Asir Shahriar Roudra

ID: B190305021

Assignment on string group by & having

**SQL Group by 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.

**SQL GROUP BY 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.

**Grouping using a Single Column**

In order to help understand the effect of SQL Group By clause, let’s execute a simple query that returns all the gender entries from the members table.

Background pattern

Description automatically generated

Suppose we want to get the unique values for genders. We can use a following query –

SELECT `gender` FROM `members` GROUP BY `gender`;

Executing the above script gives us the following results-

Background pattern

Description automatically generated

Note only two results have been returned. This is because we only have two gender types Male and Female. The GROUP BY clause in SQL grouped all the “Male” members together and returned only a single row for it. It did the same with the “Female” members.

**Grouping and aggregate functions**

Suppose we want total number of males and females in our database. We can use the following script shown below to do that.

SELECT `gender`,COUNT(`membership\_number`) FROM `members` GROUP BY `gender`;

Executing the above script gives us the following results-

Graphical user interface, application

Description automatically generated

The results shown below are grouped by every unique gender value posted and the number of grouped rows is counted using the COUNT aggregate function.

**Grouping using multiple columns**

Suppose that we want to get a list of movie category\_id and corresponding years in which they were released.

Let’s observe the output of this simple query

SELECT `category\_id`,`year\_released` FROM `movies` ;

Table

Description automatically generated

The above result has many duplicates.

Let’s execute the same query using group by in SQL –

SELECT `category\_id`,`year\_released` FROM `movies` GROUP BY `category\_id`,`year\_released`;

Table

Description automatically generatedExecuting the above script gives us the following results shown below-

The GROUP BY clause operates on both the category id and year released to identify unique rows in our above example.

If the category id is the same but the year released is different, then a row is treated as a unique one. If the category id and the year released is the same for more than one row, then it’s considered a duplicate and only one row is shown.

**SQL HAVING Clause**

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

**HAVING Syntax**

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);

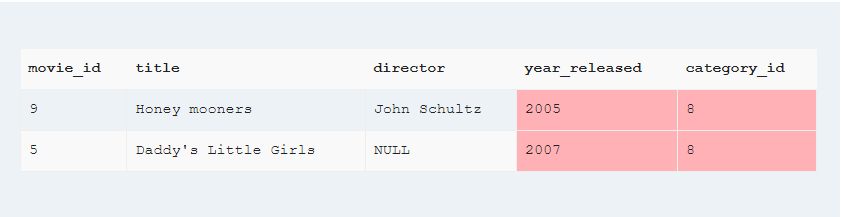
**Restricting query results using the HAVING clause**

It’s not always that we will want to perform groupings on all the data in a given table. There will be times when we will want to restrict our results to a certain given criteria. In such cases , we can use the HAVING clause.

Suppose we want to know all the release years for movie category id 8. We would use the following script to achieve our results.

SELECT \* FROM `movies` GROUP BY `category\_id`,`year\_released` HAVING `category\_id` = 8;

Executing the above script gives us the following results shown below-



Note only movies with category id 8 have been affected by our GROUP BY clause.

Summary

* The GROUP BY Clause SQL 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.
* SQL Having Clause is used to restrict the results returned by the GROUP BY clause.
* MYSQL GROUP BY Clause is used to collect data from multiple records and returned record set by one or more columns.