Experiment No :07

Name:-Neeraj Avinash Chormale
PRN:- 20220802071
Batch:- A2

Aim : to perform joins and view operation on following data base.

Reqirement:- Computer with mysql installed.

Objective : to create view based on table(s) or view(s) and observe its behavior while performing update operations on it. Theory:Bakery dataset

The dataset contains information about one month worth of sales information for a small bakery shop. The sales are made to known customers. The dataset contains

Information about the customers, the assortments of baked goods offered for sale and the purchases made.

The dataset consists of the following relations:

- customers : information about the bakery's customers

- products : information about the baked goods offered for sale by the bakery

- item_list : itemized receipt information for purchases

- receipts : general receipt information for purchases

Receipts stores information about individual receipts (purchases by customers). Each purchase may contain from one to five items, regardless of whether any items purchased are of the same kind (e.g., two "chocolate cakes" will be billed as two separate items on the receipt). item_list contains itemized receipt information.

Individual relations have the following description.

Customers

Id: unique identifier of the customer

LastName: last name of the customer

FirstName: first name of the customer

Products
Id : unique identifier of the baked product

Flavor: flavor/type of the product (e.g., "chocolate", "lemon")

Food: category of the product (e.g., "cake", "tart")

Price: price (in dollars)

Item_list

Receipt : receipt number (see reciepts.RecieptNumber)

Ordinal : position of the purchased item on the receipts. (i.e., first purchased item,second purchased item, etc...)

Item : identifier of the item purchased (see product.Id)

Receipts

ReceiptNumber : unique identifier of the receipt

Date : date of the purchase. The date is in DD-MM-YYY format,

CustomerId : id of the customer (see customers.Id) Summary of Bakery database:

CUSTOMERS (cid , fname, lname)

PRODUCTS (pid, flavor, food, price)

RECEIPTS (rno, rdate, cid)

ITEM_LIST (rno, ordinal, item) ●

Understand the database.

● Draw schema diagram for Bakery database.

● Create relations with appropriate data types and integrity constraints.

● Populate the database values using the Bakery.sql file.

STEP1:- Create Table Customers

```
mysql> create database bakery;
Query OK, 1 row affected (0.01 sec)

mysql> use bakery;
Database changed
mysql> create table customers (
    -> cid int unique,
    -> fname varchar(50),
    -> lname varchar(50)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc customers;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| cid   | int         | YES  | UNI | NULL    |       |
| fname | varchar(50) | YES  |     | NULL    |       |
| lname | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

STEP 2:- Create Table Products

```
mysql> create table products(
    -> pid int unique,
    -> flavor varchar(50),
    -> food varchar(50),
    -> price int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc products;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| pid    | int         | YES  | UNI | NULL    |       |
| flavor | varchar(50) | YES  |     | NULL    |       |
| food   | varchar(50) | YES  |     | NULL    |       |
| price  | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

STEP 3:- Create table Item_list

```
mysql> create table item_list(
    -> rno int,
    -> ordinal int,
    -> item int
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> desc item_list;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| rno     | int  | YES  |     | NULL    |       |
| ordinal | int  | YES  |     | NULL    |       |
| item    | int  | YES  |     | NULL    |       |
+---------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

STEP 4:- Create Table Receipts

```
mysql> create table receipts(
    -> rno int primary key,
    -> rdate date,
    -> cid int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc receipts;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| rno   | int  | NO   | PRI | NULL    |       |
| rdate | date | YES  |     | NULL    |       |
| cid   | int  | YES  |     | NULL    |       |
+-------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

STEP 5:- Now insert the values in the tables

```
mysql> insert into customers
    -> values(1,'Neeraj','Chormale'),
    -> (2,'Onkar','Yaglewad'),
    -> (3,'Atharva','Phadatare'),
    -> (4,'Manas','Shetty'),
    -> (5,'Kiran','Biradar'),
    -> (6,'Niraj','There');
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

```
mysql> insert into products
    -> values(1,'Chocolate','cake',20),
    -> (2,'Vanilla','ice cream',5),
    -> (3,'Strawberry','cheesecake',27.50),
    -> (4,'Blueberry','muffin',3),
    -> (5,'Caramel','popcorn',5.50);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> insert into item_list
    -> values(101,1,2),
    -> (102,2,4),
    -> (103,1,1),
    -> (104,2,3),
    -> (105,3,5);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> insert into receipts
    -> values (101, '2023-03-20', 1),
    -> (102, '2023-03-21', 2),
    -> (103, '2023-03-21', 3),
    -> (104, '2023-03-22', 4),
    -> (105, '2023-03-22', 5);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

STEP 6:- Perform these following commands in questions.

Exercise :

Write the following using Sub-query:

1. Create a view named Blue_Flavor, which display the product details (product id, food, price) of Blueberry flavor.

```
mysql> create view blue_flavor as
    -> select pid,food,price
    -> from products
    -> where flavor = 'Blueberry';
Query OK, 0 rows affected (0.01 sec)

mysql> select * from blue_flavor;
+------+--------+-------+
| pid  | food   | price |
+------+--------+-------+
|    4 | muffin |     3 |
+------+--------+-------+
1 row in set (0.01 sec)
```

2. Create a view named Cheap_Food, which display the details (product id, flavor, food, price) of products with price lesser than $1. Ensure that, the price of these food(s) should never rise above $1 through view.

```
mysql> create view cheap_food as
    -> select pid,flavor,food,
    ->     case
    ->         when price < 1 then 1
    ->         else price
    ->     end as price
    -> from products
    -> where price < 1;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cheap_food;
Empty set (0.00 sec)
```

3. Create a view called Hot_Food that show the product id and its quantity where the same product is ordered more than once in the same receipt.

```
mysql> create view hot_food as
    -> select item as pid,COUNT(*) as quantity
    -> from item_list
    -> group by rno, item
    -> having COUNT(*) > 1;
Query OK, 0 rows affected (0.01 sec)
```

4. Create a view named Pie_Food that will display the details (customer lname, flavor, receipt number and date, ordinal) who had ordered the Pie food with receipt details.

```
mysql> create view pie_food as
    -> select c.lname,p.flavor, r.rno, r.rdate, i.ordinal
    -> from customers c
    ->  join receipts r on c.cid = r.cid
    -> join item_list i on r.rno = i.rno
    -> join products p on i.item = p.pid
    -> where p.food = 'pie';
Query OK, 0 rows affected (0.01 sec)
```

5. Create a view Cheap_View from Cheap_Food that shows only the product id, flavor and food.

```
mysql> create view cheap_view as
    -> select pid,flavor,food
    -> from cheap_food;
Query OK, 0 rows affected (0.01 sec)
```

Conclusion:- We have solved all the following commands .