Name : Bhavani Rajpurohit

Class : AIA-3 Subject: DBMS LAB

Roll No : 2213688 Batch : B

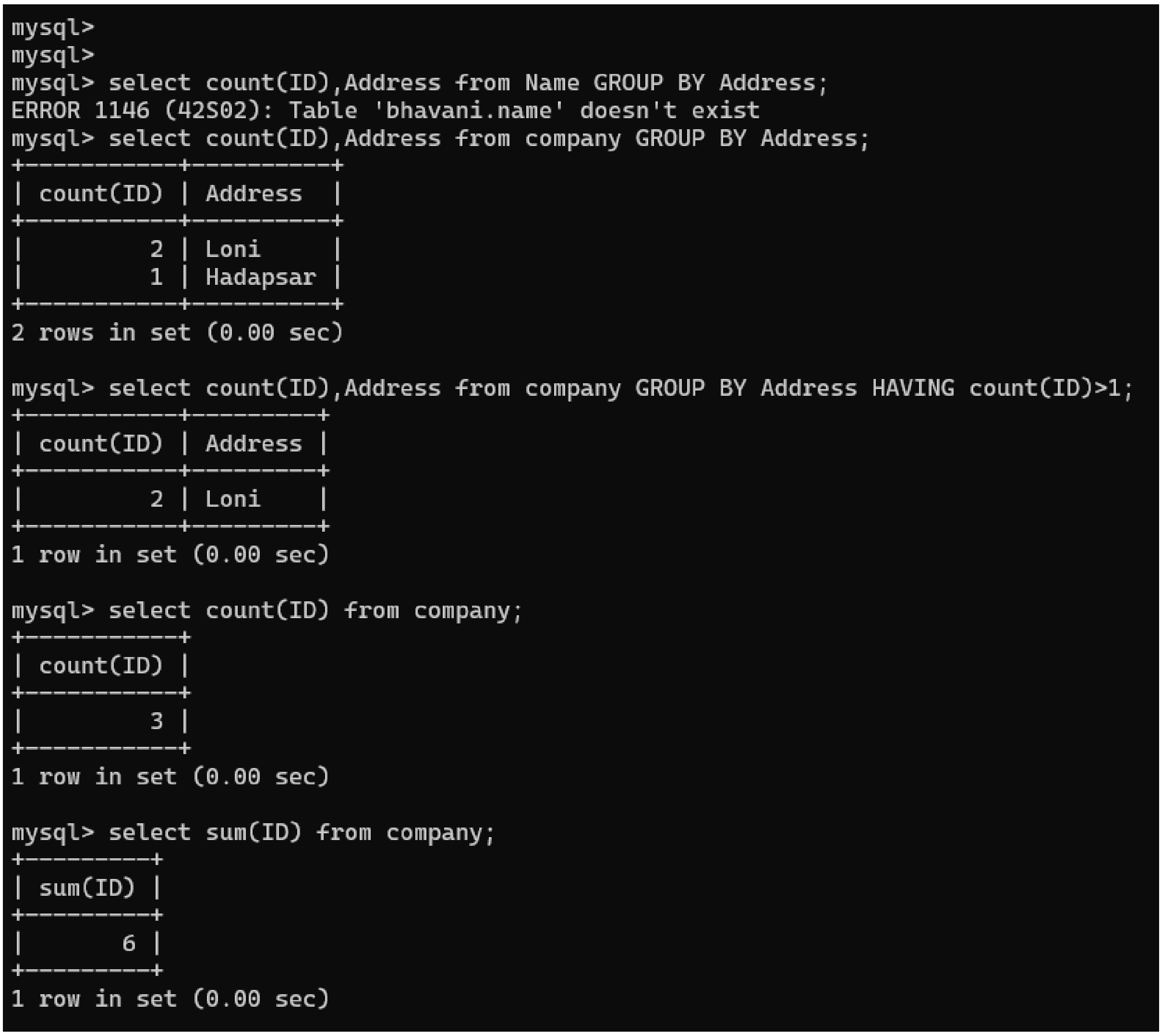
ASSIGNMENT NO: 04

Aim: Database Querying – Simple queries, Queries using aggregate functions, GROUP BY and HAVING clauses.

1) Write a Group-by query for one/two columns in Manufacturing industry /Hospital/ Company table
2) Write a Having clause query for Manufacturing industry / Hospital/ Company table
3) Write a queries to make use of aggregate functions Count(),Sum(),Avg(),Min(),Max()

Software required: MySQL

CLI Screenshots:

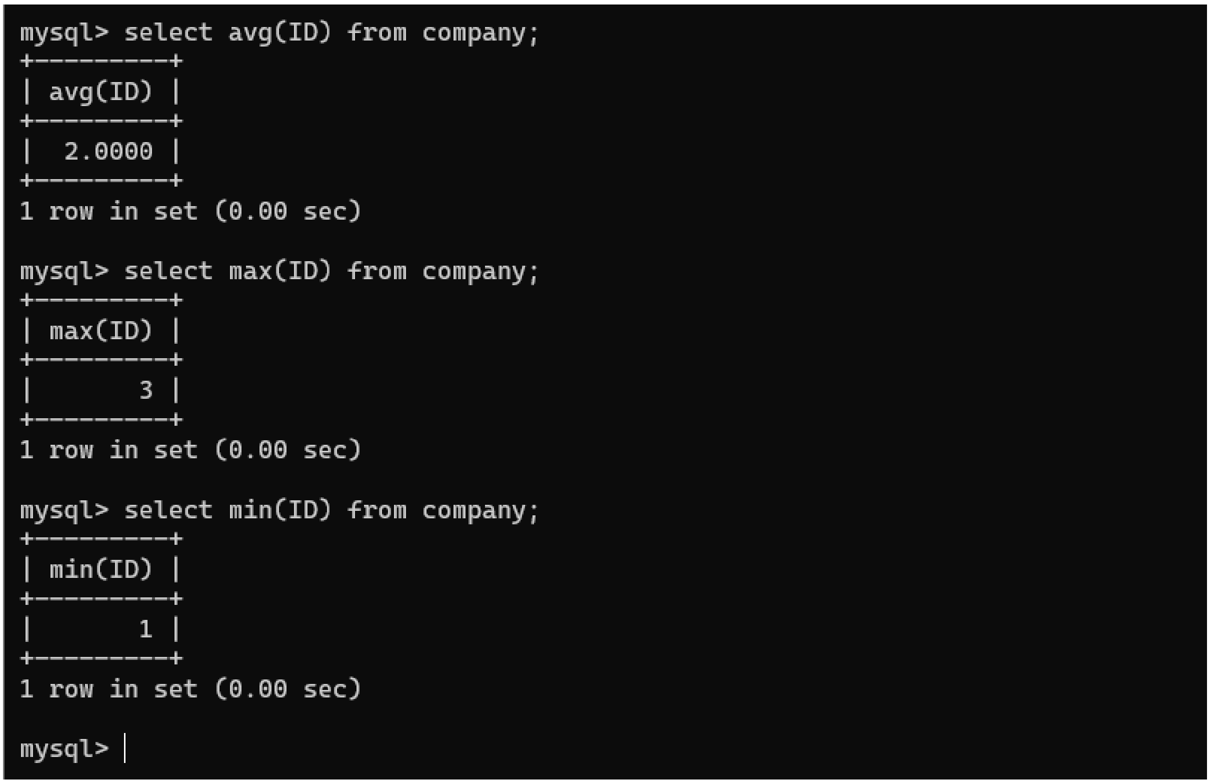


```
mysql>
mysql>
mysql> select count(ID),Address from Name GROUP BY Address;
ERROR 1146 (42S02): Table 'bhavani.name' doesn't exist
mysql> select count(ID),Address from company GROUP BY Address;
+-----------+----------+
| count(ID) | Address  |
+-----------+----------+
|         2 | Loni     |
|         1 | Hadapsar |
+-----------+----------+
2 rows in set (0.00 sec)

mysql> select count(ID),Address from company GROUP BY Address HAVING count(ID)>1;
+-----------+---------+
| count(ID) | Address |
+-----------+---------+
|         2 | Loni    |
+-----------+---------+
1 row in set (0.00 sec)

mysql> select count(ID) from company;
+-----------+
| count(ID) |
+-----------+
|         3 |
+-----------+
1 row in set (0.00 sec)

mysql> select sum(ID) from company;
+---------+
| sum(ID) |
+---------+
|       6 |
+---------+
1 row in set (0.00 sec)
```



```
mysql> select avg(ID) from company;
+---------+
| avg(ID) |
+---------+
|  2.0000 |
+---------+
1 row in set (0.00 sec)

mysql> select max(ID) from company;
+---------+
| max(ID) |
+---------+
|       3 |
+---------+
1 row in set (0.00 sec)

mysql> select min(ID) from company;
+---------+
| min(ID) |
+---------+
|       1 |
+---------+
1 row in set (0.00 sec)

mysql> |
```

FAQs:

I. What is the difference between the WHERE clause and the HAVING clause?

Ans) WHERE clause is used to filter individual rows based on column values before grouping or aggregating. HAVING clause is used to filter groups of rows based on aggregate function results after grouping and aggregating. WHERE is used with non-aggregated columns, while HAVING is used with aggregated columns. WHERE is applied before GROUP BY, and HAVING is applied after GROUP BY.

II. Can I use the GROUP BY clause without any aggregate functions?

Ans) Yes, you can use the GROUP BY clause without any aggregate functions. It will group the rows based on the specified column(s) and provide a list of each unique group in the result set without performing any calculations on the grouped data.

III. Can I use the HAVING clause without the GROUP BY clause?

Ans) No, you cannot use the HAVING clause without the GROUP BY clause. The HAVING clause is used specifically in conjunction with the GROUP BY clause to filter groups of rows based on aggregate function results. It allows you to apply conditions to the grouped data after aggregations have been performed. Without the GROUP BY clause, there are no groups to filter, and the HAVING clause would have no context or meaning.

IV. Can I include columns in the SELECT statement that are not part of the GROUP BY clause?

Ans) Yes, you can include columns in the SELECT statement that are not part of the GROUP BY clause. When using the GROUP BY clause, you can include additional columns in the SELECT statement that are not part of the grouping criteria or aggregate functions.

IV. Can I use multiple aggregate functions in the HAVING clause?

Ans) Yes, you can use multiple aggregate functions in the HAVING clause. The HAVING clause allows you to filter groups of rows based on the results of aggregate functions. You can use multiple aggregate functions in the HAVING clause to apply conditions on different calculations performed on the grouped data.

Conclusion:

In summary, the GROUP BY clause helps to group rows based on specific columns, while the HAVING clause allows you to filter the grouped data based on aggregate conditions. Together, they provide powerful tools for data analysis and summarization in DBMS.

Additional problem statements:

I. Write an SQL query to analyze sales data by grouping it by product category and calculate the total revenue for each category. Display only the categories with total revenue greater than $10,000.

Ans) SELECT category, SUM(sales_amount) AS total_revenue FROM sales GROUP BY category HAVING SUM(sales_amount) > 10000;

II. Develop an SQL query to retrieve the department names and the count of employees in each department. Display only the departments that have more than 50 employees.

Ans) SELECT department_name, COUNT(*) AS employee_count FROM employees GROUP BY department_name HAVING COUNT(*) > 50;

III. Create an SQL query to group the products by their suppliers and calculate the average stock quantity for each supplier. Display only the suppliers whose average stock quantity exceeds 500 units.

Ans) SELECT supplier_id, AVG(stock_quantity) AS average_stock_quantity FROM products GROUP BY supplier_id HAVING AVG(stock_quantity) > 500;

IV. Write an SQL query to group customers by their age range (e.g., 18-25, 26-35, etc.) and calculate the count of customers in each age range. Display only the age ranges with more than 100 customers.

```
Ans) SELECT CASE
            WHEN age BETWEEN 18 AND 25 THEN '18-25'
            WHEN age BETWEEN 26 AND 35 THEN '26-35'
            WHEN age BETWEEN 36 AND 45 THEN '36-45'
            ELSE '46+' -- Assuming anyone above 45 falls into the '46+' age range
          END AS age_range,
            COUNT(*) AS customer_count
FROM customers
GROUP BY age_range
HAVING COUNT(*) > 100;
```

V. Develop an SQL query to analyze exam scores by grouping them by the subject and calculate the average score for each subject. Display only the subjects with an average score above 80.

Ans) SELECT subject, AVG(score) AS average_score FROM exam_scores GROUP BY subject HAVING AVG(score) > 80;