Name: Vishal Shashikant Salvi.

UID: 2019230069

Class: SE Comps

Batch: C

Experiment No 5

Aim: Practicing Having Group-by Clause

Theory:

Group By & Having Clause

The GROUP BY clause is a SQL command that is used to group rows that have the same values.

The GROUP BY clause is used in the SELECT statement. Optionally it is used in conjunction with aggregate functions to produce summary reports from the database.

That's what it does, summarizing data from the database.

The queries that contain the GROUP BY clause are called grouped queries and only return a single row for every grouped item.

Syntax:

```
SELECT statements...
GROUP BY column_name1[, column_name2,...]
[HAVING condition];
```

HERE

- "SELECT statements..." is the standard SQL SELECT command query.
- "GROUP BY *column_name1*" is the clause that performs the grouping based on column_name1.
- "[, column_name2,...]" is optional; represents other column names when the grouping is done on more than one column.
- "[HAVING condition]" is optional; it is used to restrict the rows affected by the GROUP BY clause. It is similar to the WHERE clause.

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax

The basic syntax of a GROUP BY clause is shown in the following code block. The GROUP BY clause must follow the conditions in the WHERE clause and must precede the ORDER BY clause if one is used.

```
SELECT column1, column2
FROM table_name
WHERE [ conditions ]
GROUP BY column1, column2
ORDER BY column1, column2
```

Example

Consider the CUSTOMERS table is having the following records −

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

If you want to know the total amount of the salary on each customer, then the GROUP BY query would be as follows.

```
SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS
  GROUP BY NAME;
```

This would produce the following result −

```
+----------+-------------+
| NAME     | SUM(SALARY) |
+----------+-------------+
| Chaitali |     6500.00 |
| Hardik   |     8500.00 |
| kaushik  |     2000.00 |
| Khilan   |     1500.00 |
| Komal    |     4500.00 |
| Muffy    |    10000.00 |
| Ramesh   |     2000.00 |
+----------+-------------+
```

SQL HAVING Clause

- HAVING filters records that work on summarized GROUP BY results.
- HAVING applies to summarized group records, whereas WHERE applies to individual records.
- Only the groups that meet the HAVING criteria will be returned.
- HAVING requires that a GROUP BY clause is present.
- WHERE and HAVING can be in the same query.

The SQL HAVING syntax

The general syntax is:

```
1. SELECT column-names
2.  FROM table-name
3.  WHERE condition
4.  GROUP BY column-names
5. HAVING condition
```

Summary

- The GROUP BY Clause is used to group rows with same values.
- The GROUP BY Clause is used together with the SQL SELECT statement.
- The SELECT statement used in the GROUP BY clause can only be used contain column names, aggregate functions, constants and expressions.
- The HAVING clause is used to restrict the results returned by the GROUP BY clause.

7 Page Assignment Group By & Having Clauses

1. Print the description and total qty sold for each product.

Sol : mysql> select product_master.description , sum(sales_order_details.qty_ordered)
from product_master inner join sales_order_details on product_master.product_no = sales_order_details.product_no group by product_master.product_no;

```
+--------------+-----------------------------------+
| description  | sum(sales_order_details.qty_ordered) |
+--------------+-----------------------------------+
| 1.44 Floppies |                34 |
| Monitors      |                 6 |
| Mouse         |                 1 |
| Keyboards     |                 3 |
| CD Drive      |                 5 |
| 540 HDD       |                3 |
| 1.44 Drive    |                6 |
+--------------+-----------------------------------+
7 rows in set (0.00 sec)
```

2. Find the value of each product sold.

Sol : 1)select product_no, sum(product_rate)
from sales_order_details
group by product_no;

2)mysql> select product_no , (sum(qty_ordered)*avg(product_rate)) as value from sales_order_details group by product_no;

```
+------------  +--------------        +
| product_no | value                  |
+------------  +--------------        +
| P00001       | 17850.000000       |
| P03453       |6300.000000         |
| P06734       | 12000.000000       |
| P07868       |9450.000000         |
| P07885       | 26250.000000       |
| P07965       | 25200.000000       |
| P07975       |6300.000000         |
+------------  +--------------        +

7 rows in set (0.00 sec)
```

3. Calculate the average qty sold for each client that has a maximum order value of 15000.

Sol : 1)select s.client_no, d.order_no, avg(d.qty_ordered), sum(product_rate) as sum_rate
from sales_order_details d, sales_order s
where d.order_no=s.order_no
group by d.order_no having sum_rate <= 15000;

2)mysql>select sales_order.client_no ,
avg(sales_order_details.qty_ordered) as qauntity ,
sum(sales_order_details.qty_ordered*sales_order_details.product_rate) as order_value from sales_order

-> inner join sales_order_details on sales_order.order_no =
sales_order_details.order_no -> group by client_no

-> having order_value < 15000;

```
+-----------   +----------    +------------- +
| client_no | qauntity | order_value |
+-----------   +----------    +------------- +
| C00002     |10.0000     |        5250.00 |
| C00004     |1.0000       |        9450.00 |
| C00005     |7.5000       |        10500.00 |
+-----------   +----------    +------------- +
```

3 rows in set (0.00 sec)

4. Find out the sum total of all the billed orders for the month of January.

Sol : select s.order_no, s.order_date, sum(product_rate)
from sales_order s, sales_order_details d
where s.order_no=d.order_no and s.order_date like '_____01___'
group by d.order_no;

Conclusion:

- Thus, we studied Group By-Having clause in MySQL.
- The GROUP BY Clause is used together with the SQL SELECT statement.
- The SELECT statement used in the GROUP BY clause can only be used contain column names, aggregate functions, constants and expressions.
- The HAVING clause is used to restrict the results returned by the GROUP BY clause.