MIT World Peace University

Database Management Systems
Second Year B. Tech, Semester 4

Group Functions, Join and Nested Queries.

Assignment No. 4

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

April 13, 2023

Contents

1 Aim

Write suitable select command to get requested data from tables

2 Objectives

1. To study Subqueries, Group, Joins and Views

3 Problem Statement

Create tables and solve given queries using , Group, Joins and Views

4 Theory

4.1 Group Functions

Group functions in SQL are functions that operate on groups of rows in a table, and return a single result for each group. They are often used in conjunction with the GROUP BY clause, which divides a table into groups based on one or more columns, and applies the group functions to each group.

The common group functions in SQL are:

1. COUNT - Counts the number of rows in a group.
2. SUM - Calculates the sum of a column in a group.
3. AVG - Calculates the average value of a column in a group.
4. MIN - Finds the minimum value of a column in a group.
5. MAX - Finds the maximum value of a column in a group.

Syntax:

```
SELECT column_name, group_function(column_name)
FROM table_name
WHERE condition
GROUP BY column_name;
```

Example:
Consider the following table "employees":

id	name	department	salary
1	Alice	HR	5000
2	Bob	IT	6000
3	Charlie	HR	4500
4	David	IT	7000
5	Emma	Sales	5500

To count the number of employees in each department, you can use the following query:

```
SELECT department, COUNT(*)
FROM employees
GROUP BY department;
```

department	**COUNT(*)**
HR	2
IT	2
Sales	1

To find the average salary of employees in each department, you can use the following query:

```
SELECT department, AVG(salary)
FROM employees
GROUP BY department;
```

department	**AVG(salary)**
HR	4750
IT	6500
Sales	5500

To find the maximum salary in the entire table, you can use the following query:

```
SELECT MAX(salary)
FROM employees;
```

MAX(salary)
7000

4.2 SQL Join Types

In SQL, join is used to combine two or more tables based on a common column between them. There are several types of joins in SQL, each with its own syntax and usage.

Consider the Following Tables

```
MariaDB [dbms_lab]> select * from booking;
+---------+---------+------------+------------+--------+
| HotelNo | GuestNo | DateFrom   | DateTo     | RoomNo |
+---------+---------+------------+------------+--------+
|       7 |      10 | 2096-04-21 | 2099-12-21 |     10 |
|       8 |       5 | 2077-09-29 | 2109-09-10 |     11 |
|      11 |       4 | 2123-01-05 | 2063-08-30 |      2 |
|      10 |       5 | 2027-02-05 | 2119-12-21 |      7 |
|       9 |       5 | 2081-07-11 | 2031-06-20 |     13 |
|       5 |       5 | 2059-11-19 | 2113-05-22 |     11 |
+---------+---------+------------+------------+--------+
6 rows in set (0.001 sec)

MariaDB [dbms_lab]> select * from Hotel;
+---------+------------------+-----------------------+
| HotelNo | Name             | City                  |
+---------+------------------+-----------------------+
|       1 | Hotel love       | Guernsey              |
|       2 | Hotel imagine    | Jordan                |
|       3 | Hotel rice       | Equatorial Guinea     |
|       4 | Hotel perhaps    | Bolivia               |
|       5 | Hotel show       | Reunion               |
|       6 | Hotel native     | Brunei                |
|       7 | Hotel pool       | Panama                |
```

```
|       8 | Hotel spin       | Guyana                 |
|       9 | Hotel toward     | St. Barthelemy         |
|      10 | Hotel expression | St. Pierre & Miquelon  |
|      11 | Hotel cheese     | Guinea-Bissau          |
|      12 | Hotel motion     | Latvia                 |
|      13 | Hotel lay        | Fiji                   |
|      14 | Hotel stiff      | Brazil                 |
|      15 | Hotel suddenly   | Lithuania              |
|      16 | Hotel stretch    | Montenegro             |
|      17 | Hotel current    | Isle of Man            |
|      18 | Hotel forest     | Haiti                  |
+---------+------------------+------------------------+
18 rows in set (0.001 sec)
```

4.3 Inner Join or Simple Join

Defintion

The inner join is used to select all matching rows or columns in both tables or as long as the defined condition is valid in SQL.

Figure

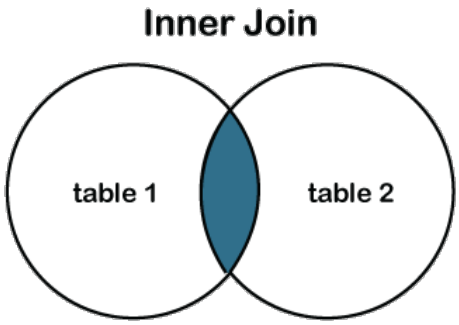

Figure 1: Inner Join

Syntax

```
Select column_1, column_2, column_3 FROM table_1 INNER JOIN table_2 ON table_1.
    column = table_2.column;
```

Example

```
MariaDB [dbms_lab]> select booking.HotelNo, Hotel.name from booking inner join
    Hotel on booking.HotelNo = Hotel.HotelNo;
+---------+------------------+
| HotelNo | name             |
+---------+------------------+
|       5 | Hotel show       |
|       7 | Hotel pool       |
|       8 | Hotel spin       |
```

```
|        9 | Hotel toward      |
|       10 | Hotel expression  |
|       11 | Hotel cheese      |
+----------+-------------------+
6 rows in set (0.001 sec)
```

4.4 Left Join

Defintion

The LEFT JOIN is used to retrieve all records from the left table (table1) and the matched rows or columns from the right table (table2). If both tables do not contain any matched rows or columns, it returns the NULL.

Figure

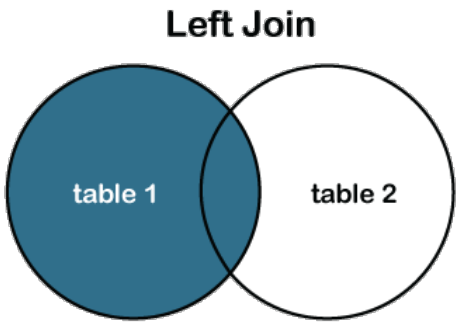

Figure 2: Left Join

Syntax

```
Select column_1, column_2, column(s) FROM table_1 LEFT JOIN table_2 ON table_1.
    column_name = table_2.column_name;
```

Example

```
MariaDB [dbms_lab]> select * from booking left join Hotel on booking.HotelNo =
    Hotel.HotelNo;
+--------+---------+-------------------+------------------------+
| RoomNo | HotelNo | Name              | City                   |
+--------+---------+-------------------+------------------------+
|     10 |       7 | Hotel pool        | Panama                 |
|     11 |       8 | Hotel spin        | Guyana                 |
|      2 |      11 | Hotel cheese      | Guinea-Bissau          |
|      7 |      10 | Hotel expression  | St. Pierre & Miquelon  |
|     13 |       9 | Hotel toward      | St. Barthelemy         |
|     11 |       5 | Hotel show        | Reunion                |
+--------+---------+-------------------+------------------------+
6 rows in set (0.001 sec)
```

4.5 Right Join

Defintion

The RIGHT JOIN is used to retrieve all records from the right table (table2) and the matched rows or columns from the left table (table1). If both tables do not contain any matched rows or columns, it returns the NULL.

Figure

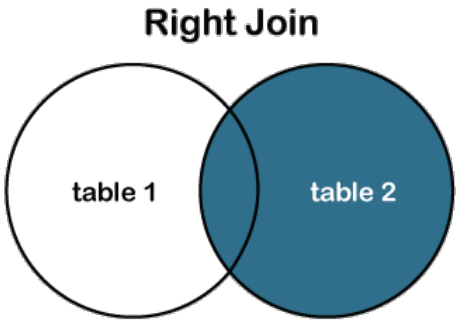

Figure 3: Right Join

Syntax

```
Select column_1, column_2, column_3 FROM table_1 RIGHT JOIN table_2 ON table_1.
    column = table_2.column;
```

Example

```

MariaDB [dbms_lab]> select * from booking right join Hotel on booking.HotelNo =
    Hotel.HotelNo;
+--------+---------+-------------------+-----------------------+
| RoomNo | HotelNo | Name              | City                  |
+--------+---------+-------------------+-----------------------+
|     10 |       7 | Hotel pool        | Panama                |
|     11 |       8 | Hotel spin        | Guyana                |
|      2 |      11 | Hotel cheese      | Guinea-Bissau         |
|      7 |      10 | Hotel expression  | St. Pierre & Miquelon |
|     13 |       9 | Hotel toward      | St. Barthelemy        |
|     11 |       5 | Hotel show        | Reunion               |
|   NULL |       1 | Hotel love        | Guernsey              |
|   NULL |       2 | Hotel imagine     | Jordan                |
|   NULL |       3 | Hotel rice        | Equatorial Guinea     |
|   NULL |       4 | Hotel perhaps     | Bolivia               |
|   NULL |       6 | Hotel native      | Brunei                |
|   NULL |      12 | Hotel motion      | Latvia                |
|   NULL |      13 | Hotel lay         | Fiji                  |
|   NULL |      14 | Hotel stiff       | Brazil                |
|   NULL |      15 | Hotel suddenly    | Lithuania             |
|   NULL |      16 | Hotel stretch     | Montenegro            |
```

```
|   NULL |       17 | Hotel current    | Isle of Man           |
|   NULL |       18 | Hotel forest     | Haiti                 |
+--------+----------+------------------+-----------------------+
18 rows in set (0.002 sec)
```

4.6 Natural Join

Defintion

It is a type of inner type that joins two or more tables based on the same column name and has the same data type present on both tables.

Syntax

```
Select * from tablename1 Natural JOIN tablename_2;
```

Example

```
MariaDB [dbms_lab]> select * from booking natural join Hotel
-> ;
+--------+------------------+-----------------------+
| RoomNo | Name             | City                  |
+--------+------------------+-----------------------+
|     10 | Hotel pool       | Panama                |
|     11 | Hotel spin       | Guyana                |
|      2 | Hotel cheese     | Guinea-Bissau         |
|      7 | Hotel expression | St. Pierre & Miquelon |
|     13 | Hotel toward     | St. Barthelemy        |
|     11 | Hotel show       | Reunion               |
+--------+------------------+-----------------------+
6 rows in set (0.001 sec)
```

4.7 Full Outer Join

Defintion

It is a combination result set of both LEFT JOIN and RIGHT JOIN. The joined tables return all records from both the tables and if no matches are found in the table, it places NULL. It is also called a FULL OUTER JOIN.

Figure

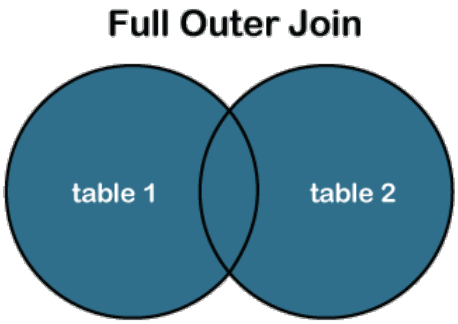

Figure 4: Right Join

Syntax

```
  Select column_1, column_2, column(s) FROM table_1 FULL JOIN table_2 ON table_1.
    column_name = table_2.column_name;
```

Example

```

MariaDB [dbms_lab]> select * from booking left join Hotel on booking.HotelNo =
    Hotel.HotelNo
-> union
-> select * from booking right join Hotel on booking.HotelNo = Hotel.HotelNo;
+--------+---------+-------------------+-----------------------+
| RoomNo | HotelNo | Name              | City                  |
+--------+---------+-------------------+-----------------------+
|     10 |       7 | Hotel pool        | Panama                |
|     11 |       8 | Hotel spin        | Guyana                |
|      2 |      11 | Hotel cheese      | Guinea-Bissau         |
|      7 |      10 | Hotel expression  | St. Pierre & Miquelon |
|     13 |       9 | Hotel toward      | St. Barthelemy        |
|     11 |       5 | Hotel show        | Reunion               |
|   NULL |       1 | Hotel love        | Guernsey              |
|   NULL |       2 | Hotel imagine     | Jordan                |
|   NULL |       3 | Hotel rice        | Equatorial Guinea     |
|   NULL |       4 | Hotel perhaps     | Bolivia               |
|   NULL |       6 | Hotel native      | Brunei                |
|   NULL |      12 | Hotel motion      | Latvia                |
|   NULL |      13 | Hotel lay         | Fiji                  |
|   NULL |      14 | Hotel stiff       | Brazil                |
|   NULL |      15 | Hotel suddenly    | Lithuania             |
|   NULL |      16 | Hotel stretch     | Montenegro            |
|   NULL |      17 | Hotel current     | Isle of Man           |
|   NULL |      18 | Hotel forest      | Haiti                 |
+--------+---------+-------------------+-----------------------+
18 rows in set (0.008 sec)
```

4.8 Cross Join

Defintion

It is also known as CARTESIAN JOIN, which returns the Cartesian product of two or more joined tables. The CROSS JOIN produces a table that merges each row from the first table with each second table row. It is not required to include any condition in CROSS JOIN.

Syntax

```
Select * from table_1 cross join table_2;
```

Example

```

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> select booking.HotelNo, booking.RoomNo, Hotel.name, Hotel.City
    from booking cross join Hotel;
+---------+--------+------------------+-----------------------+
| HotelNo | RoomNo | name             | City                  |
+---------+--------+------------------+-----------------------+
|       7 |     10 | Hotel love       | Guernsey              |
|       8 |     11 | Hotel love       | Guernsey              |
|      11 |      2 | Hotel love       | Guernsey              |
|      10 |      7 | Hotel love       | Guernsey              |
|       9 |     13 | Hotel love       | Guernsey              |
|       5 |     11 | Hotel love       | Guernsey              |
|       7 |     10 | Hotel imagine    | Jordan                |
|       8 |     11 | Hotel imagine    | Jordan                |
|      11 |      2 | Hotel imagine    | Jordan                |
|      10 |      7 | Hotel imagine    | Jordan                |
|       9 |     13 | Hotel imagine    | Jordan                |
|       5 |     11 | Hotel imagine    | Jordan                |
|       7 |     10 | Hotel rice       | Equatorial Guinea     |
|       8 |     11 | Hotel rice       | Equatorial Guinea     |
|      11 |      2 | Hotel rice       | Equatorial Guinea     |
|      10 |      7 | Hotel rice       | Equatorial Guinea     |
|       9 |     13 | Hotel rice       | Equatorial Guinea     |
|       5 |     11 | Hotel rice       | Equatorial Guinea     |
|       7 |     10 | Hotel perhaps    | Bolivia               |
|       8 |     11 | Hotel perhaps    | Bolivia               |
|      11 |      2 | Hotel perhaps    | Bolivia               |
|      10 |      7 | Hotel perhaps    | Bolivia               |
|       9 |     13 | Hotel perhaps    | Bolivia               |
|       5 |     11 | Hotel perhaps    | Bolivia               |
|       7 |     10 | Hotel show       | Reunion               |
|       8 |     11 | Hotel show       | Reunion               |
|      11 |      2 | Hotel show       | Reunion               |
|      10 |      7 | Hotel show       | Reunion               |
|       9 |     13 | Hotel show       | Reunion               |
|       5 |     11 | Hotel show       | Reunion               |
|       7 |     10 | Hotel native     | Brunei                |
|       8 |     11 | Hotel native     | Brunei                |
|      11 |      2 | Hotel native     | Brunei                |
|      10 |      7 | Hotel native     | Brunei                |
|       9 |     13 | Hotel native     | Brunei                |
```

```
|        5 |     11 | Hotel native     | Brunei                |
|        7 |     10 | Hotel pool       | Panama                |
|        8 |     11 | Hotel pool       | Panama                |
|       11 |      2 | Hotel pool       | Panama                |
|       10 |      7 | Hotel pool       | Panama                |
|        9 |     13 | Hotel pool       | Panama                |
|        5 |     11 | Hotel pool       | Panama                |
|        7 |     10 | Hotel spin       | Guyana                |
|        8 |     11 | Hotel spin       | Guyana                |
|       11 |      2 | Hotel spin       | Guyana                |
|       10 |      7 | Hotel spin       | Guyana                |
|        9 |     13 | Hotel spin       | Guyana                |
|        5 |     11 | Hotel spin       | Guyana                |
|        7 |     10 | Hotel toward     | St. Barthelemy        |
|        8 |     11 | Hotel toward     | St. Barthelemy        |
|       11 |      2 | Hotel toward     | St. Barthelemy        |
|       10 |      7 | Hotel toward     | St. Barthelemy        |
|        9 |     13 | Hotel toward     | St. Barthelemy        |
|        5 |     11 | Hotel toward     | St. Barthelemy        |
|        7 |     10 | Hotel expression | St. Pierre & Miquelon |
|        8 |     11 | Hotel expression | St. Pierre & Miquelon |
|       11 |      2 | Hotel expression | St. Pierre & Miquelon |
|       10 |      7 | Hotel expression | St. Pierre & Miquelon |
|        9 |     13 | Hotel expression | St. Pierre & Miquelon |
|        5 |     11 | Hotel expression | St. Pierre & Miquelon |
|        7 |     10 | Hotel cheese     | Guinea-Bissau         |
|        8 |     11 | Hotel cheese     | Guinea-Bissau         |
|       11 |      2 | Hotel cheese     | Guinea-Bissau         |
|       10 |      7 | Hotel cheese     | Guinea-Bissau         |
|        9 |     13 | Hotel cheese     | Guinea-Bissau         |
|        5 |     11 | Hotel cheese     | Guinea-Bissau         |
|        7 |     10 | Hotel motion     | Latvia                |
|        8 |     11 | Hotel motion     | Latvia                |
|       11 |      2 | Hotel motion     | Latvia                |
|       10 |      7 | Hotel motion     | Latvia                |
|        9 |     13 | Hotel motion     | Latvia                |
|        5 |     11 | Hotel motion     | Latvia                |
|        7 |     10 | Hotel lay        | Fiji                  |
|        8 |     11 | Hotel lay        | Fiji                  |
|       11 |      2 | Hotel lay        | Fiji                  |
|       10 |      7 | Hotel lay        | Fiji                  |
|        9 |     13 | Hotel lay        | Fiji                  |
|        5 |     11 | Hotel lay        | Fiji                  |
|        7 |     10 | Hotel stiff      | Brazil                |
|        8 |     11 | Hotel stiff      | Brazil                |
|       11 |      2 | Hotel stiff      | Brazil                |
|       10 |      7 | Hotel stiff      | Brazil                |
|        9 |     13 | Hotel stiff      | Brazil                |
|        5 |     11 | Hotel stiff      | Brazil                |
|        7 |     10 | Hotel suddenly   | Lithuania             |
|        8 |     11 | Hotel suddenly   | Lithuania             |
|       11 |      2 | Hotel suddenly   | Lithuania             |
|       10 |      7 | Hotel suddenly   | Lithuania             |
|        9 |     13 | Hotel suddenly   | Lithuania             |
|        5 |     11 | Hotel suddenly   | Lithuania             |
|        7 |     10 | Hotel stretch    | Montenegro            |
|        8 |     11 | Hotel stretch    | Montenegro            |
|       11 |      2 | Hotel stretch    | Montenegro            |
|       10 |      7 | Hotel stretch    | Montenegro            |
```

```
|        9 |      13 | Hotel stretch    | Montenegro             |
|        5 |      11 | Hotel stretch    | Montenegro             |
|        7 |      10 | Hotel current    | Isle of Man            |
|        8 |      11 | Hotel current    | Isle of Man            |
|       11 |       2 | Hotel current    | Isle of Man            |
|       10 |       7 | Hotel current    | Isle of Man            |
|        9 |      13 | Hotel current    | Isle of Man            |
|        5 |      11 | Hotel current    | Isle of Man            |
|        7 |      10 | Hotel forest     | Haiti                  |
|        8 |      11 | Hotel forest     | Haiti                  |
|       11 |       2 | Hotel forest     | Haiti                  |
|       10 |       7 | Hotel forest     | Haiti                  |
|        9 |      13 | Hotel forest     | Haiti                  |
|        5 |      11 | Hotel forest     | Haiti                  |
+----------+---------+------------------+------------------------+
108 rows in set (0.000 sec)
```

4.9 Self Join

Defintion

It is a SELF JOIN used to create a table by joining itself as there were two tables. It makes temporary naming of at least one table in an SQL statement.

Syntax

```
  Select column1, column2, column(s) FROM table_1 Tbl1, table_1 Tbl2 WHERE
    condition;
```

5 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Draw.io for Drawing the ER diagram.

6 Input

Given Database from the Problem Statement for the Assignment for our batch. (A1 PA 20)

7 Executed Queries

7.1 Questions SetA

```
MariaDB [dbms_lab]> select * from Room;
+--------+---------+-------+-------+
| RoomNo | HotelNo | Type  | Price |
+--------+---------+-------+-------+
|      1 |       1 | Suite |  1646 |
|      2 |       2 | Suite |  1264 |
|      3 |       1 | 2 Bed |   773 |
|      4 |       4 | 2 Bed |  1949 |
|      5 |       1 | 3 Bed |  1959 |
|      6 |       3 | 3 Bed |   674 |
```

```
|      7 |       1 | 1 Bed |  1018 |
|      8 |       3 | 1 Bed |  1314 |
|      9 |       1 | Suite |  1308 |
|     10 |       9 | 3 Bed |  1366 |
|     11 |      10 | 1 Bed |   666 |
|     12 |       7 | 2 Bed |  1498 |
|     13 |       7 | Suite |   984 |
+--------+---------+-------+-------+
13 rows in set (0.001 sec)

MariaDB [dbms_lab]> select * from Hotel;
+---------+------------------+-----------------------+
| HotelNo | Name             | City                  |
+---------+------------------+-----------------------+
|       1 | Hotel love       | Guernsey              |
|       2 | Hotel imagine    | Jordan                |
|       3 | Hotel rice       | Equatorial Guinea     |
|       4 | Hotel perhaps    | Bolivia               |
|       5 | Hotel show       | Reunion               |
|       6 | Hotel native     | Brunei                |
|       7 | Hotel pool       | Panama                |
|       8 | Hotel spin       | Guyana                |
|       9 | Hotel toward     | St. Barthelemy        |
|      10 | Hotel expression | St. Pierre & Miquelon |
|      11 | Hotel cheese     | Guinea-Bissau         |
|      12 | Hotel motion     | Latvia                |
|      13 | Hotel lay        | Fiji                  |
|      14 | Hotel stiff      | Brazil                |
|      15 | Hotel suddenly   | Lithuania             |
|      16 | Hotel stretch    | Montenegro            |
|      17 | Hotel current    | Isle of Man           |
|      18 | Hotel forest     | Haiti                 |
+---------+------------------+-----------------------+
18 rows in set (0.001 sec)

MariaDB [dbms_lab]> select * from booking;
+---------+---------+------------+------------+--------+
| HotelNo | GuestNo | DateFrom   | DateTo     | RoomNo |
+---------+---------+------------+------------+--------+
|       7 |      10 | 2096-04-21 | 2099-12-21 |     10 |
|       8 |       5 | 2077-09-29 | 2109-09-10 |     11 |
|      11 |       4 | 2123-01-05 | 2063-08-30 |      2 |
|      10 |       5 | 2027-02-05 | 2119-12-21 |      7 |
|       9 |       5 | 2081-07-11 | 2031-06-20 |     13 |
|       5 |       5 | 2059-11-19 | 2113-05-22 |     11 |
+---------+---------+------------+------------+--------+
6 rows in set (0.000 sec)

MariaDB [dbms_lab]> select * from Guest;
+---------+----------------+-----------------+
| GuestNo | GuestName      | GuessAddress    |
+---------+----------------+-----------------+
|       2 | Patrick Taylor | Lebanon         |
|       4 | Mattie Vargas  | St. Barthelemy  |
|       5 | Travis Frazier | Gambia          |
|      10 | Sarah Ramsey   | Jamaica         |
|      11 | Rachel Keller  | Kenya           |
|      15 | Nathan Higgins | Puerto Rico     |
|      16 | Maude Gonzales | St. Lucia       |
```

```
+---------+----------------+-----------------+
7 rows in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 1. many hotels are there?
MariaDB [dbms_lab]> select count(*) from Hotel;
+----------+
| count(*) |
+----------+
|       18 |
+----------+
1 row in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 2. the price and type of all rooms at the Grosvenor Hotel.
MariaDB [dbms_lab]> select price, type, Name from Room, Hotel where Room.HotelNo =
    Hotel.HotelNo and Name = 'Hotel love';
+-------+-------+------------+
| price | type  | Name       |
+-------+-------+------------+
|  1646 | Suite | Hotel love |
|   773 | 2 Bed | Hotel love |
|  1959 | 3 Bed | Hotel love |
|  1018 | 1 Bed | Hotel love |
|  1308 | Suite | Hotel love |
+-------+-------+------------+
5 rows in set (0.001 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 3. the number of rooms in each hotel.
MariaDB [dbms_lab]> select Room.HotelNo, Hotel.NAME,  count(*) from Room, Hotel
   where Room.HotelNo = Hotel.HotelNo group by HotelNo;
+---------+------------------+----------+
| HotelNo | NAME             | count(*) |
+---------+------------------+----------+
|       1 | Hotel love       |        5 |
|       2 | Hotel imagine    |        1 |
|       3 | Hotel rice       |        2 |
|       4 | Hotel perhaps    |        1 |
|       7 | Hotel pool       |        2 |
|       9 | Hotel toward     |        1 |
|      10 | Hotel expression |        1 |
+---------+------------------+----------+
7 rows in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 4. Update the price of all rooms by 5%.
MariaDB [dbms_lab]> select r.Price, r.Price + r.Price * 0.05 as Updated_price from
    Room r;
+-------+---------------+
| Price | Updated_price |
+-------+---------------+
|  1646 |       1728.30 |
|  1264 |       1327.20 |
|   773 |        811.65 |
|  1949 |       2046.45 |
|  1959 |       2056.95 |
|   674 |        707.70 |
```

```
|   1018 |        1068.90 |
|   1314 |        1379.70 |
|   1308 |        1373.40 |
|   1366 |        1434.30 |
|    666 |         699.30 |
|   1498 |        1572.90 |
|    984 |        1033.20 |
+-------+----------------+
13 rows in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 5. full details of all hotels in London.
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> select * from Hotel where City = 'Jordan';
+---------+---------------+--------+
| HotelNo | Name          | City   |
+---------+---------------+--------+
|       2 | Hotel imagine | Jordan |
+---------+---------------+--------+
1 row in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 6. What is the average price of a room?
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> select avg(Price) from Room;
+------------+
| avg(Price) |
+------------+
|  1263.0000 |
+------------+
1 row in set (0.000 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 7. all guests currently staying at the Grosvenor Hotel.
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> select Guest.* from Guest, booking, Hotel where Guest.GuestNo
    = booking.GuestNo and booking.HotelNo = Hotel.HotelNo and Hotel.Name = 'Hotel
    pool';
+---------+--------------+--------------+
| GuestNo | GuestName    | GuessAddress |
+---------+--------------+--------------+
|      10 | Sarah Ramsey | Jamaica      |
+---------+--------------+--------------+
1 row in set (0.001 sec)

MariaDB [dbms_lab]>
MariaDB [dbms_lab]> -- 8. the number of rooms in each hotel in London.
MariaDB [dbms_lab]>
MariaDB [dbms_lab]> select count(*) from Room, Hotel where Room.HotelNo = Hotel.
    HotelNo and Hotel.City = 'Jordan';
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.000 sec)

MariaDB [dbms_lab]>
```

7.2 Questions Set B

```
MariaDB [dbms_lab]> CREATE TABLE zipcode (
    ->   zip VARCHAR(10) PRIMARY KEY,
    ->   city VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.50 sec)

MariaDB [dbms_lab]> CREATE TABLE customers (
    ->   cno INT PRIMARY KEY,
    ->   cname VARCHAR(50) NOT NULL,
    ->   street VARCHAR(50),
    ->   zip VARCHAR(10),
    ->   phone VARCHAR(20),
    ->   CONSTRAINT zip_fk FOREIGN KEY (zip) REFERENCES zipcode (zip)
    -> );
Query OK, 0 rows affected (0.39 sec)


MariaDB [dbms_lab]> CREATE TABLE emp (
    ->   eno INT PRIMARY KEY,
    ->   ename VARCHAR(50) NOT NULL,
    ->   zip VARCHAR(10),
    ->   hdate DATE,
    ->   FOREIGN KEY (zip) REFERENCES zipcode (zip)
    -> );
Query OK, 0 rows affected (0.44 sec)

MariaDB [dbms_lab]> CREATE TABLE parts (
    ->   pno INT PRIMARY KEY,
    ->   pname VARCHAR(50) NOT NULL,
    ->   qty_on_hand INT CHECK (qty_on_hand >= 0),
    ->   price DECIMAL(10, 2) CHECK (price >= 0)
    -> );
Query OK, 0 rows affected (0.40 sec)

MariaDB [dbms_lab]> ^C
MariaDB [dbms_lab]> CREATE TABLE orders (
    ->   ono INT PRIMARY KEY,
    ->   cno INT,
    ->   receivedate DATE,
    ->   shippeddate DATE,
    ->   CONSTRAINT cno_fk FOREIGN KEY (cno) REFERENCES customers (cno)
    -> );
Query OK, 0 rows affected (0.22 sec)

MariaDB [dbms_lab]> CREATE TABLE odetails (
    ->   ono INT,
    ->   pno INT,
    ->   qty INT CHECK (qty >= 0),
    ->   PRIMARY KEY (ono, pno),
    ->   CONSTRAINT ono_fk FOREIGN KEY (ono) REFERENCES orders (ono),
    ->   CONSTRAINT pno_fk FOREIGN KEY (pno) REFERENCES parts (pno)
    -> );
Query OK, 0 rows affected (0.24 sec)

MariaDB [dbms_lab]> INSERT INTO zipcode (zip, city) VALUES
    ->   ('10001', 'New York'),
    ->   ('10002', 'Los Angeles'),
```

```
    ->   ('10003', 'Chicago');
Query OK, 3 rows affected (0.44 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [dbms_lab]> INSERT INTO emp (eno, ename, zip, hdate) VALUES
    ->   (1, 'John Smith', '10001', '2022-01-01'),
    ->   (2, 'Jane Doe', '10002', '2022-02-01'),
    ->   (3, 'Bob Johnson', '10003', '2022-03-01');
Query OK, 3 rows affected (0.39 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [dbms_lab]> INSERT INTO parts (pno, pname, qty_on_hand, price) VALUES
    ->   (1, 'Widget', 10, 19.99),
    ->   (2, 'Gizmo', 5, 29.99),
    ->   (3, 'Doodad', 20, 9.99);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [dbms_lab]> INSERT INTO customers (cno, cname, street, zip, phone) VALUES
    ->   (1, 'Acme Inc.', '123 Main St.', '10001', '555-1234'),
    ->   (2, 'Globex Corp.', '456 Elm St.', '10002', '555-5678'),
    ->   (3, 'Initech Ltd.', '789 Oak St.', '10003', '555-9101');
Query OK, 3 rows affected (0.38 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [dbms_lab]> INSERT INTO orders (ono, cno, receivedate, shippeddate) VALUES
    ->   (1, 1, '2022-01-01', '2022-01-02'),
    ->   (2, 2, '2022-02-01', NULL),
    ->   (3, 3, '2022-03-01', '2022-03-02');
Query OK, 3 rows affected (0.08 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [dbms_lab]> INSERT INTO odetails (ono, pno, qty) VALUES
    ->   (1, 1, 5),
    ->   (1, 2, 2),
    ->   (2, 3, 10),
    ->   (3, 1, 3),
    ->   (3, 2, 1),
    ->   (3, 3, 5);
Query OK, 6 rows affected (0.13 sec)
Records: 6  Duplicates: 0  Warnings: 0


-- q1
SELECT pno, pname
FROM parts
WHERE price < 20.00 at line 1
MariaDB [dbms_lab]> SELECT pno, pname
    -> FROM parts
    -> WHERE price < 20.00;
+-----+--------+
| pno | pname  |
+-----+--------+
|   1 | Widget |
|   3 | Doodad |
+-----+--------+
2 rows in set (0.00 sec)

MariaDB [dbms_lab]> -- q2
```

```
MariaDB [dbms_lab]> SELECT orders.ono, customers.cname
    -> FROM orders
    -> JOIN customers ON orders.cno = customers.cno
    -> WHERE orders.shippeddate IS NULL;
+-----+--------------+
| ono | cname        |
+-----+--------------+
|   2 | Globex Corp. |
+-----+--------------+
1 row in set (0.00 sec)

MariaDB [dbms_lab]> -- q4
MariaDB [dbms_lab]> SELECT DISTINCT zipcode.city
    -> FROM emp
    -> JOIN zipcode ON emp.zip = zipcode.zip
    -> UNION
    -> SELECT DISTINCT zipcode.city
    -> FROM customers
    -> JOIN zipcode ON customers.zip = zipcode.zip;

+------------+
| city       |
+------------+
| New York   |
| Los Angeles|
| Chicago    |
+------------+

MariaDB [dbms_lab]> -- q5
MariaDB [dbms_lab]> SELECT SUM(qty)
    -> FROM odetails
    -> WHERE pno = '1060';
+----------+
| SUM(qty) |
+----------+
|     NULL |
+----------+
1 row in set (0.00 sec)

MariaDB [dbms_lab]> -- q6
MariaDB [dbms_lab]> SELECT COUNT(*) as total_customers
    -> FROM customers;
+-----------------+
| total_customers |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.37 sec)

MariaDB [dbms_lab]> -- q7
MariaDB [dbms_lab]> CREATE VIEW customer_orders AS
    -> SELECT orders.ono, orders.receivedate, orders.shippeddate, customers.cname
    -> FROM orders
    -> JOIN customers ON orders.cno = customers.cno;
Query OK, 0 rows affected (0.44 sec)


MariaDB [dbms_lab]> select * from customer_orders;
+-----+-------------+-------------+--------------+
```

```
| ono | receivedate | shippeddate | cname        |
+-----+-------------+-------------+--------------+
|   1 | 2022-01-01  | 2022-01-02  | Acme Inc.    |
|   2 | 2022-02-01  | NULL        | Globex Corp. |
|   3 | 2022-03-01  | 2022-03-02  | Initech Ltd. |
+-----+-------------+-------------+--------------+
3 rows in set (0.35 sec)
```

8 Conclusion

Thus, we have learned to Select Group By, Joins and Subqueries commands thoroughly.

9 FAQ

1. **When to use self join? How does it differ from other joins?**

   A self join is used when you need to join a table with itself, typically to find relationships between rows in the same table. It differs from other joins in that you are joining a table with itself rather than joining two separate tables. A self join can be performed using an alias to distinguish between the two copies of the table being joined.

```sql
SELECT t1.employee_name, t2.employee_name
FROM employees t1
JOIN employees t2 ON t1.manager_id = t2.employee_id;
```

2. **Compare Cross Join with Natural Join. Share your comments.**

   A cross join produces the Cartesian product of two tables, resulting in a combination of all rows from one table with all rows from another table. A natural join matches two tables based on their common column names. It automatically eliminates duplicate columns from the result set, and the result set only contains the columns with the same name from both tables.

```sql
  SELECT *
  FROM table1
  CROSS JOIN table2;
```

3. **What is the importance of SQL joins in database management? Explain its types.**

   SQL joins are important in database management because they allow you to combine data from two or more tables into a single result set. This allows you to extract meaningful information from your data by revealing relationships between tables. There are four main types of SQL joins: inner join, left join, right join, and full outer join.

4. **What are the different types of Joins in SQL?**

   The different types of SQL joins are:

   - Inner join: returns only the matching rows from both tables based on the specified join condition.
   - Left join: returns all the rows from the left table and the matching rows from the right table based on the specified join condition.
   - Right join: returns all the rows from the right table and the matching rows from the left table based on the specified join condition.
   - Full outer join: returns all the rows from both tables, matching rows where possible and filling in NULL values for non-matching rows.

```sql
SELECT *
FROM table1
INNER JOIN table2
ON table1.column = table2.column;

```

```sql
SELECT *
FROM table1
LEFT JOIN table2
ON table1.column = table2.column;
```

5. **State the difference between inner join and left join.**

   The main difference between an inner join and a left join is that an inner join only returns matching rows from both tables based on the specified join condition, while a left join returns all the rows from the left table and the matching rows from the right table based on the specified join condition.

```sql
SELECT *
FROM table1
LEFT JOIN table2
ON table1.column = table2.column;
```

```sql
SELECT *
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
```

6. **State difference between left join and right join.**

   The main difference between a left join and a right join is that a left join returns all the rows from the left table and the matching rows from the right table based on the specified join condition, while a right join returns all the rows from the right table and the matching rows from the left table based on the specified join condition.

```sql
SELECT *
FROM table1
LEFT JOIN table2
ON table1.column = table2.column;
```

```sql
SELECT *
FROM table1
RIGHT JOIN table2
ON table1.column = table2.column;

```