DBMS - LAB 3

Aggregate functions and Join expressions
Art Gallery Management (AGM)

NAME :GAURAV B V

SECTION:C

SRN:PES2UG21CS175

INSTRUCTIONS

- In Lab 3, the students are expected to solve **three** tasks that are to be completed and submitted.
- For this Lab, the students are provided with an SQL file. This file is required for tasks 2, 3.The students are required to run the sql file and then execute these (task 2,3) tasks. (Not needed for task 1)
- As a part of LAB 3, there are three tasks that are to be completed as described below:
- **TASK 1:** There is a small case study on the Café database that has been given. You must understand the case study, the set of records in the tables, then you are supposed to answer, whether the task1 commands given are executable or not. Also justify the same with appropriate reasons. (NO NEED FOR ANY EXECUTION)
- **TASK 2:** A relational algebraic expression is given. You are expected to convert the relational algebra query into an SQL query, and then execute it. Elaborate the findings about the query execution.
- **Task 3:** This is the last task. Here, there will be three questions and you are expected to understand the questions write the query and execute the same.

  Support each question with the corresponding screenshot.

- The screenshots that are to be taken for each task are specified in detail below "EXAMPLE".
- As a part of the submission process, the following are to be submitted:
- A **PDF** document, containing all the Screenshots for all 3 tasks as suggested
- Name of the file: `<your SRN>_Art_Gallery_DB_Lab3.pdf`

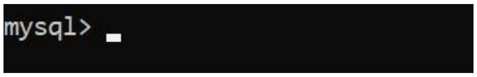
Example:

Refer to the sample submissions given below. This will give you an idea about the details that must be included in your submissions

NOTE: Screenshots must be taken from "**Command Line".**

Changing your command line prompt:

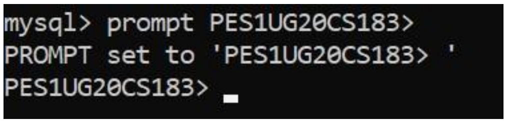
Before :



```
mysql> _
```

prompt PES1UG20CS183>

After:

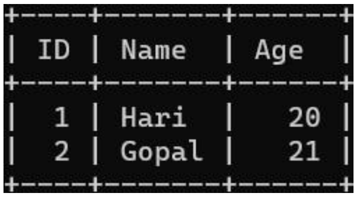


```
mysql> prompt PES1UG20CS183>
PROMPT set to 'PES1UG20CS183> '
PES1UG20CS183> _
```

Task 1:

No screenshot is required as you are not going to execute anything. Only reasoning of whether the query would get executed or not is supposed to be given as shown below:

Sample_lab3 (ID, Name, Age)



```
+----+-------+------+
| ID | Name  | Age  |
+----+-------+------+
|  1 | Hari  |   20 |
|  2 | Gopal |   21 |
+----+-------+------+
```

Question: can the following statement be executed?

- SELECT * FROM Sample_lab3 WHERE Age > 20; **Expected answer:**

The above query can be executed without any error. The result of this execution would be a table as shown:

ID	Name	Age

2	Gopal	21

NOTE: you have to draw or make the table DO NOT execute it and take ss for task 1

Task 2:

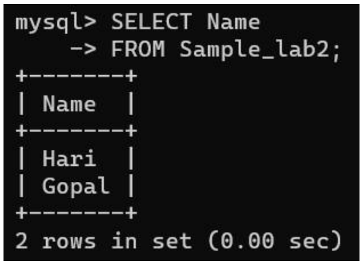
Convert the following algebraic expression to an SQL query and find out what it does.

- π_{name} (Sample_lab3) **Expected answer:**

Converted SQL Query:

- SELECT Name FROM Sample_lab3;

O/P screenshot:



```
mysql> SELECT Name
    -> FROM Sample_lab2;
+-------+
| Name  |
+-------+
| Hari  |
| Gopal |
+-------+
2 rows in set (0.00 sec)
```

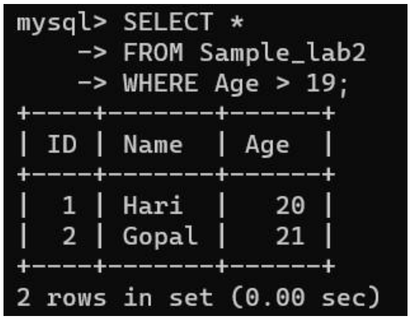
The given algebraic expression displays the NAME attribute in the Sample_lab3 table.

TASK 3:

- Find all the people whose age is greater than 19.

Expected answer:

- SELECT *
  FROM Sample_lab3
  WHERE Age > 19;



```
mysql> SELECT *
    -> FROM Sample_lab2
    -> WHERE Age > 19;
+----+-------+------+
| ID | Name  | Age  |
+----+-------+------+
|  1 | Hari  |   20 |
|  2 | Gopal |   21 |
+----+-------+------+
2 rows in set (0.00 sec)
```

Lab 3 Exercises

Task 1: Consider this schema

Person (Name, Age, Gender)

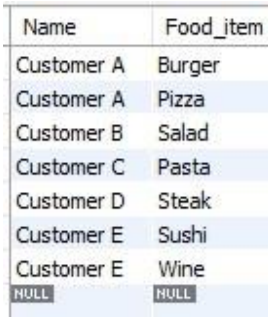
Serves (Café_name, Food_item, price)

Visits (Name, Café_name)

Orders (Name, Food_item)

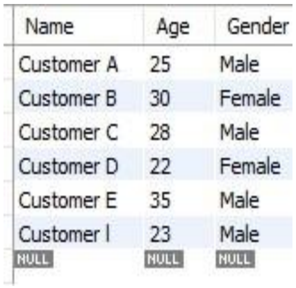
- The "Person" table stores information about customers who visit the cafes.
- The "Serves" table contains information about the food items served at different cafes.
- The "Visits" table keeps track of which customers frequently visit which cafes.
- The "Orders" table records the food items ordered by customers.
- All the attributes having the same names represent the same type of data (ex: café_name is the same for all serves, visits table and there is a referencing relation between visits and serves)

Orders:



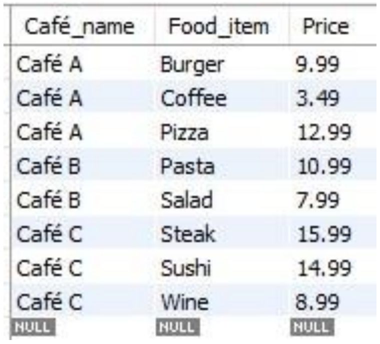
Name	Food_item
Customer A	Burger
Customer A	Pizza
Customer B	Salad
Customer C	Pasta
Customer D	Steak
Customer E	Sushi
Customer E	Wine
NULL	NULL

Person:



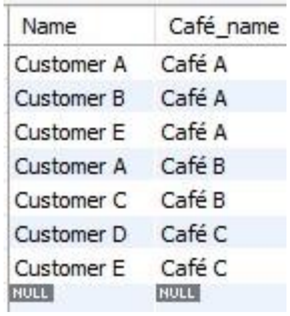
Name	Age	Gender
Customer A	25	Male
Customer B	30	Female
Customer C	28	Male
Customer D	22	Female
Customer E	35	Male
Customer I	23	Male
NULL	NULL	NULL

Serves:



Café_name	Food_item	Price
Café A	Burger	9.99
Café A	Coffee	3.49
Café A	Pizza	12.99
Café B	Pasta	10.99
Café B	Salad	7.99
Café C	Steak	15.99
Café C	Sushi	14.99
Café C	Wine	8.99
NULL	NULL	NULL

visits:



Name	Café_name
Customer A	Café A
Customer B	Café A
Customer E	Café A
Customer A	Café B
Customer C	Café B
Customer D	Café C
Customer E	Café C
NULL	NULL

1) SELECT S.Café_name, AVG(S.price) AS AveragePrice

FROM Serves S group by price;

No this query can be executed cause v cant group with price.

2) SELECT V1.Name

```
FROM Visits V1 JOIN Visits V2 ON V1.Name = V2.Name
WHERE V1.café_name = 'Café A' AND V2.café_name='Café B';
```

YES it can be executed without any error

This results in a table which contains the people who visited both café A and café B

Which is

Name	
Customer A	

3)

```
SELECT P.Name, SUM(S.price) AS TotalPrice
FROM person p JOIN orders O ON P.Name =O.Name
JOIN Serves S ON O. Food_item = S.Food_item;
```

This statement gives an error cause v are not using "group by" clause.

4)

```
SELECT V.Name
FROM ( SELECT Name, COUNT(DISTINCT café_name) AS cafecount
FROM Visits GROUP BY Name)V WHERE V.cafecount >=2;
```

Can be executer without error

Name
Customer A
Customer E

5) SELECT DISTINCT P.Name

FROM Person P INNER JOIN Visits USING (Name);

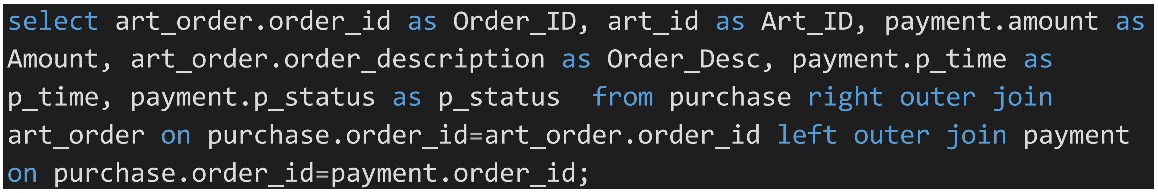
Yes can be executed without errors

Name
Customer A
Customer B
Customer C
Customer D
Customer E
Customer I

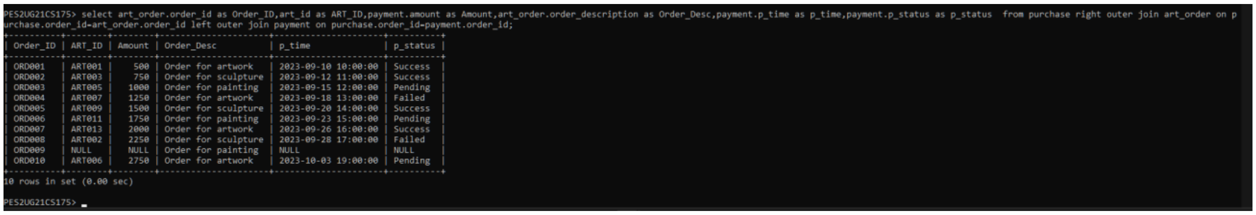
Task 2:

Convert 6th and 7th question relational algebra into SQL query execute it and attach a screen .

6) $\square_{(Order\_ID,Art\_ID,Amount,Order\_Desc,Payment\_time,Payment\_Status)}$(□ order_id, art_id, amount, order_description, p_time, p_status ((p(purchase) $⟖_{order\_id=order\_id}$ art_order) $⟗_{order\_id=order\_id}$ py (payment)))



```
select art_order.order_id as Order_ID, art_id as Art_ID, payment.amount as
Amount, art_order.order_description as Order_Desc, payment.p_time as
p_time, payment.p_status as p_status  from purchase right outer join
art_order on purchase.order_id=art_order.order_id left outer join payment
on purchase.order_id=payment.order_id;
```

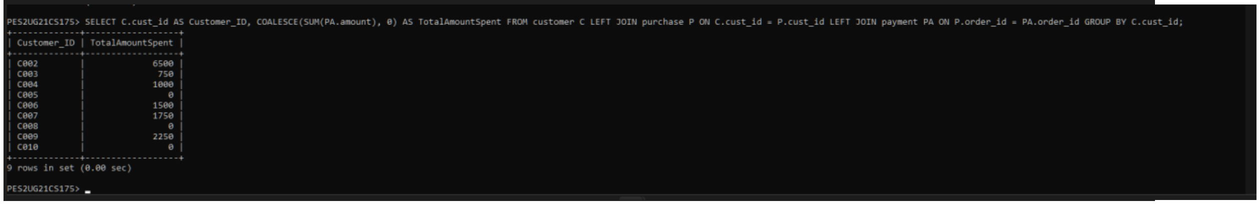


```
PES2UG21CS175> select art_order.order_id as Order_ID,art_id as ART_ID,payment.amount as Amount,art_order.order_description as Order_Desc,payment.p_time as p_time,payment.p_status as p_status  from purchase right outer join art_order on p
urchase.order_id=art_order.order_id left outer join payment on purchase.order_id=payment.order_id;
+----------+--------+--------+---------------------+---------------------+----------+
| Order_ID | ART_ID | Amount | Order_Desc          | p_time              | p_status |
+----------+--------+--------+---------------------+---------------------+----------+
| ORD001   | ART001 |    500 | Order for artwork   | 2023-09-10 10:00:00 | Success  |
| ORD002   | ART003 |    750 | Order for sculpture | 2023-09-12 11:00:00 | Success  |
| ORD003   | ART005 |   1000 | Order for painting  | 2023-09-15 12:00:00 | Pending  |
| ORD004   | ART007 |   1250 | Order for artwork   | 2023-09-18 13:00:00 | Failed   |
| ORD005   | ART009 |   1500 | Order for sculpture | 2023-09-20 14:00:00 | Success  |
| ORD006   | ART011 |   1750 | Order for painting  | 2023-09-23 15:00:00 | Pending  |
| ORD007   | ART013 |   2000 | Order for artwork   | 2023-09-26 16:00:00 | Success  |
| ORD008   | ART002 |   2250 | Order for sculpture | 2023-09-28 17:00:00 | Failed   |
| ORD009   | NULL   |   NULL | Order for painting  | NULL                | NULL     |
| ORD010   | ART006 |   2750 | Order for artwork   | 2023-10-03 19:00:00 | Pending  |
+----------+--------+--------+---------------------+---------------------+----------+
10 rows in set (0.00 sec)

PES2UG21CS175> _
```

7) ρ R(Customer_ID, Customer_Income)(c.cust _id ℑ (p.cust_id, sum(amount)) (p (purchase)⋈order_id=order_id pa (payment)))

```
SELECT C.cust_id AS Customer_ID, COALESCE(SUM(PA.amount), 0) AS
TotalAmountSpent FROM customer C LEFT JOIN purchase P ON C.cust_id =
P.cust_id LEFT JOIN payment PA ON P.order_id = PA.order_id GROUP BY
C.cust_id;
```



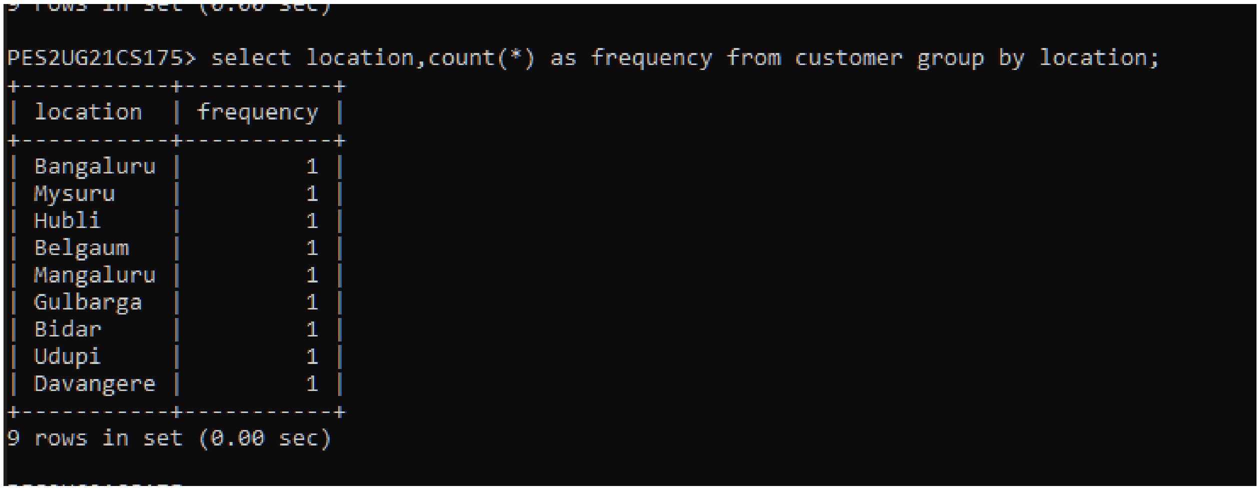
```
PES2UG21CS175> SELECT C.cust_id AS Customer_ID, COALESCE(SUM(PA.amount), 0) AS TotalAmountSpent FROM customer C LEFT JOIN purchase P ON C.cust_id = P.cust_id LEFT JOIN payment PA ON P.order_id = PA.order_id GROUP BY C.cust_id;
+-------------+------------------+
| Customer_ID | TotalAmountSpent |
+-------------+------------------+
| C002        |             6500 |
| C003        |              750 |
| C004        |             1000 |
| C005        |                0 |
| C006        |             1500 |
| C007        |             1750 |
| C008        |                0 |
| C009        |             2250 |
| C010        |                0 |
+-------------+------------------+
9 rows in set (0.00 sec)

PES2UG21CS175> _
```

Task 3:

8) Tally the customer count in each location. This data assists in understanding the distribution of customers across various geographical areas, allowing the art gallery to tailor its services and promotions to cater effectively to different regions and demographics.

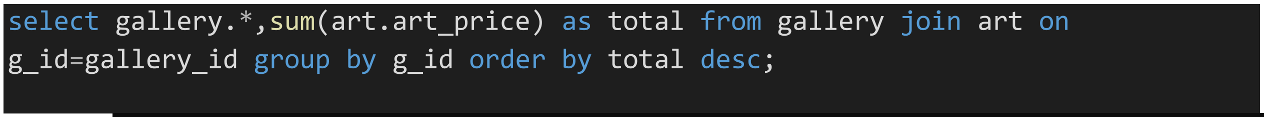
```
select location,count(*) as frequency from customer group by location;
```



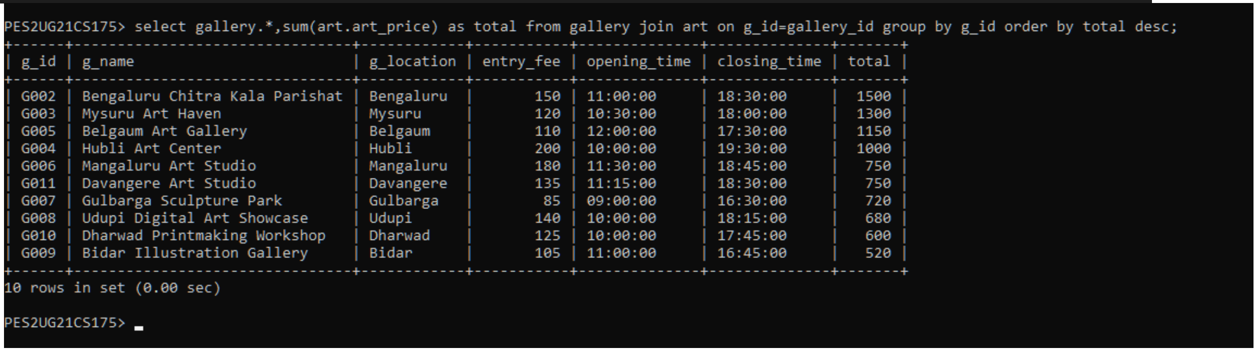
```
9 rows in set (0.00 sec)

PES2UG21CS175> select location,count(*) as frequency from customer group by location;
+-----------+-----------+
| location  | frequency |
+-----------+-----------+
| Bangaluru |         1 |
| Mysuru    |         1 |
| Hubli     |         1 |
| Belgaum   |         1 |
| Mangaluru |         1 |
| Gulbarga  |         1 |
| Bidar     |         1 |
| Udupi     |         1 |
| Davangere |         1 |
+-----------+-----------+
9 rows in set (0.00 sec)
```

9) In the world of art galleries, understanding the financial performance of each gallery is paramount. To achieve this, we embark on a quest to unveil the total revenue generated by each gallery, shedding light on their respective contributions to the gallery's economy .list should be sorted based on decreasing order of revenue .



```
select gallery.*,sum(art.art_price) as total from gallery join art on
g_id=gallery_id group by g_id order by total desc;
```



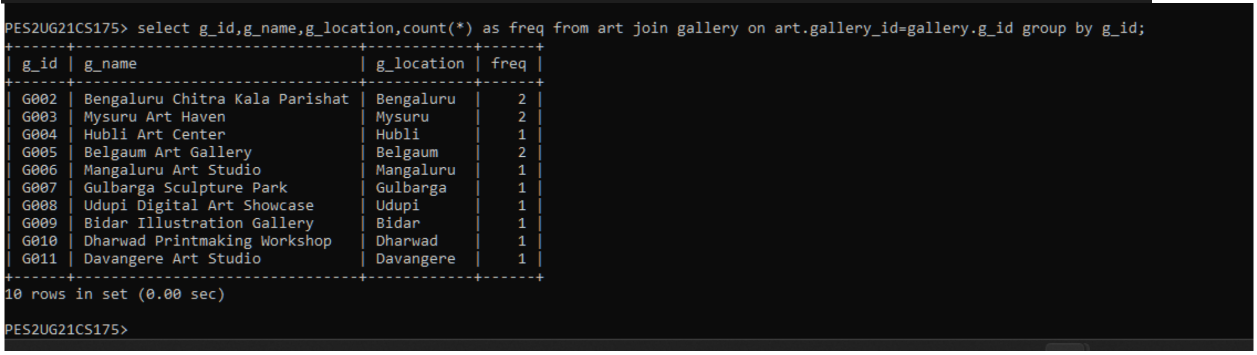
```
PES2UG21CS175> select gallery.*,sum(art.art_price) as total from gallery join art on g_id=gallery_id group by g_id order by total desc;
+------+-------------------------------+------------+-----------+--------------+--------------+-------+
| g_id | g_name                        | g_location | entry_fee | opening_time | closing_time | total |
+------+-------------------------------+------------+-----------+--------------+--------------+-------+
| G002 | Bengaluru Chitra Kala Parishat | Bengaluru  |       150 | 11:00:00     | 18:30:00     |  1500 |
| G003 | Mysuru Art Haven              | Mysuru     |       120 | 10:30:00     | 18:00:00     |  1300 |
| G005 | Belgaum Art Gallery           | Belgaum    |       110 | 12:00:00     | 17:30:00     |  1150 |
| G004 | Hubli Art Center              | Hubli      |       200 | 10:00:00     | 19:30:00     |  1000 |
| G006 | Mangaluru Art Studio          | Mangaluru  |       180 | 11:30:00     | 18:45:00     |   750 |
| G011 | Davangere Art Studio          | Davangere  |       135 | 11:15:00     | 18:30:00     |   750 |
| G007 | Gulbarga Sculpture Park       | Gulbarga   |        85 | 09:00:00     | 16:30:00     |   720 |
| G008 | Udupi Digital Art Showcase    | Udupi      |       140 | 10:00:00     | 18:15:00     |   680 |
| G010 | Dharwad Printmaking Workshop  | Dharwad    |       125 | 10:00:00     | 17:45:00     |   600 |
| G009 | Bidar Illustration Gallery    | Bidar      |       105 | 11:00:00     | 16:45:00     |   520 |
+------+-------------------------------+------------+-----------+--------------+--------------+-------+
10 rows in set (0.00 sec)

PES2UG21CS175> _
```

10) Retrieve galleries that exhibit over 1 art works. This information aids in identifying galleries with a substantial collection, making it easier for

visitors to explore diverse art pieces and enhancing the gallery's reputation as a vibrant artistic hub.

```
select g_id,g_name,g_location,count(*) as freq from art join gallery on
art.gallery_id=gallery.g_id group by g_id;
```

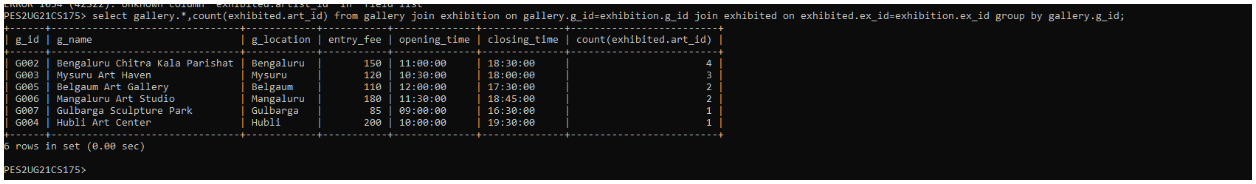


```
PES2UG21CS175> select g_id,g_name,g_location,count(*) as freq from art join gallery on art.gallery_id=gallery.g_id group by g_id;
+------+-------------------------------+-----------+------+
| g_id | g_name                        | g_location | freq |
+------+-------------------------------+-----------+------+
| G002 | Bengaluru Chitra Kala Parishat | Bengaluru  |    2 |
| G003 | Mysuru Art Haven              | Mysuru     |    2 |
| G004 | Hubli Art Center              | Hubli      |    1 |
| G005 | Belgaum Art Gallery           | Belgaum    |    2 |
| G006 | Mangaluru Art Studio          | Mangaluru  |    1 |
| G007 | Gulbarga Sculpture Park       | Gulbarga   |    1 |
| G008 | Udupi Digital Art Showcase    | Udupi      |    1 |
| G009 | Bidar Illustration Gallery    | Bidar      |    1 |
| G010 | Dharwad Printmaking Workshop  | Dharwad    |    1 |
| G011 | Davangere Art Studio          | Davangere  |    1 |
+------+-------------------------------+-----------+------+
10 rows in set (0.00 sec)

PES2UG21CS175>
```

11) In the bustling world of art galleries, it's essential to recognize exhibitions that truly stand out those that feature the most artworks, showcasing the pinnacle of artistic expression. Imagine you're the curator of a renowned gallery, and you want to spotlight the exhibition that has exhibited the highest number of artworks. By doing so, you can celebrate the artists, generate buzz around your gallery, and provide visitors with a remarkable experience.

```
select gallery.*,count(exhibited.art_id) from gallery join exhibition on
gallery.g_id=exhibition.g_id join exhibited on exhibited.ex_id=exhibition.ex_id
group by gallery.g_id;
```



```
ERROR 1054 (42S22): Unknown column 'exhibited.artist_id' in 'field list'
PES2UG21CS175> select gallery.*,count(exhibited.art_id) from gallery join exhibition on gallery.g_id=exhibition.g_id join exhibited on exhibited.ex_id=exhibition.ex_id group by gallery.g_id;
+------+--------------------------------+------------+-----------+--------------+--------------+-------------------------+
| g_id | g_name                         | g_location | entry_fee | opening_time | closing_time | count(exhibited.art_id) |
+------+--------------------------------+------------+-----------+--------------+--------------+-------------------------+
| G002 | Bengaluru Chitra Kala Parishat | Bengaluru  |       150 | 11:00:00     | 18:30:00     |                       4 |
| G003 | Mysuru Art Haven               | Mysuru     |       120 | 10:30:00     | 18:00:00     |                       3 |
| G005 | Belgaum Art Gallery            | Belgaum    |       110 | 12:00:00     | 17:30:00     |                       2 |
| G006 | Mangaluru Art Studio           | Mangaluru  |       180 | 11:30:00     | 18:45:00     |                       2 |
| G007 | Gulbarga Sculpture Park        | Gulbarga   |        85 | 09:00:00     | 16:30:00     |                       1 |
| G004 | Hubli Art Center               | Hubli      |       200 | 10:00:00     | 19:30:00     |                       1 |
+------+--------------------------------+------------+-----------+--------------+--------------+-------------------------+
6 rows in set (0.00 sec)

PES2UG21CS175>
```

Lab3 exercise is concluded