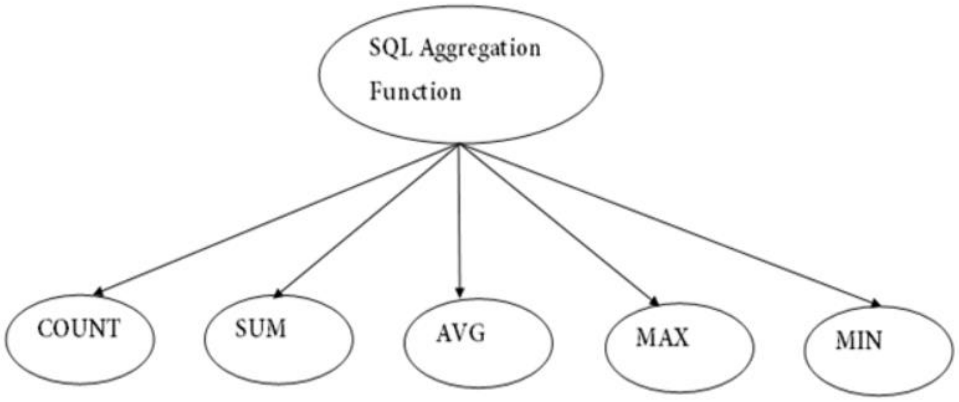
Name	Sahil Shah
UID No.	2021300115
Experiment No.	5

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**  SQL Aggregation Function COUNT SUM AVG MAX MIN 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;
```

2) 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;
   ```

3) 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;
   ```

4) 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;
   ```

5) 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

6) The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
7) 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

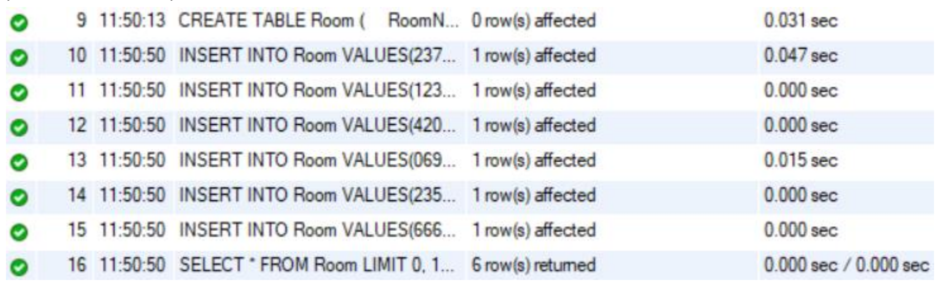
8) The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

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);
```

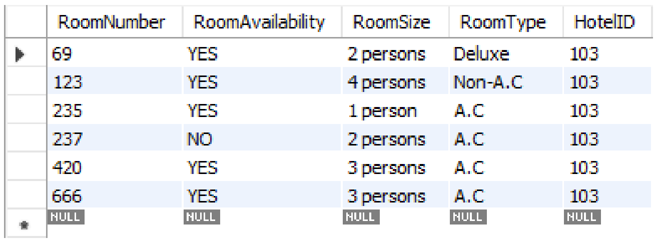
QUERIES:

Using Create, Insert Into, Select Commands:



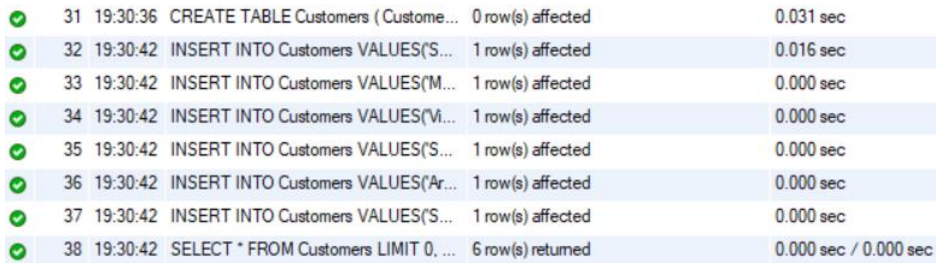
```
✓  9  11:50:13  CREATE TABLE Room (    RoomN...   0 row(s) affected   0.031 sec
✓ 10  11:50:50  INSERT INTO Room VALUES(237...   1 row(s) affected   0.047 sec
✓ 11  11:50:50  INSERT INTO Room VALUES(123...   1 row(s) affected   0.000 sec
✓ 12  11:50:50  INSERT INTO Room VALUES(420...   1 row(s) affected   0.000 sec
✓ 13  11:50:50  INSERT INTO Room VALUES(069...   1 row(s) affected   0.015 sec
✓ 14  11:50:50  INSERT INTO Room VALUES(235...   1 row(s) affected   0.000 sec
✓ 15  11:50:50  INSERT INTO Room VALUES(666...   1 row(s) affected   0.000 sec
✓ 16  11:50:50  SELECT * FROM Room LIMIT 0, 1...  6 row(s) returned   0.000 sec / 0.000 sec
```

Table Room



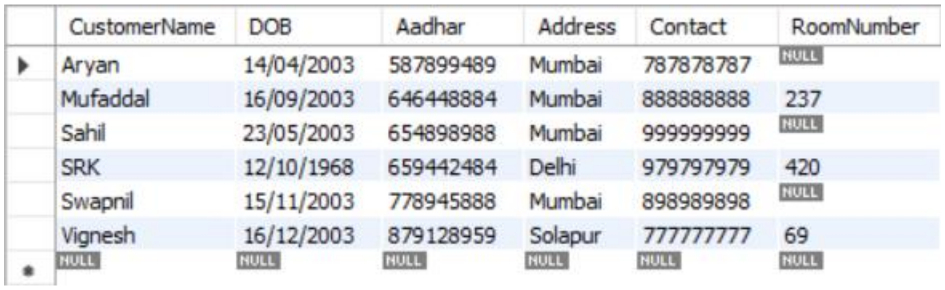
RoomNumber	RoomAvailability	RoomSize	RoomType	HotelID
69	YES	2 persons	Deluxe	103
123	YES	4 persons	Non-A.C	103
235	YES	1 person	A.C	103
237	NO	2 persons	A.C	103
420	YES	3 persons	A.C	103
666	YES	3 persons	A.C	103
NULL	NULL	NULL	NULL	NULL

Using Create, Insert Into, Select Commands:



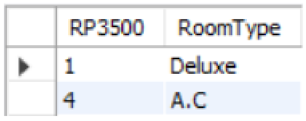
```
31  19:30:36  CREATE TABLE Customers ( Custome...  0 row(s) affected   0.031 sec
32  19:30:42  INSERT INTO Customers VALUES('S...   1 row(s) affected   0.016 sec
33  19:30:42  INSERT INTO Customers VALUES('M...   1 row(s) affected   0.000 sec
34  19:30:42  INSERT INTO Customers VALUES('Vi...  1 row(s) affected   0.000 sec
35  19:30:42  INSERT INTO Customers VALUES('S...   1 row(s) affected   0.000 sec
36  19:30:42  INSERT INTO Customers VALUES('Ar...  1 row(s) affected   0.000 sec
37  19:30:42  INSERT INTO Customers VALUES('S...   1 row(s) affected   0.000 sec
38  19:30:42  SELECT * FROM Customers LIMIT 0, ... 6 row(s) returned   0.000 sec / 0.000 sec
```

Table Customers



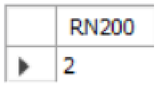
CustomerName	DOB	Aadhar	Address	Contact	RoomNumber
Aryan	14/04/2003	587899489	Mumbai	787878787	NULL
Mufaddal	16/09/2003	646448884	Mumbai	888888888	237
Sahil	23/05/2003	654898988	Mumbai	999999999	NULL
SRK	12/10/1968	659442484	Delhi	979797979	420
Swapnil	15/11/2003	778945888	Mumbai	898989898	NULL
Vignesh	16/12/2003	879128959	Solapur	777777777	69
NULL	NULL	NULL	NULL	NULL	NULL

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';



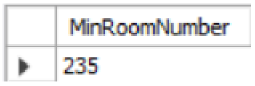
RP3500	RoomType
1	Deluxe
4	A.C

2) 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;



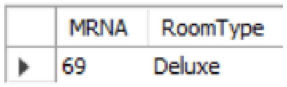
RN200
2

3) 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';



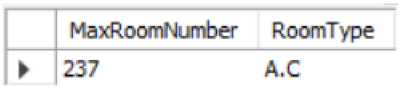
	MinRoomNumber
▶	235

4) 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;



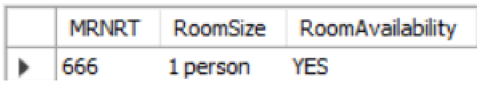
	MRNA	RoomType
▶	69	Deluxe

5) 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';



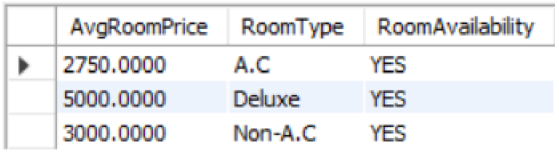
	MaxRoomNumber	RoomType
▶	237	A.C

6) 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';



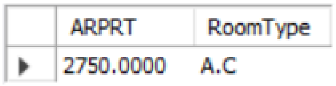
	MRNRT	RoomSize	RoomAvailability
▶	666	1 person	YES

7) 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';



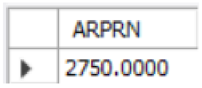
	AvgRoomPrice	RoomType	RoomAvailability
▶	2750.0000	A.C	YES
	5000.0000	Deluxe	YES
	3000.0000	Non-A.C	YES

8) 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';



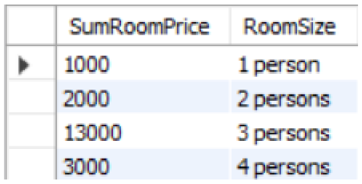
	ARPRT	RoomType
▶	2750.0000	A.C

9) 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;



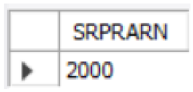
	ARPRN
▶	2750.0000

10) 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;



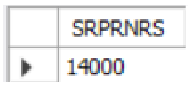
	SumRoomPrice	RoomSize
▶	1000	1 person
	2000	2 persons
	13000	3 persons
	3000	4 persons

11) 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;



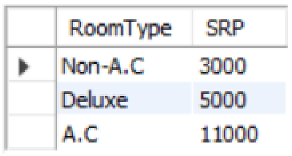
	SRPRARN
▶	2000

12) 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';



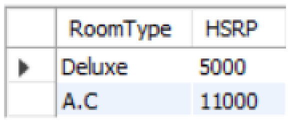
	SRPRNRS
▶	14000

13) 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;



	RoomType	SRP
▶	Non-A.C	3000
	Deluxe	5000
	A.C	11000

14) 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;



	RoomType	HSRP
▶	Deluxe	5000
	A.C	11000

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.