MySQL - GROUP BY Clause

MySQL GROUP BY Clause

The GROUP BY clause in MySQL is used to arrange identical data in a table into groups.

For example, let us suppose you have a table of sales data of an organization consisting of date, product, and sales amount. To calculate the total sales in a particular year, the GROUP BY clause can be used to group the sales of products made in that year. Similarly, you can group the data by date to calculate the total sales for each day, or by a combination of product and date to calculate the total sales for each product on each day.

This GROUP BY clause follows the WHERE clause in an SQL statement and precedes the ORDER BY or HAVING clause (if they exist). You can use GROUP BY to group values from a column, and, if you wish, perform calculations on that column. You can use COUNT, SUM, AVG, etc., functions on the grouped column.

Syntax

Following is the basic syntax to use GROUP BY with SELECT statement –

```
SELECT column_name(s) FROM table_name
GROUP BY [condition | column_name(s)];
```

Example

This example demonstrates how to use aggregate functions with GROUP BY clause.

First of all, create a table named CUSTOMERS, using the following CREATE TABLE query –

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR (20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

```
INSERT INTO CUSTOMERS VALUES
(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, 'Hyderabad', 4500.00),
(7, 'Muffy', 24, 'Indore', 10000.00);
```

Now, use the following GROUP BY query to group the customers based on their age –

```
SELECT AGE, COUNT(Name) FROM CUSTOMERS GROUP BY AGE;
```

MySQL GROUP BY on Single Column

When we use the GROUP BY clause on a single column, all common values in that column will be added together making it a single record.

Example

In this example, let us group the customers by their age and calculate the average salary for each age using the following query –

```
SELECT AGE, AVG(SALARY) AS AVG_SALARY
FROM CUSTOMERS
GROUP BY AGE;
```

MySQL GROUP BY on Multiple Columns

When we use the GROUP BY clause with multiple columns, the common record obtained by combining values from these columns will be grouped together into a single record.

Example

In this example, if you want to know the total amount of salary for each customer age wise, then the GROUP BY query would be as follows –

```
SELECT CONCAT(AGE, ' - ', SALARY) AS SALARY_AGEWISE
FROM CUSTOMERS
GROUP BY AGE, SALARY;
```

MySQL GROUP BY with ORDER BY Clause

We can use the ORDER BY clause with GROUP BY in MySQL to sort the result set by one or more columns.

Syntax

Following is the syntax for using ORDER BY clause with GROUP BY clause in SQL –

```

```

```
SELECT column1, column2, ..., aggregate_function(columnX) AS alias
FROM table
GROUP BY column1, column2, ...
ORDER BY column1 [ASC | DESC], column2 [ASC | DESC], ...;
```

Example

In here, we are trying to find the highest salary for each age, sorted by high to low –

```
SELECT AGE, MAX(salary) AS MAX_SALARY
FROM CUSTOMERS
GROUP BY AGE
ORDER BY MAX(salary) DESC;
```

MySQL GROUP BY with HAVING Clause

We can also use the GROUP BY clause with the HAVING clause to filter the results of a query based on conditions applied to groups of data. The condition can be applied to an aggregate function that is used in the SELECT statement or to a column in the GROUP BY clause.

Syntax

Following is the syntax for using ORDER BY clause with HAVING clause in SQL –

```
SELECT column1, column2, aggregate_function(column)
FROM table_name
GROUP BY column1, column2
HAVING condition;
```

Example

In the following query, we are grouping the customers by their age and calculating the average salary for each group. The HAVING clause is used to filter the results to show only those groups where the average salary is greater than 8000 –

```
SELECT AGE, AVG(SALARY) AS AVG_SALARY
FROM CUSTOMERS
GROUP BY AGE HAVING AVG(salary) > 8000;
```