|  |  |
| --- | --- |
| **Name** | Sahil Shah |
| **UID No.** | 2021300115 |
| **Experiment No.** | 5 |

|  |  |
| --- | --- |
| **AIM:** | To perform aggregate functions and Group By-Having clause on the database. |
| **Program 1** | |
| **PROBLEM STATEMENT:** | Write queries on the tables in the database using aggregate functions on MySQL. |
| **THEORY:** | **SQL Aggregate Functions**   * SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value. * It is also used to summarize the data.   **Types of SQL Aggregate Functions**     1. The COUNT() function returns the number of rows that matches a specified criterion.   **Its syntax is as follows:**  SELECT COUNT(column\_name)  FROM table\_name  WHERE condition;   1. The AVG() function returns the average value of a numeric column.   **Its syntax is as follows:**  SELECT AVG(column\_name)  FROM table\_name  WHERE condition;   1. The SUM() function returns the total sum of a numeric column.   **Its syntax is as follows:**  SELECT SUM(column\_name)  FROM table\_name  WHERE condition;   1. The MIN() function returns the smallest value of the selected column.   **Its syntax is as follows:**  SELECT MIN(column\_name)  FROM table\_name  WHERE condition;   1. The MAX() function returns the largest value of the selected column.   **Its syntax is as follows:**  SELECT MAX(column\_name)  FROM table\_name  WHERE condition;  **MySQL GROUP BY Statement**   1. The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country". 2. The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result set by one or more columns.   **Its syntax is as follows:**  SELECT column\_name(s)  FROM table\_name  WHERE condition  GROUP BY column\_name(s)  ORDER BY column\_name(s);  **MySQL HAVING Clause**   1. The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.   **Its syntax is as follows:**  SELECT column\_name(s)  FROM table\_name  WHERE condition  GROUP BY column\_name(s)  HAVING condition  ORDER BY column\_name(s); |
| **QUERIES:**  **Using Create, Insert Into, Select Commands:**    **Table Room**    **Using Create, Insert Into, Select Commands:**    **Table Customers**     1. Using the Count function to find the count of the room numbers having price greater than 3500 and group by room type:   SELECT COUNT(RoomPrice) AS RP3500, RoomType  FROM Room  GROUP BY RoomType DESC  HAVING RoomType = 'Deluxe' OR RoomType = 'A.C';     1. Using the Count function to find the number of rooms available having a room number less than 200:   SELECT COUNT(RoomNumber) AS RN200  FROM Room  WHERE RoomNumber < 200;     1. Using the Min function to find the minimum room number of room type A.C:   SELECT MIN(RoomNumber) AS MinRoomNumber  FROM Room  WHERE RoomType = 'A.C';     1. Using the Min function to find the minimum room number when the room is available and the price of the room is greater than or equal to 3500:   SELECT MIN(RoomNumber) AS MRNA, RoomType  FROM Room  WHERE RoomAvailability = 'YES' AND RoomPrice >= 3500;     1. Using the Max function to find the maximum room number and room type where the room size is 2 persons:   SELECT MAX(RoomNumber) AS MaxRoomNumber, RoomType  FROM Room  WHERE RoomSize = '2 persons';     1. Using the Max function to find the maximum room number, room size, and room availability where the room type is A.C:   SELECT MAX(RoomNumber) AS MRNRT, RoomSize, RoomAvailability  FROM Room  WHERE RoomType = 'A.C';     1. Using the Avg function to find the average room price grouped by the room type and which are available:   SELECT AVG(Roomprice) AS AvgRoomPrice, RoomType, RoomAvailability  FROM Room  GROUP BY RoomType  HAVING RoomAvailability = 'YES';     1. Using the Avg function to find the average room price given that the room type is ‘A.C’:   SELECT AVG(Roomprice) AS ARPRT, RoomType  FROM Room  WHERE RoomType = 'A.C';     1. Using the Avg function to find the average room price given that the room number is less than 400:   SELECT AVG(Roomprice) AS ARPRN  FROM Room  WHERE RoomNumber < 400;     1. Using the Sum function to find the sum of the prices of the rooms grouped by the room size:   SELECT SUM(RoomPrice) AS SumRoomPrice, RoomSize  FROM Room  GROUP BY RoomSize;     1. Using the Sum function to find the sum of the room prices of the rooms which are available and have room numbers less than 300:   SELECT SUM(RoomPrice) AS SRPRARN  FROM Room  WHERE RoomAvailability = 'NO' AND RoomNumber < 300;     1. Using the Sum function to find the sum of the room prices which have room numbers greater than 100 or the size for 2 persons:   SELECT SUM(RoomPrice) AS SRPRNRS  FROM Room  WHERE RoomNumber > 100 OR RoomSize = '2 persons';     1. Using the Sum function to find the sum of room prices grouped by room type and ordered by the room type in descending order:   SELECT RoomType, SUM(RoomPrice) AS SRP  FROM Room  GROUP BY RoomType  ORDER BY RoomType DESC;     1. Using the Sum function to find the sum of room prices grouped by room type having the sum of the prices of the room types greater than 5000 and ordered in descending order by the room type:   SELECT RoomType, SUM(RoomPrice) AS HSRP  FROM Room  GROUP BY RoomType  HAVING HSRP >= 5000  ORDER BY RoomType DESC; | |
| **CONCLUSION:**  I learned about various types of aggregate functions in MySQL in this experiment – which are the Sum, Count, Avg, Min, and Max functions. I also learned about the Group By and Having clauses which when coupled with the aggregate functions enabled me to successfully complete this experiment. | |