NEXTVAL, '765','MIRPUR
1','DHAKA');

INSERT INTO CUSTOMER_ADDRESS VALUES(seq.NEXTVAL, '095','MIRPUR
10','DHAKA');

INSERT INTO CUSTOMER_ADDRESS VALUES(seq.NEXTVAL, '045','MIRPUR
6','DHAKA');
```

9. CUSTOMER TABLE

```
INSERT INTO CUSTOMER VALUES (seq.NEXTVAL, 'SAKIB','0178659','0167543','20');

INSERT INTO CUSTOMER VALUES (seq.NEXTVAL,
'RAFSAN','0198659','0169543','21');

INSERT INTO CUSTOMER VALUES (seq.NEXTVAL,
'MUSHFIQ','0188659','0157543','22');

INSERT INTO CUSTOMER VALUES (seq.NEXTVAL,
'NOWRIN','0178959','0127543','23');

INSERT INTO CUSTOMER VALUES (seq.NEXTVAL, 'SRUTI','0156659','0187543','24');
```

10. P_C_F_KEY TABLE

```
INSERT INTO P_C_F_Key VALUES (1,25);

INSERT INTO P_C_F_Key VALUES (2,26);
```

```
INSERT INTO P_C_F_Key VALUES (3,27);

INSERT INTO P_C_F_Key VALUES (4,28);

INSERT INTO P_C_F_Key VALUES (5,29);
```

11. EMPLOYEE_ADDRESS TABLE

```
INSERT INTO EMPLOYEE_ADDRESS VALUES (seq.NEXTVAL, '088','MIRPUR
2','DHAKA');

INSERT INTO EMPLOYEE_ADDRESS VALUES(seq.NEXTVAL,'057','MIRPUR
12','DHAKA' );

INSERT INTO EMPLOYEE_ADDRESS VALUES(seq.NEXTVAL, '235','MIRPUR
11','DHAKA');

INSERT INTO EMPLOYEE_ADDRESS VALUES(seq.NEXTVAL, '765','MIRPUR
1','DHAKA');

INSERT INTO EMPLOYEE_ADDRESS VALUES(seq.NEXTVAL, '095','MIRPUR
6','DHAKA');
```

12. EMPLOYEE TABLE

```
INSERT INTO EMPLOYEE VALUES (seq.NEXTVAL,
'RAFIM','2500','0166543','017344','30');

INSERT INTO EMPLOYEE VALUES (seq.NEXTVAL,
'RAFI','1500','01897643','0156446','31');
```

```
INSERT INTO EMPLOYEE VALUES (seq.NEXTVAL,
'MAHDI','2000','0158659','0155543','32');

INSERT INTO EMPLOYEE VALUES (seq.NEXTVAL,
'TAMIM','2500','01797959','0156543','33');

INSERT INTO EMPLOYEE VALUES (seq.NEXTVAL,
'ABID','3000','0198659','01897543','34');
```

13. M_E_F_KEY TABLE

```
INSERT INTO M_E_F_KEY VALUES ('15','35' );

INSERT INTO M_E_F_KEY VALUES ('16','36' );

INSERT INTO M_E_F_KEY VALUES ('17','37' );

INSERT INTO M_E_F_KEY VALUES ('18','38' );

INSERT INTO M_E_F_KEY VALUES ('19','39' );
```

14. C_E_F_KEY TABLE

```
INSERT INTO C_E_F_KEY VALUES ('25','35' );

INSERT INTO C_E_F_KEY VALUES ('26','36' );

INSERT INTO C_E_F_KEY VALUES ('27','37' );

INSERT INTO C_E_F_KEY VALUES ('28','38' );

INSERT INTO C_E_F_KEY VALUES ('29','39' );
```

15. ITEM TABLE

```
INSERT INTO ITEM VALUES (seq.NEXTVAL, 'BLACK FOREST PASTRY','150','10');
INSERT INTO ITEM VALUES (seq.NEXTVAL, 'RED VELVET PASTRY','200','15');
INSERT INTO ITEM VALUES (seq.NEXTVAL, 'WHITE FOREST PASTRY','120','20');
INSERT INTO ITEM VALUES (seq.NEXTVAL, 'CUP CAKE','140','17');
INSERT INTO ITEM VALUES (seq.NEXTVAL, 'Éclair','300','10');
```

16. ORDER_LIST TABLE

```
INSERT INTO ORDER_LIST VALUES (seq.NEXTVAL,'01 JANUARY 2019');
INSERT INTO ORDER_LIST VALUES (seq.NEXTVAL,'01 JANUARY 2019');
INSERT INTO ORDER_LIST VALUES (seq.NEXTVAL, '11 JANUARY 2019');
INSERT INTO ORDER_LIST VALUES (seq.NEXTVAL, '15 JANUARY 2019');
INSERT INTO ORDER_LIST VALUES (seq.NEXTVAL, '17 JANUARY 2019');
```

17. RECEIPT TABLE

```
INSERT INTO RECEIPT VALUES (seq.NEXTVAL,'01 JANUARY 2019');
INSERT INTO RECEIPT VALUES (seq.NEXTVAL,'01 JANUARY 2019');
INSERT INTO RECEIPT VALUES (seq.NEXTVAL, '11 JANUARY 2019');
INSERT INTO RECEIPT VALUES (seq.NEXTVAL, '15 JANUARY 2019');
```

```
INSERT INTO RECEIPT VALUES (seq.NEXTVAL, '17 JANUARY 2019');
```

18. C_R_F_KEY TABLE

```
INSERT INTO C_R_F_Key VALUES (25,55);
INSERT INTO C_R_F_Key VALUES (26,56);
INSERT INTO C_R_F_Key VALUES (27,57);
INSERT INTO C_R_F_Key VALUES (28,58);
INSERT INTO C_R_F_Key VALUES (29,59);
```

19. R_E_F_KEY TABLE

```
INSERT INTO R_E_F_KEY VALUES ('55','35' );
INSERT INTO R_E_F_KEY VALUES ('56','36' );
INSERT INTO R_E_F_KEY VALUES ('57','37' );
INSERT INTO R_E_F_KEY VALUES ('58','38' );
INSERT INTO R_E_F_KEY VALUES ('59','39' );
```

20. C_I_F_KEY TABLE

```
INSERT INTO C_I_F_Key VALUES (25,40);
INSERT INTO C_I_F_Key VALUES (26,41);
```

```
INSERT INTO C_I_F_Key VALUES (27,42);

INSERT INTO C_I_F_Key VALUES (28,43);

INSERT INTO C_I_F_Key VALUES (29,44);
```

21. E_O_F_KEY TABLE

```
INSERT INTO E_O_F_KEY VALUES ('35','50' );

INSERT INTO E_O_F_KEY VALUES ('36','51' );

INSERT INTO E_O_F_KEY VALUES ('37','52' );

INSERT INTO E_O_F_KEY VALUES ('38','53' );

INSERT INTO E_O_F_KEY VALUES ('39','54' );
```

22. I_O_F_KEY TABLE

```
INSERT INTO I_O_F_KEY VALUES ('40','50' );

INSERT INTO I_O_F_KEY VALUES ('41','51' );

INSERT INTO I_O_F_KEY VALUES ('42','52' );

INSERT INTO I_O_F_KEY VALUES ('43','53' );

INSERT INTO I_O_F_KEY VALUES ('44','54' );
```

Screenshots of The Tables After Inserting Data

1. Payment Table Data

Results Explain Describe Saved SQL History

PAYMENT_ID	PAYMENT_METHOD
1	CASH
2	BKASH
3	CREDIT CARD
4	NAGAD
5	ROCKET

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

2. Manager Address Table Data

Results Explain Describe Saved SQL History

M_ADD_ID	HOUSE	STREET	CITY
10	342	MIRPUR	DHAKA
11	343	Uttara	Dhaka
12	453	Kazipara	Rangpur
13	675	Ullapara	Sylhet
14	897	Sornogachi	Chittagong

3. Manager Table Data

Results Explain Describe Saved SQL History

MANAGER_ID	M_NAME	M_SALARY	PHONE_NUMBER1	PHONE_NUMBER2	PHONE_NUMBER3	M_ADD_ID
15	MUMU	3500	17453	91563	12647	10
16	NISADUL	3600	18453	61563	72647	11
17	SATHI	3700	17653	91763	19647	12
18	TOKY	3600	17453	21563	16647	13
19	FAHIM	3700	17493	91863	12947	14

5 rows returned in 0.02 seconds CSV Export

Application Express 2.1.0.00.39

4. P_M_F_Key Table Data

Results Explain Describe Saved SQL History

PAYMENT_ID	MANAGER_ID
1	15
2	16
3	17
4	18
5	19

5 rows returned in 0.00 seconds CSV Export

5. Pastry_Shop_Address Table Data

Results Explain Describe Saved SQL History

S_ADD_ID	HOUSE	STREET	CITY
1243	362	MIRPUR 2	DHAKA

1 rows returned in 0.00 seconds CSV Export

6. Pastry_Shop Table Data

Results Explain Describe Saved SQL History

PASTRY_SHOP_ID	S_NAME	PHONE_NUMBER1	PHONE_NUMBER2	S_ADD_ID
943	CAKE N BAKE	17468	173655	1243

1 rows returned in 0.00 seconds CSV Export

7. M_S_F_Key Table Data

Results Explain Describe Saved SQL History

MANAGER_ID	PASTRY_SHOP_ID
15	943
16	943
17	943
18	943
19	943

5 rows returned in 0.02 seconds CSV Export

Application Express 2.1.0.00.39

8. Customer_Address Table Data

Results Explain Describe Saved SQL History

C_ADD_ID	HOUSE	STREET	CITY
20	98	MIRPUR 11	DHAKA
21	87	MIRPUR 12	DHAKA
22	765	MIRPUR 1	DHAKA
23	95	MIRPUR 10	DHAKA
24	45	MIRPUR 6	DHAKA

5 rows returned in 0.01 seconds CSV Export

Application Express 2.1.0.00.39

9. Customer Table Data

Results Explain Describe Saved SQL History

CUSTOMER_ID	C_NAME	PHONE_NUMBER1	PHONE_NUMBER2	C_ADD_ID
25	SAKIB	178659	167543	20
26	RAFSAN	198659	169543	21
27	MUSHFIQ	188659	157543	22
28	NOWRIN	178959	127543	23
29	SRUTI	156659	187543	24

5 rows returned in 0.00 seconds CSV Export

10. P_C_F_Key Table Data

Results Explain Describe Saved SQL History

PAYMENT_ID	CUSTOMER_ID
1	25
2	26
3	27
4	28
5	29

5 rows returned in 0.00 seconds CSV Export

11.Employee_Address Table Data

Results Explain Describe Saved SQL History

E_ADD_ID	HOUSE	STREET	CITY
30	88	MIRPUR 2	DHAKA
31	57	MIRPUR 12	DHAKA
32	235	MIRPUR 11	DHAKA
33	765	MIRPUR 1	DHAKA
34	95	MIRPUR 6	DHAKA

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

12. Employee Table Data

Results Explain Describe Saved SQL History

EMPLOYEE_ID	E_NAME	E_SALARY	PHONE_NUMBER1	PHONE_NUMBER2	E_ADD_ID
35	RAFIM	2500	166543	17344	30
36	RAFI	1500	1897643	156446	31
37	MAHDI	2000	158659	155543	32
38	TAMIM	2500	1797959	156543	33
39	ABID	3000	198659	1897543	34

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

13. M_E_F_Key Table Data

Results Explain Describe Saved SQL History

MANAGER_ID	EMPLOYEE_ID
15	35
16	36
17	37
18	38
19	39

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

14. C_E_F_Key Table Data

Results Explain Describe Saved SQL History

CUSTOMER_ID	EMPLOYEE_ID
25	35
26	36
27	37
28	38
29	39

5 rows returned in 0.01 seconds CSV Export

Application Express 2.1.0.00.39

15. Item Table Data

Results Explain Describe Saved SQL History

ITEM_NO	I_NAME	I_PRICE	I_QUANTITY
40	BLACK FOREST PASTRY	150	10
41	RED VELVET PASTRY	200	15
42	WHITE FOREST PASTRY	120	20
43	CUP CAKE	140	17
44	Éclair	300	10

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

16. Order_List Table Data

Results Explain Describe Saved SQL History

ORDER_ID	DT
50	01-JAN-19
51	01-JAN-19
52	11-JAN-19
53	15-JAN-19
54	17-JAN-19

5 rows returned in 0.00 seconds CSV Export

17. Receipt Table Data

Results Explain Describe Saved SQL History

RECEIPT_ID	DT
55	01-JAN-19
56	01-JAN-19
57	11-JAN-19
58	15-JAN-19
59	17-JAN-19

5 rows returned in 0.00 seconds CSV Export

18. C_R_F_Key Table Data

Results Explain Describe Saved SQL History

CUSTOMER_ID	RECEIPT_ID
25	55
26	56
27	57
28	58
29	59

5 rows returned in 0.00 seconds CSV Export

19. R_E_F_key Table Data

Results Explain Describe Saved SQL History

RECEIPT_ID	EMPLOYEE_ID
55	35
56	36
57	37
58	38
59	39

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39

20. C_I_F_Key Table Data

Results Explain Describe Saved SQL History

CUSTOMER_ID	ITEM_NO
25	40
26	41
27	42
28	43
29	44

5 rows returned in 0.01 seconds CSV Export

21. E_O_F_Key Table Data

Results Explain Describe Saved SQL History

EMPLOYEE_ID	ORDER_ID
35	50
36	51
37	52
38	53
39	54

5 rows returned in 0.00 seconds CSV Export

22. I_O_F_Key Table Data

Results Explain Describe Saved SQL History

ITEM_NO	ORDER_ID
40	50
41	51
42	52
43	53
44	54

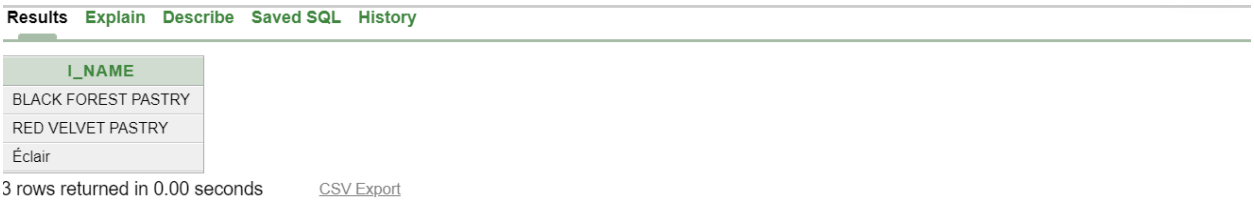
5 rows returned in 0.00 seconds CSV Export

Query Writing

Subquery

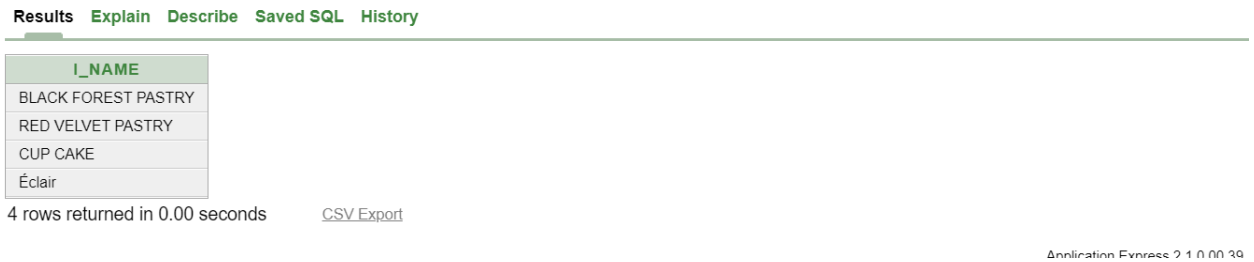
1. Which items price are more than the price of cup cake?

Answer: select I_NAME FROM ITEM WHERE I_PRICE > (SELECT I_PRICE FROM ITEM WHERE I_NAME= 'CUP CAKE');



2. Which items quantity are less than the quantity of items where the word contains “ite” along with other words?

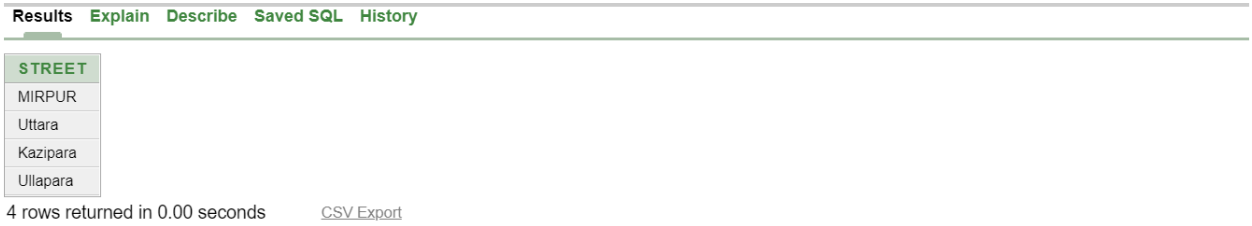
Answer: select I_NAME FROM ITEM where I_QUANTITY < (SELECT I_QUANTITY FROM ITEM WHERE I_NAME LIKE '%ITE%');



Joining

1.Display the name of all the street of manager address which are outside of Chittagong.

Answer: Select MA.STREET FROM MANAGER_ADDRESS MA, MANAGER M WHERE MA.M_ADD_ID = M.M_ADD_ID AND CITY NOT IN (' Chittagong ');



2. Display the name of the street of customer address which are located in Dhaka.

Answer : Select CA.STREET FROM CUSTOMER_ADDRESS CA, CUSTOMER C WHERE CA.C_ADD_ID = C.C_ADD_ID AND CITY IN ('DHAKA');

View

1. Create a View called C_add_view based on the street and city from the Customer_Address table where city is Dhaka.

Answer: CREATE VIEW C_ADD_VIEW AS SELECT STREET,CITY FROM CUSTOMER_ADDRESS WHERE CITY= 'DHAKA';

Results Explain Describe Saved SQL History

STREET	CITY
MIRPUR 11	DHAKA
MIRPUR 12	DHAKA
MIRPUR 1	DHAKA
MIRPUR 10	DHAKA
MIRPUR 6	DHAKA

5 rows returned in 0.00 seconds CSV Export

2. Create a view called item_view based on the i_name and i_quantity from item table where price is more than 100

Answer: CREATE VIEW ITEM_VIEW AS SELECT I_NAME,I_QUANTITY FROM ITEM WHERE I_PRICE > 100;

Results Explain Describe Saved SQL History

I_NAME	I_QUANTITY
BLACK FOREST PASTRY	10
RED VELVET PASTRY	15
WHITE FOREST PASTRY	20
CUP CAKE	17
Éclair	10

5 rows returned in 0.00 seconds CSV Export

Relational Algebra

1. Find the name of the manager where manager id is 15.

Answer: π m_name (σmanager_id = “15” (MANAGER))

2. Find the name of the item where item no is 42.

Answer: π i_name (σitem_no = “42” (ITEM))

3. Find the payment id where payment method is BKASH.

Answer: π payment_id (σpayment_method = “BKASH” (PAYMENT))

4. Find the id of the customer whose name is SAKIB

Answer: π customer_id (σc_name = “SAKIB” (CUSTOMER))

5. Find the street name of the employee where employee address id is 30.

Answer: π street (σe_add_id = “30” (EMPLOYEE_ADDRESS))

CONCLUSION

In the pastry shop management system project, the scenario description was written to give a brief overview of the whole project. The gist part of the project was done by making the ER diagram and doing the normalization to avoid redundancy. The schema diagram helped us putting only the related data. On the table creation part, the predesigned tables were created and on the insertion part data was being stored. The project ended giving some relational algebra and queries. By this effective database project, the organizational data accessibility will be lot quicker and efficient. Both the customers and the shop will be able to save their time and effort.

The project is planned to improve by ensuring more security to protect the data. Moreover, making the shop a franchise will add some new tables and data to the project. Also, the future plan is to make the customer experience more convenient.