# Assignment - 6
## Topic: Advanced SQL Queries
## Course: DBMS Lab [MA39204]

**Department of Mathematics**
**Indian Institute of Technology Kharagpur**

Time: **2 hours 45 mins** Date: **08-Feb-2023**

==================================================================================

General Instructions: Do NOT use mobile phones during Lab hours. If anyone indulges in malpractice, his/her marks will be deducted.

Submit the assignment to the email-id: dbmslabmathiitkgp@gmail.com
Subject of the email should be: rollNumber_dd_mm
Send a text file which should contain the mysql inputs and outputs. Name of that file should be: rollNumber_dd_mm.txt

---

- ORDER BY - used to sort the result-set in ascending or descending order.
- MIN() - returns the smallest value of the selected column.
- MAX() - returns the largest value of the selected column.
- COUNT()- returns the number of rows that matches a specified criterion.
- AVG() - returns the average value of a numeric column.
- LIKE - used in a WHERE clause to search for a specified pattern in a column.
- The percent sign (%) represents zero, one, or multiple characters.
- The underscore sign (_) represents one, single character.
- GROUP BY - statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
- SELECT DISTINCT - used to return only distinct (different) values.
- AS - used to give a table, or a column in a table, a temporary name.
- BETWEEN - operator selects values within a given range.
- JOIN - used to combine rows from two or more tables, based on a related column between them.

---

1. Write SQL statements to generate the following tables for User, Product, Offer, Brand, Purchase and Product has.

***User***

| Field | Type | Key |
|---|---|---|
| User_ID | decimal(3,0) | PRI |
| Name | varchar(30) | |
| City | varchar(30) | |
| Email_ID | varchar(30) | |
| DOB | date | |

***Product***

| Field | Type | Key |
|---|---|---|
| Prod_ID | decimal(5,0) | PRI |
| Prod_Name | varchar(30) | |
| Description | varchar(100) | |
| Price | decimal(3,0) | |
| Ratings | decimal(3,0) | |
| Brand_ID | decimal(3,0) | |

***Offer***

| Field | Type | Key |
|---|---|---|
| Offer_ID | decimal(3,0) | PRI |

***Brand***

| Field | Type | Key |
|---|---|---|
| Brand_ID | decimal(3,0) | PRI |

|  |  |  |
|---|---|---|
| Offer_Des | varchar(30) |  |
| Validity | varchar(30) |  |

|  |  |  |
|---|---|---|
| Brand_Name | varchar(30) |  |
| Brand_Type | varchar(30) |  |
| Rank | decimal(3,0) |  |

***Purchase***

| **Field** | **Type** | **Key** |
|---|---|---|
| User_ID | decimal(3,0) |  |
| Prod_ID | decimal(5,0) |  |

***Product Has***

| **Field** | **Type** | **Key** |
|---|---|---|
| Prod_ID | decimal(5,0) |  |
| Brand_ID | decimal(3,0) |  |
| Offer_ID | decimal(3,0) |  |

2. INSERT the following data into the tables

***User***

| User_ID | Name | City | Email_ID | DOB |
|---|---|---|---|---|
| 50 | Bishal Nayek | Bhubaneswar | bn30@email.com | 2001-05-30 |
| 39 | Vaishali | Hyderabad | v23@email.com | 1997-12-23 |
| 42 | Varun Sharma | Mumbai | vs01@email.com | 1985-08-01 |
| 98 | Abhijit Sen | Kolkata | as05@email.com | 1993-07-05 |
| 09 | Praveen Kumar | Delhi | pk09@email.com | 1999-11-09 |

***Product***

| Prod_ID | Prod_Name | Description | Price | Ratings | Brand_ID |
|---|---|---|---|---|---|
| 0325 | Tablet_1 | Big Display | 49990 | 4.3 | 05 |
| 9826 | Smart Tv_New | Dolby Sound | 29999 | 4.1 | 09 |
| 3518 | Face Combo | All Skin Type | 1549 | 3.9 | 10 |
| 1437 | ASOIAF_GOT | Paperback | 3110 | 4.5 | 01 |
| 2549 | Men's jeans | Slimfit | 2120 | 3.7 | 03 |

***Purchase***

| User_ID | Prod_ID |
|---|---|
| 42 | 0325 |

***Offer***

| Offer_ID | Offer_Des | Validity |
|---|---|---|
| 05 | 10% discount on ABI credit card | 30 days |

| | |
|---|---|
| 50 | 1437 |
| 39 | 3518 |
| 98 | 0325 |
| 09 | 9826 |
| 39 | 2549 |

| | | |
|---|---|---|
| 20 | No Cost EMI | 15 days |
| 16 | 5% Cashback | Unlimited |
| 29 | Exchange Old product | limited |
| 18 | RS 1500 off over RS 20000 purchase | 45 days |

***Brand***

| Brand_ID | Brand_Name | Brand_Type | Rank |
|---|---|---|---|
| 01 | Harper Collins | Publisher | 6 |
| 05 | Apple | Electronics | 7 |
| 09 | Samsung | Electronics | 4 |
| 03 | Mcaffine | Skin Care | 4 |
| 10 | Peter England | Clothing | 8 |

***Product Has***

| Prod_ID | Brand_ID | Offer_ID |
|---|---|---|
| 0325 | 05 | 29 |
| 9826 | 09 | 18 |
| 3518 | 10 | 16 |
| 1437 | 01 | 20 |
| 2549 | 03 | 16 |

3. Find the users whose name starts with ‘v’.
4. Find the Brand Name and their rank in their field whose Brand Type contains ‘c’ or ‘C’.
5. Find the Brand_ID’s which are greater than any Offer_ID.
6. Find the number of products, where product price is under 10000.
7. Select the top 50% of the offer table.
8. Categorize products as price categories according to their price (price> 30000 is high, 10000< price <30000 is medium and price< 10000 is low) from the product table and select the Prod_ID and Prod_Name column together with the price category. (Use Case)
9. Find the Standard deviation of price and rating from the Product table.
10. Extract the month of birth of the user, whose city is “Kolkata”.
11. Write SQL Code to find the Natural Join of the Product and Brand table. Find the same using Inner Join.
12. Using Natural Join, find the Description of the products having the same Brand Type and at the same time rating greater than 4.
13. Find the product name, purchased by the user whose day of birth is between 4 to 24 using Natural Join.
14. Find the name of the users who are from the same city.
15. Find the product name which is bought by multiple users.
16. Find the Email_ID’s of the users with the most frequent buys.
17. Find the User’s name who got 5% cashback along with the product name.
18. Print the name of the electronic brand for which the offer is available.
19. Print the name and email ID of the users from ‘Kolkata’ who bought ‘Tablet_1’.
20. Print the details of the user’s who are older than 21 years and have bought ‘Face Combo’.