HOTEL CHAIN

MYSQL QUERIES ON NORMALIZATION PROJECT

GROUP 16

Nimra Ashraf & Noor Fatima

110829 & 110830

MORNING

Submitted to:

Mam Sehrish Khan

TABLE OF CONTENTS

1. DESCRIPTION
2. ENTITY RELATIONSHIP DIAGRAM
3. RELATIONAL SCHEMA
4. SQL QUERIES

- SHOW DATABASES
- CREATING DATABASE & TABLES
- USING DATABASE
- DESCRIBE TABLES
- SELECT
- INSERT INTO
- ALTER TABLES
- TABLE UPDATION
- TABLE CONSTRAINTS
- ARITHMETIC OPERATIONS
- RELATIONAL OPERATIONS
- AGGREGATE FUNCTIONS
- ORDER BY CLAUSE
- GROUP BY CLAUSE
- TABLE ALIAS
- WHERE CLAUSE
- DISTINCT COMMAND
- BETWEEN COMMAND
- COUNT
- HAVING
- AND & OR
- IN
- LIKE
- IS NULL & IS NOT NULL
- DELETE
- DROP,
- TRUNCATE
- GRANT
- PRIVILEGE

5. NORMALIZATION TABLES
6. PRACTICAL IMPLEMENTATION OF QUERIES

DESCRIPTION

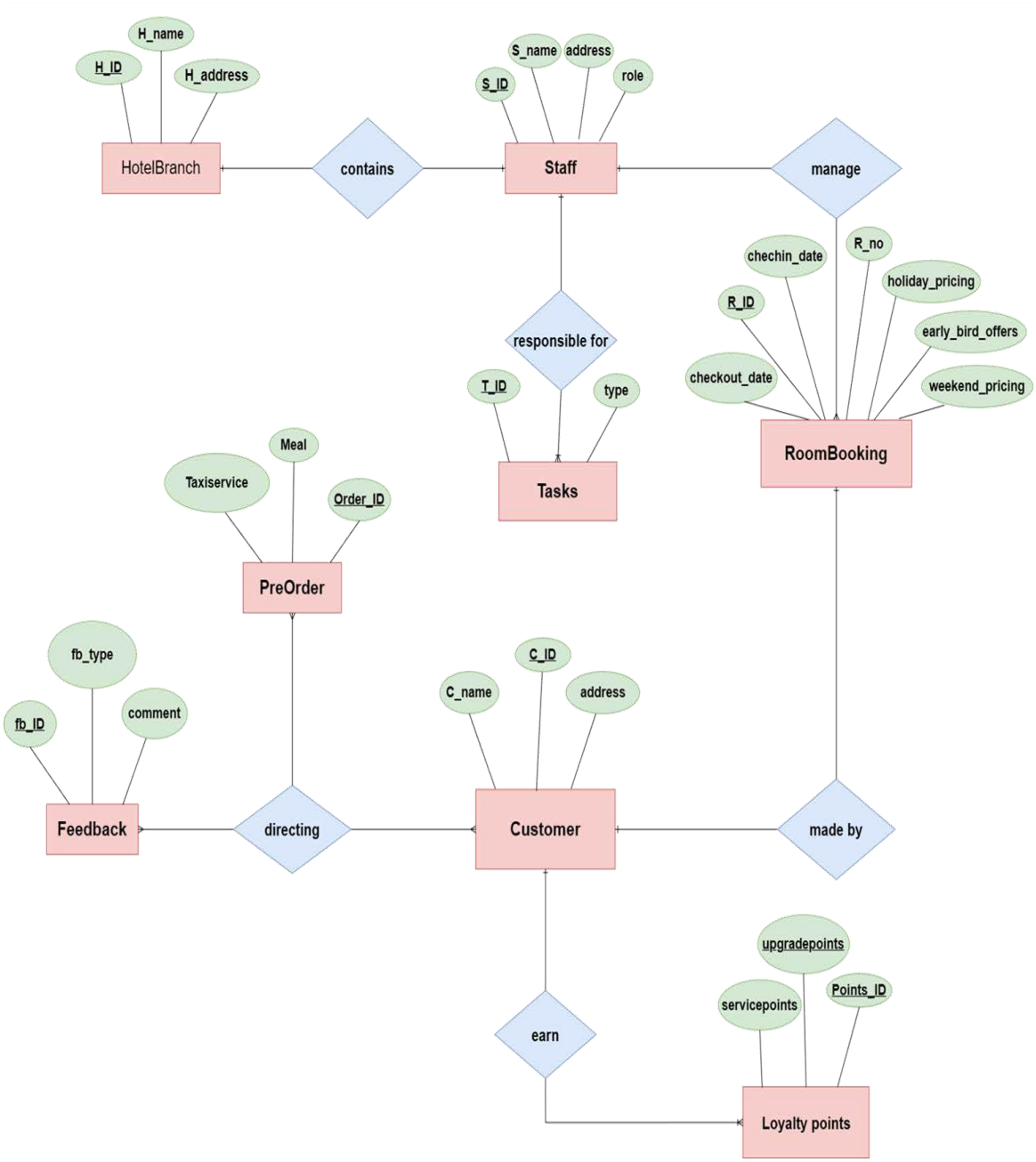
SQL stands for structure query language. It is a widely used, open source relational database management system.

A hotel chain management system provides a centralized platform for managing multiple hotel properties within a chain. This database would store information about different hotels, including room details, staff managing, reservations, guest records pricing information etc.

Pros:

- Scalability
- Reliability
- Cost effectiveness
- Flexibility

ENTITY RELATIONSHIP DIAGRAM



RELATIONAL SCHEMA

1. **hotel_branch** (h_id, h_name, h_address, staff_id)
2. **staff** (s_id, s_name, s_address, s_role)
3. **room_booking** (R_id, R_no, check_in_date, check_out_date, weekend_pricing, holiday_pricing, early_bird_offers, s_id)
4. **customer** (c_id, c_name, c_address, r_id)
5. **loyalty_points** (points_id, upgrade_points, service_points, cus_id)
6. **feedback** (f_id, f_type, comment, customer_id)
7. **pre_order** (order_id, meal, taxi_service, cust_id)
8. **task** (t_id, type, s_id)

SQL QUERIES

Show DATABASES

Show all databases and tables placed in a database MySQL use the following command:

⇒ **show databases;**
⇒ **show tables;**

CREATE DATABASE & TABLES

Create a new database or table.

⇒ **create database;**
⇒ **create table table_name (attribute datatype (size), …);**

USING DATABASE

Use a database already saved in MySQL.

⇒ **use database database_name;**

DESCRIBE TABLES

To see the constraints we have assigned to tables, DESCRIBE keyword is used.

⇒ **DESCRIBE table_name;**

SELECT

Used to retrieve rows selected from one or more tables.

⇒ **Select * from table_name;**
⇒ **Select attribute1_name from table_name where attribute2_name= '---';**

INSERT INTO

Insert values in tables.

⇒ **INSERT INTO table_name values ('attribute' datatype (value), ….);**

ALTER TABLES

Used for many purposes such as:

1. **To rename a column of a table**

⇒ **ALTER table table_name RENAME COLUMN column_name from existing_name to new_name;**

2. **To add a new column in a table**

⇒ **ALTER table table_name ADD column datatype (size);**

3. **To make an attribute foreign key**

⇒ **ALTER table table_name ADD column_name FOREIGN KEY REFERENCES referencetable_name (P.K);**

TABLE UPDATION

To reset the values of attributes in a table.

⇒ **Update table_name set attribute='value' WHERE attribute_PK='target-value';**

TABLE CONTRAINTS

There are two constraints for tables, Primary key and Foreign key.

Primary Key:

The attribute of a table on which all the other attributes of that depend.

Foreign Key:

When Primary key of a table is used in another table, it becomes Foreign key.

⇒ **ALTER table table_name ADD FOREIGN KEY (key_name) REFERENCES reference_table (P.K);**

Arithmetic Operations

Arithmetic operations include operators such as +, -, *, / etc.

select attribute operator value from table;

Logical / Relational Operations

Logical operations include operators such as >, <, <=, >=, !=, == etc. to relate any two attributes of a table.

⇒ **select attribute1 from table_name where attribute2 relational operator value;**

Aggregation Functions

Aggregate functions include avg, max, min etc. operations.

⇒ **select aggregate_function (attribute) from table;**

ORDER BY

ORDER BY keyword is used to sort the values of tables in ascending or descending order. By default ascending order is

⇒ **select * from table ORDER BY attribute DESC;**
⇒ **select * from table ORDER BY attribute ASC;**

GROUP BY

GROUP BY clause is important used to group rows from a table based on the values of one or more column Used with aggregate functions like **AVG, MAX, MIN, SUM and COUNT** to perform calculations on grouped data.

Syntax:

⇒ **select aggreagate_function (attribute) from table GROUP BY attributes;**

AS

It allows for temporary renaming with a query, which can simplify complex queries and result sets.

Where

This clause is used to filter records. It is used to extract only those records that fulfill a specified condition.

DISTINCT

The DISTINCT keyword in MySQL is used to remove duplicate records from the results of a SELECT query

Syntax:

⇒ **select DISTINCT attribute from table;**

BETWEEN

The BETWEEN clause is used to show the values/contents of the table between a given limit. It filter the result set within a specified range.

Syntax:

⇒ **select column_name from table where column_name BETWEEN value1 AND value2;**

Count & Count (*)

This function counts all rows in a table regardless of whether they contain NULL values.

Syntax:

⇒ **select count (*) attribute_name from table;**
⇒ **select count (attribute_name) from table;**

HAVING

The HAVING clause is similar to the WHERE clause but is specifically applied after grouping and aggregation.

Syntax:

⇒ **select aggreagate_function (attribute) from table GROUP BY attributes HAVING count (attribute) >1;**

AND & OR

The AND & OR operators are used to filter records based on more than one conditions:

- The AND operator displays a record if all the conditions separated by AND are true.
- The AND operator displays a record if any of the conditions separated by AND are true.

IN

The IN operator allows you to specify multiple values in a where clause. It is a shorthand for multiple OR conditions.

⇒ **SELECT attribute_name(s) FROM table_name**
WHERE attribute_name IN (value1, value2, ...);

LIKE

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

Syntax:

⇒ **SELECT column1. . .FROM table_name**
WHERE columnN LIKE pattern;

1. The percent sign (%) represents zero, one, or multiple characters
2. The underscore sign (_) represents one, single character

IS NULL & IS NOT NULL

These keywords are used for checking that the values of attributes checked

NULL or not.

Syntax:

⇒ **select attribute_name from table where attribute IS NULL;**
⇒ **select attribute_name from table where attribute IS NOT NULL;**

JOINS

Joins allows to retrieve related data from multiple tables in a single query, avoiding the need for different separate queries.

Syntax:

⇒ **Select column_list from table1 JOIN table2 ON table1. column = table1;**

INNER JOIN:

The joins in which both the tables have matching values in them are called inner join.

Left Join:

This join return all rows from the left table and matching rows in right table.

Cross join:

A cross join is type of join that return cartesian product of rows from the tables in the join.

Equi join:

It is join operation in sql that combines two table based on a matching column between them.

Right Join:

This join return all rows from right table and matching rows from left table.

VIEWS

A MySQL view is a predefined select query that operates on existing data without duplicating it. A view acts as a virtual table.

Syntax:

⇒ **create or replace VIEW view_name AS select column1, column2 from table_name;**

DELETE

DELETE statement is used to delete rows in a table. It deletes a specific row using where clause.

Syntax:

⇒ **delete from table where column_name= 'value';**

DROP

DROP statement is used to delete the whole table along with table structure, attribute and indexes.

Syntax:

⇒ **drop table table_name;**

TRUNCATE

The truncate statement is used to delete all data in the table not the whole table.

Syntax:

⇒ **truncate table_name;**

Sub Query

Subqueries are also known as inner queries or nested queries.

Syntax:

⇒ **Select column1, column2… from table where column operator (select column from another_table where condition); column1, column2, …: The columns you want to retrieve**.

GRANT

Grant is a statement used to assign privileges to user accounts, allowing them to perform specific actions on database projects.

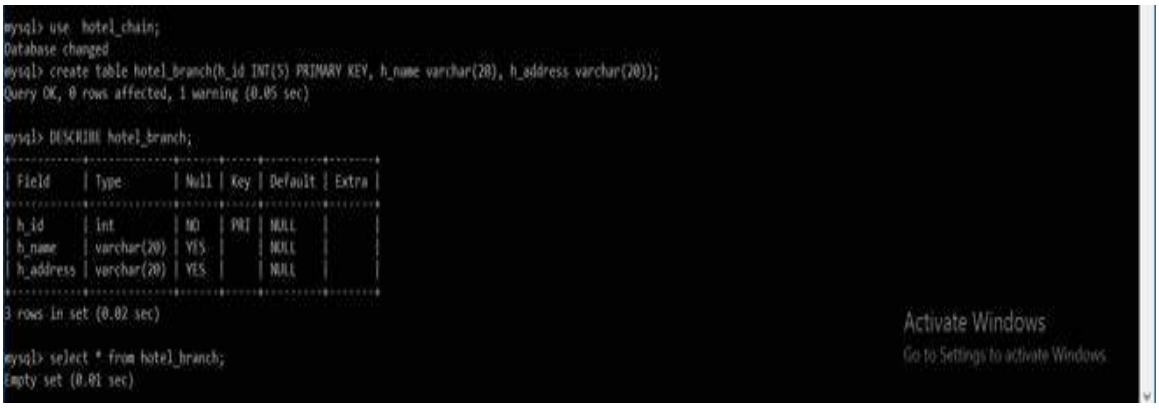
PRIVILEGES

Privileges are the rights or permissions assigned to users that determine what actions they can perform on the database.

NORMALIZATION TABLES

CREATE DATABASE HOTEL_CHAIN

CREATE Hotel_Branch Table



INSERTING VALUES IN HOTEL_BRANCH

3 NF OF HOTEL_BRANCH

```
mysql> select h_name , h_address from Hotel_branch;
+-----------+------------+
| h_name    | h_address  |
+-----------+------------+
| unique    | lahore     |
| five_star | Multan     |
| Super     | Sheikhpura |
+-----------+------------+
3 rows in set (0.00 sec)
```

CREATE STAFF TABLE

```
mysql> create table Staff(s_id INT(5) PRIMARY KEY, s_name varchar(20),s_address varchar(20),s_role varchar(19),R_id INT(5));
Query OK, 0 rows affected, 2 warnings (0.04 sec)

mysql> DESCRIBE staff;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| s_id      | int         | NO   | PRI | NULL    |       |
| s_name    | varchar(20) | YES  |     | NULL    |       |
| s_address | varchar(20) | YES  |     | NULL    |       |
| s_role    | varchar(19) | YES  |     | NULL    |       |
| R_id      | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

INSERTING VALUES IN STAFF

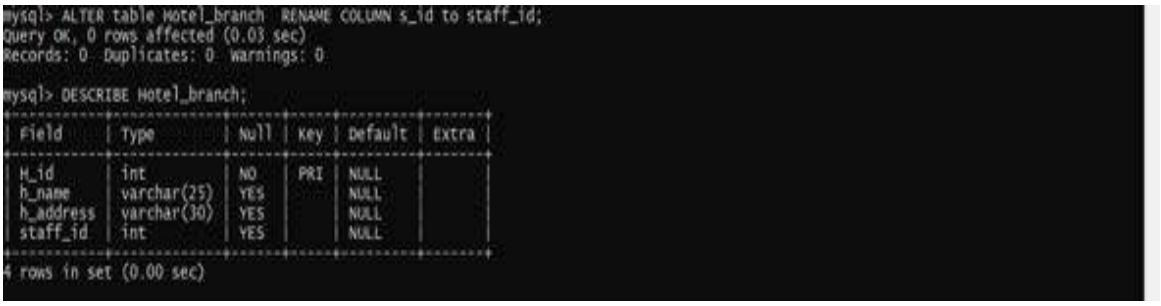
```
mysql> INSERT INTO staff values('1','haram','multan','manager','10'),('2','ahmad','sheikhpura','sweeper','20'),('3','sohail','lahore','
receptionist','30');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from staff;
+------+--------+------------+--------------+------+
| s_id | s_name | s_address  | s_role       | R_id |
+------+--------+------------+--------------+------+
|    1 | haram  | multan     | manager      |   10 |
|    2 | ahmad  | sheikhpura | sweeper      |   20 |
|    3 | sohail | lahore     | receptionist |   30 |
+------+--------+------------+--------------+------+
3 rows in set (0.00 sec)
```

3RD NORMAL FORM OF STAFF
ADD s_id COLUMN IN HOTEL BRANCH

```
mysql> select s_name, s_address, s
    -> ^C
mysql> select s_name,s_address,s_role from staff;
+--------+-----------+--------------+
| s
+--mysql> ALTER table hotel_branch ADD s_id INT(5);
| Query OK, 0 rows affected, 1 warning (0.03 sec)
| Records: 0  Duplicates: 0  Warnings: 1
|
| 5mysql> DESCRIBE Hotel_branch;
+---+-----------+-------------+------+-----+---------+-------+
3 r| Field     | Type        | Null | Key | Default | Extra |
  +-----------+-------------+------+-----+---------+-------+
  | H_id      | int         | NO   | PRI | NULL    |       |
  | h_name    | varchar(25) | YES  |     | NULL    |       |
  | h_address | varchar(30) | YES  |     | NULL    |       |
  | s_id      | int         | YES  |     | NULL    |       |
  +-----------+-------------+------+-----+---------+-------+
  4 rows in set (0.00 sec)
```

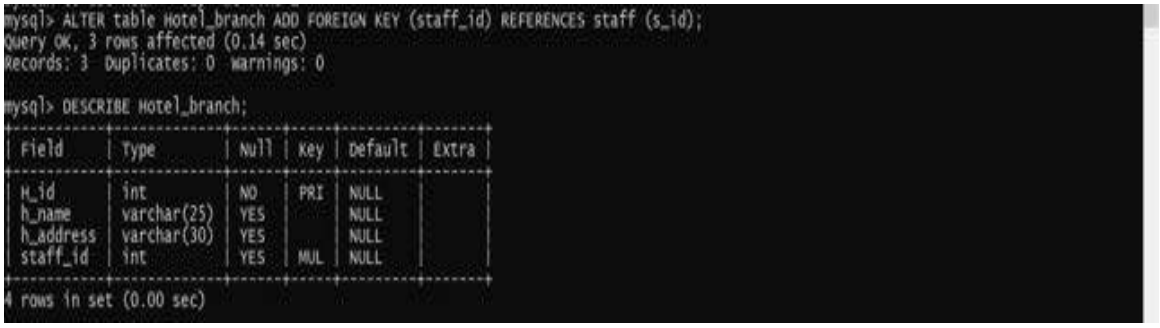
TABLE

CHANGE COLUMN NAME (FROM s_id TO staff_id)



```
mysql> ALTER table Hotel_branch  RENAME COLUMN s_id to staff_id;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Hotel_branch;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| H_id      | int         | NO   | PRI | NULL    |       |
| h_name    | varchar(25) | YES  |     | NULL    |       |
| h_address | varchar(30) | YES  |     | NULL    |       |
| staff_id  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

MAKE staff_id FOREIGN KEY



```
mysql> ALTER table Hotel_branch ADD FOREIGN KEY (staff_id) REFERENCES staff (s_id);
Query OK, 3 rows affected (0.14 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Hotel_branch;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| H_id      | int         | NO   | PRI | NULL    |       |
| h_name    | varchar(25) | YES  |     | NULL    |       |
| h_address | varchar(30) | YES  |     | NULL    |       |
| staff_id  | int         | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

INSERTING VALUES IN FOREIGN KEY

```
mysql> Update Hotel_branch set staff_id='1' where H_id='1001';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Update Hotel_branch set staff_id='2' where H_id='1002';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Update Hotel_branch set staff_id='3' where H_id='1003';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

SELECT FROM HOTEL_BRANCH

```
mysql> select * from Hotel_branch;
+------+-----------+------------+----------+
| H_id | h_name    | h_address  | staff_id |
+------+-----------+------------+----------+
| 1001 | unique    | lahore     |        1 |
| 1002 | five_star | Multan     |        2 |
| 1003 | Super     | Sheikhpura |        3 |
+------+-----------+------------+----------+
3 rows in set (0.00 sec)
```

CREATE ROOM_BOOKING TABLE

```
mysql> create table  Room_booking(room_id INT(5) PRIMARY KEY, r_no INT(6), check_in_date INT(20), check_out_date INT(20), weekend_prici
ng INT(20), hoilday_pricing INT(19), early_bird_offers INT(25));
Query OK, 0 rows affected, 7 warnings (0.04 sec)

mysql> DESCRIBE Room_booking;
+-------------------+------+------+-----+---------+-------+
| Field             | Type | Null | Key | Default | Extra |
+-------------------+------+------+-----+---------+-------+
| room_id           | int  | NO   | PRI | NULL    |       |
| r_no              | int  | YES  |     | NULL    |       |
| check_in_date     | int  | YES  |     | NULL    |       |
| check_out_date    | int  | YES  |     | NULL    |       |
| weekend_pricing   | int  | YES  |     | NULL    |       |
| hoilday_pricing   | int  | YES  |     | NULL    |       |
| early_bird_offers | int  | YES  |     | NULL    |       |
+-------------------+------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

MODIFY CHECK_IN & CHECK_OUT DATES

```
mysql> ALTER TABLE Room_booking
    -> MODIFY COLUMN check_in_date DATE;
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER table Room_booking MODIFY COLUMN check_in_date DATE;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER table Room_booking MODIFY COLUMN check_out_date DATE;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

DESCRIBE ROOM_BOOKING

```
mysql> DESCRIBE Room_booking;
+-------------------+------+------+-----+---------+-------+
| Field             | Type | Null | Key | Default | Extra |
+-------------------+------+------+-----+---------+-------+
| room_id           | int  | NO   | PRI | NULL    |       |
| r_no              | int  | YES  |     | NULL    |       |
| check_in_date     | date | YES  |     | NULL    |       |
| check_out_date    | date | YES  |     | NULL    |       |
| weekend_pricing   | int  | YES  |     | NULL    |       |
| hoilday_pricing   | int  | YES  |     | NULL    |       |
| early_bird_offers | int  | YES  |     | NULL    |       |
+-------------------+------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

MODIFY EARLY_BIRD_OFFERS & INSERTING VALUES

```
mysql> ALTER table Room_booking MODIFY COLUMN early_bird_offers varchar(29);
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO Room_booking values('10','202','2025-01-05','2025-01-25','8000','9000','40%');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Room_booking values('20','504','2025-02-4','2025-02-8','7000','8000','30%');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Room_booking values('30','301','2025-07-4','2025-07-8','8000','12000','30%');
Query OK, 1 row affected (0.01 sec)
```

SELECT FROM ROOM_BOOKING

```
mysql> select * from Room_booking;
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+
| room_id | r_no | check_in_date | check_out_date | weekend_pricing | hoilday_pricing | early_bird_offers |
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+
|      10 |  202 | 2025-01-05    | 2025-01-25     |            8000 |            9000 | 40%               |
|      20 |  504 | 2025-02-04    | 2025-02-08     |            7000 |            8000 | 30%               |
|      30 |  301 | 2025-07-04    | 2025-07-08     |            8000 |           12000 | 30%               |
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+
3 rows in set (0.00 sec)
```

ADD STAFF_ID FOREIGN KEY

```
mysql> ALTER table room_booking ADD FOREIGN KEY (STAFF_ID) REFERENCES staff(s_id);
Query OK, 3 rows affected (0.12 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DESCRIBE room_booking;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| room_id         | int         | NO   | PRI | NULL    |       |
| r_no            | int         | YES  |     | NULL    |       |
| check_in_date   | date        | YES  |     | NULL    |       |
| check_out_date  | date        | YES  |     | NULL    |       |
| weekend_pricing | int         | YES  |     | NULL    |       |
| hoilday_pricing | int         | YES  |     | NULL    |       |
| early_bird_offers | varchar(20) | YES  |     | NULL    |       |
| STAFF_ID        | int         | YES  | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

INSERT VALUES IN FOREIGN KEY

```
mysql> Update room_booking set STAFF_ID='1' where room_id='10';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Update room_booking set STAFF_ID='2' where room_id='20';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Update room_booking set STAFF_ID='3' where room_id='30';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

SELECT FROM ROOM_BOOKING

```
mysql> select * from room_booking;
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+----------+
| room_id | r_no | check_in_date | check_out_date | weekend_pricing | hoilday_pricing | early_bird_offers | STAFF_ID |
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+----------+
|      10 |  202 | 2025-01-05    | 2025-01-25     |            8000 |            9000 | 40%               |        1 |
|      20 |  504 | 2025-02-04    | 2025-02-08     |            7000 |            8000 | 30%               |        2 |
|      30 |  301 | 2025-07-04    | 2025-07-08     |            8000 |           12000 | 30%               |        3 |
+---------+------+---------------+----------------+-----------------+-----------------+-------------------+----------+
3 rows in set (0.00 sec)
```

3 NF OF ROOM_BOOKING

```
mysql> select r_no,weekend_pricing,holiday_pricing,early_bird_offers from Room_booking;
ERROR 1054 (42S22): Unknown column 'holiday_pricing' in 'field list'
mysql> select r_no,weekend_pricing,hoilday_pricing,early_bird_offers from Room_booking;
+------+-----------------+-----------------+-------------------+
| r_no | weekend_pricing | hoilday_pricing | early_bird_offers |
+------+-----------------+-----------------+-------------------+
|  202 |            8000 |            9000 | 40%               |
|  504 |            7000 |            8000 | 30%               |
|  301 |            8000 |           12000 | 30%               |
+------+-----------------+-----------------+-------------------+
3 rows in set (0.00 sec)
```

MAKE ROOM_ID FOREIGN KEY IN STAFF

```
mysql> ALTER table staff ADD FOREIGN KEY (R_id) REFERENCES Room_booking (room_id);
Query OK, 3 rows affected (0.14 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> DESCRIBE staff;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| s_id      | int         | NO   | PRI | NULL    |       |
| s_name    | varchar(20) | YES  |     | NULL    |       |
| s_address | varchar(20) | YES  |     | NULL    |       |
| s_role    | varchar(10) | YES  |     | NULL    |       |
| R_id      | int         | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

CREATE CUSTOMER TABLE

```
mysql>  create table customer(c_id INT(9) primary key,c_name varchar(15), r_id INT(5));
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> DESCRIBE customer;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| c_id   | int         | NO   | PRI | NULL    |       |
| c_name | varchar(15) | YES  |     | NULL    |       |
| r_id   | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

ADD MISING C_ADDRESS COLUMN IN CUSTOMER TABLE

```
mysql> ALTER  table customer ADD c_address varchar(30);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE customer;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| c_id      | int         | NO   | PRI | NULL    |       |
| c_name    | varchar(15) | YES  |     | NULL    |       |
| r_id      | int         | YES  |     | NULL    |       |
| c_address | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

INSERTING VALUES

```
mysql> INSERT INTO customer values('100','NOOR', '10', 'Farooqabad');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer values('200','NIMRA', '20', 'sheikhpura'),('300','shanzey', '30', 'sheikhpura');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from student;
ERROR 1146 (42S02): Table 'hotel_chain.student' doesn't exist
mysql> select * from customer;
+------+--------+------+-----------+
| c_id | c_name | r_id | c_address |
+------+--------+------+-----------+
|  100 | NOOR   |   10 | Farooqabad |
|  200 | NIMRA  |   20 | sheikhpura |
|  300 | shanzey |  30 | sheikhpura |
+------+--------+------+-----------+
3 rows in set (0.00 sec)
```

MAKE r_id FOREIGN KEY IN CUSTOMER TABLE

```
mysql> ALTER table customer ADD FOREIGN KEY (r_id) REFERENCES Room_booking (room_id);
Query OK, 3 rows affected (0.22 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> describe customer;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| c_id      | int         | NO   | PRI | NULL    |       |
| c_name    | varchar(15) | YES  |     | NULL    |       |
| r_id      | int         | YES  | MUL | NULL    |       |
| c_address | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

CREATE LOYALTY POINTS TABLE

```
mysql> create table loyalty_points(point_id INT(10) PRIMARY KEY, service_points INT(8), upgrade_points INT(6),cus_id INT(9));
Query OK, 0 rows affected, 4 warnings (0.04 sec)

mysql> DESCRIBE loyalty_points;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| point_id       | int  | NO   | PRI | NULL    |       |
| service_points | int  | YES  |     | NULL    |       |
| upgrade_points | int  | YES  |     | NULL    |       |
| cus_id         | int  | YES  |     | NULL    |       |
+----------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

MAKE cus_id FOREIGN KEY

```
mysql> ALTER table loyalty_points ADD FOREIGN KEY (cus_id) REFERENCES customer (c_id);
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe loyalty_points;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| point_id       | int  | NO   | PRI | NULL    |       |
| service_points | int  | YES  |     | NULL    |       |
| upgrade_points | int  | YES  |     | NULL    |       |
| cus_id         | int  | YES  | MUL | NULL    |       |
+----------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

INSERTING VALUES & DESCRIBE

```
mysql> INSERT INTO loyalty_points values('104','300','50','100');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO loyalty_points values('208','500','100','200'),('310','400','60','300');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from loyalty_points;
+----------+----------------+----------------+--------+
| point_id | service_points | upgrade_points | cus_id |
+----------+----------------+----------------+--------+
|      104 |            300 |             50 |    100 |
|      208 |            500 |            100 |    200 |
|      310 |            400 |             60 |    300 |
+----------+----------------+----------------+--------+
3 rows in set (0.00 sec)
```

CREATE FEEDBACK TABLE

```
mysql> create table feedback(fb_id INT(15) PRIMARY KEY, fb_type varchar(20), comment varchar(30), customer_id INT(10));
Query OK, 0 rows affected, 2 warnings (0.04 sec)

mysql> DESCRIBE feedback;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| fb_id       | int         | NO   | PRI | NULL    |       |
| fb_type     | varchar(20) | YES  |     | NULL    |       |
| comment     | varchar(30) | YES  |     | NULL    |       |
| customer_id | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

MAKE customer_id FOREIGN KEY IN FEEDBACK TABLE

```
mysql> INSERT INTO feedback values('203','compliment','great food','100');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO feedback values('202','complaint','late services','200');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO feedback values('300','compliment','good services','300');
Query OK, 1 row affected (0.01 sec)

mysql> select * from feedback;
+-------+-----------+---------------+-------------+
| fb_id | fb_type   | comment       | customer_id |
+-------+-----------+---------------+-------------+
|   202 | complaint | late services |         200 |
|   203 | compliment | great food   |         100 |
|   300 | compliment | good services |         300 |
+-------+-----------+---------------+-------------+
3 rows in set (0.00 sec)
```

INSERTING VALUES

```
mysql> ALTER table feedback ADD FOREIGN KEY (customer_id) REFERENCES customer (c_id);
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe feedback;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| fb_id       | int         | NO   | PRI | NULL    |       |
| fb_type     | varchar(20) | YES  |     | NULL    |       |
| comment     | varchar(30) | YES  |     | NULL    |       |
| customer_id | int         | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

CREATE PRE_ORDER TABLE

```
mysql> create table PreOrder(order_id INT(30) PRIMARY KEY,meal varchar(40),taxi_services varchar(10),cust_id INT(10));
Query OK, 0 rows affected, 2 warnings (0.04 sec)

mysql> DESCRIBE PreOrder;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| order_id      | int         | NO   | PRI | NULL    |       |
| meal          | varchar(40) | YES  |     | NULL    |       |
| taxi_services | varchar(10) | YES  |     | NULL    |       |
| cust_id       | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

MADE cus_id FOREIGN KEY

```
mysql> ALTER table PreOrder ADD FOREIGN KEY (cust_id) REFERENCES customer (c_id);
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe PreOrder;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| order_id      | int         | NO   | PRI | NULL    |       |
| meal          | varchar(40) | YES  |     | NULL    |       |
| taxi_services | varchar(10) | YES  |     | NULL    |       |
| cust_id       | int         | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

INSERTING VALUES IN ORDER

```
mysql> INSERT INTO PreOrder values('110','breakfast','yes','100');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PreOrder values('220','Dinner','No','200');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PreOrder values('330','lunch','yes','300');
Query OK, 1 row affected (0.01 sec)

mysql> select * from PreOrder;
+----------+-----------+---------------+---------+
| order_id | meal      | taxi_services | cust_id |
+----------+-----------+---------------+---------+
|      110 | breakfast | yes           |     100 |
|      220 | Dinner    | No            |     200 |
|      330 | lunch     | yes           |     300 |
+----------+-----------+---------------+---------+
3 rows in set (0.00 sec)
```

CREATE TASK TABLE

```
mysql> create table task(t_id INT(5) PRIMARY KEY, type varchar(20), st_id INT(5));
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> DESCRIBE tsk;
ERROR 1146 (42S02): Table 'hotel.thale.tsk' doesn't exist
mysql> DESCRIBE task;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| t_id  | int         | NO   | PRI | NULL    |       |
| type  | varchar(20) | YES  |     | NULL    |       |
| st_id | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

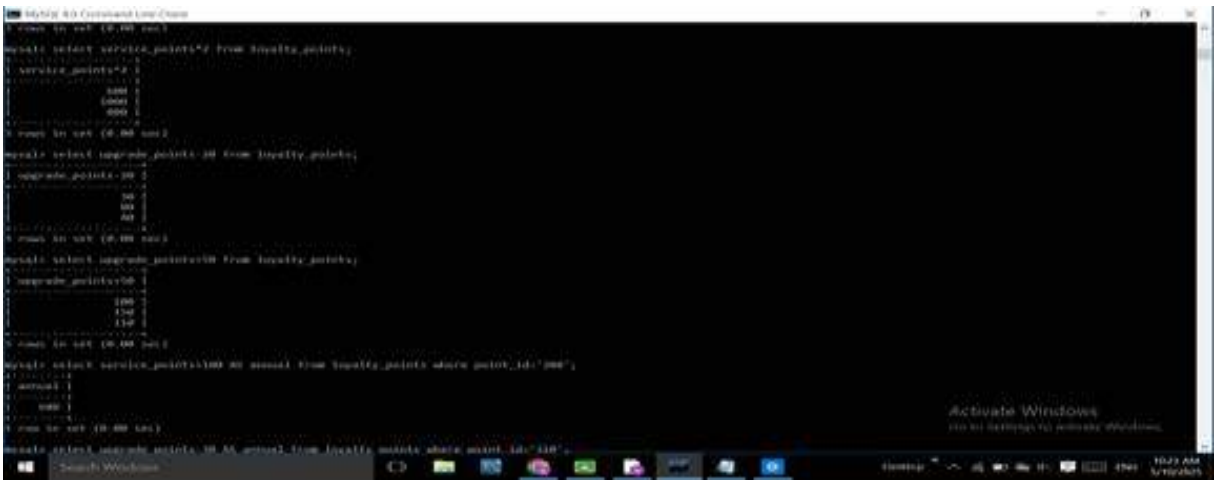
ADD st_id FOREIGN KEY AND INSERTION

```
mysql> ALTER table task ADD FOREIGN KEY (st_id) REFERENCES staff(s_id);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE task;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| t_id  | int         | NO   | PRI | NULL    |       |
| type  | varchar(20) | YES  |     | NULL    |       |
| st_id | int         | YES  | MUL | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

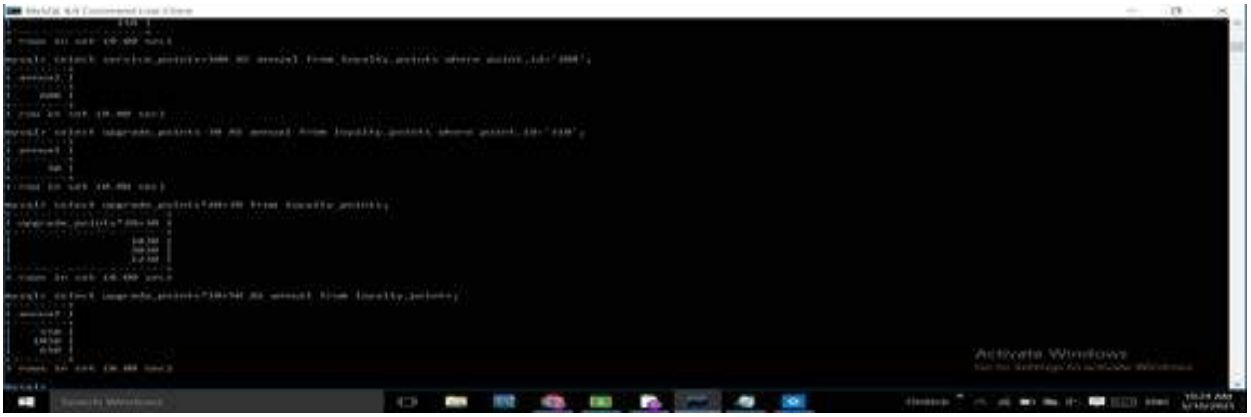
```
mysql> select * from task;
+------+-------------+-------+
| t_id | type        | st_id |
+------+-------------+-------+
|  500 | cleaning    |     1 |
|  201 | maintenance |     2 |
|  405 | guard       |     3 |
+------+-------------+-------+
```

PRACTICAL IMPLEMENTATION

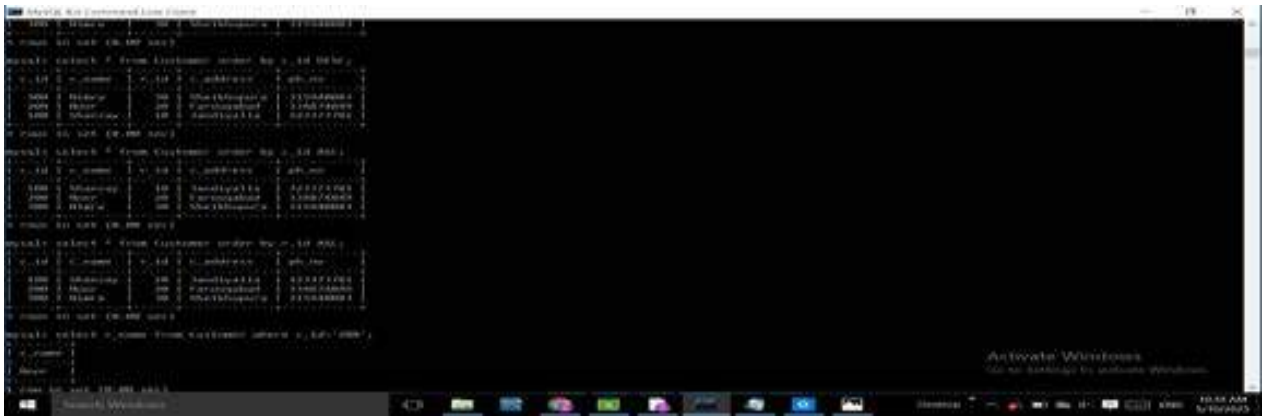
ARITHMETIC & LOGICAL OPERATIONS ON LOYALTY_POINTS TABLE

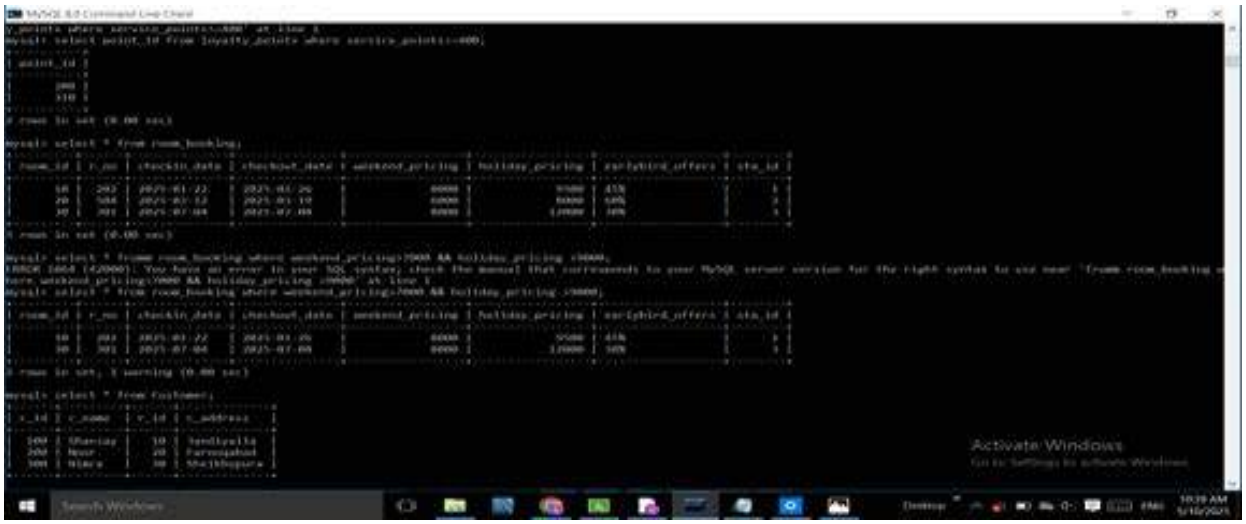


LOGICAL OPERATIONS
ADDING ph_no COLUMN IN CUSTOMER

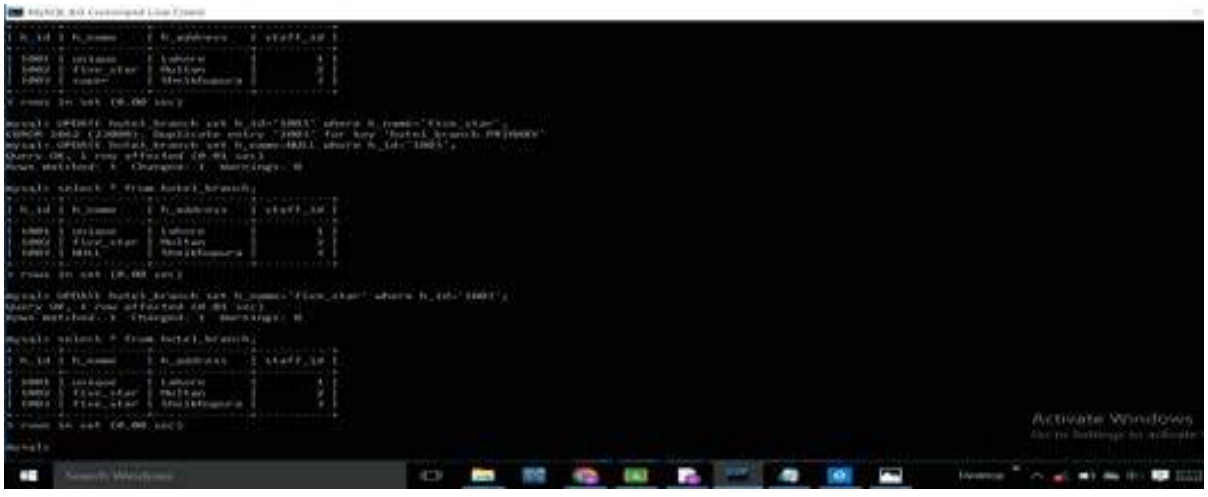


ORDER BY CLAUSE ON CUSTOMER

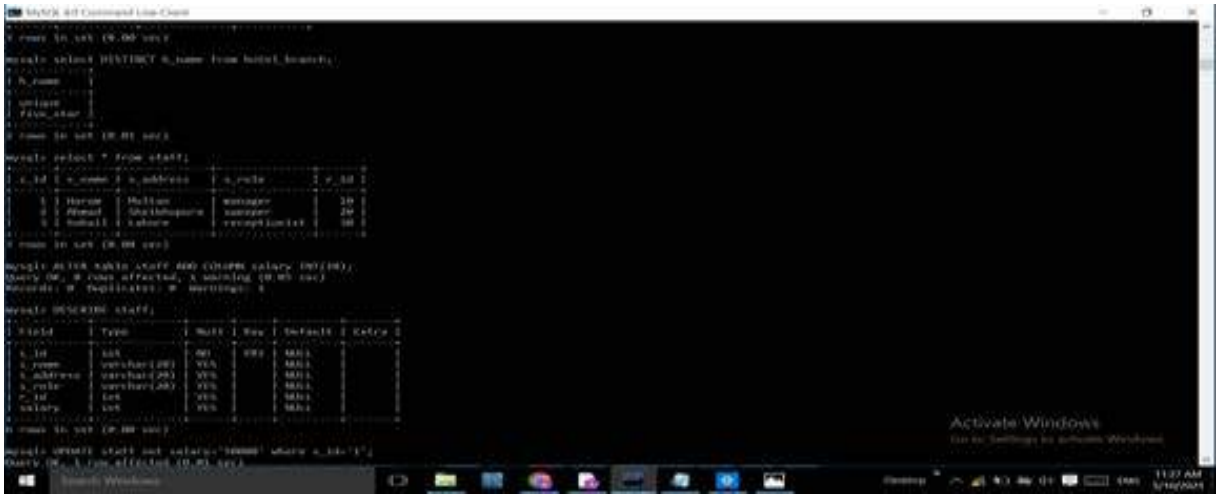




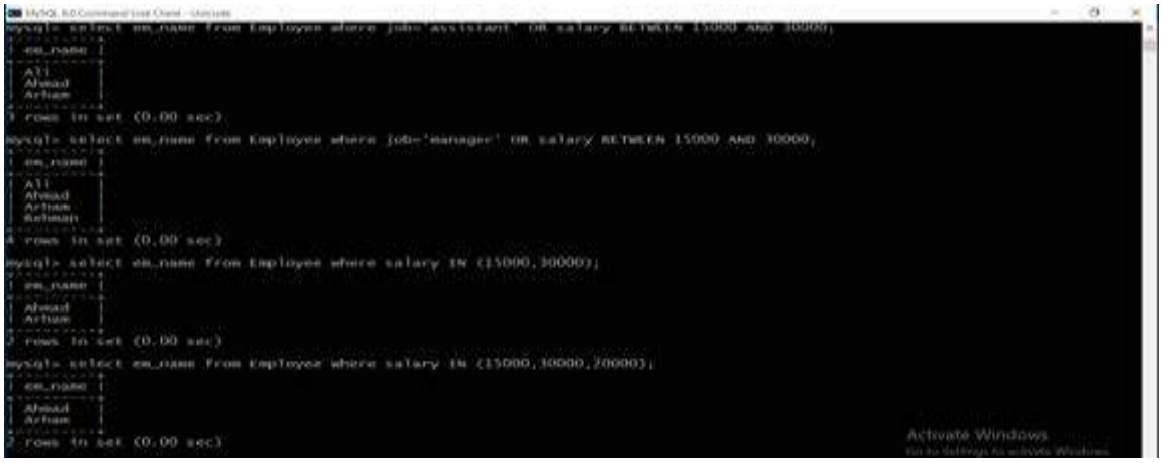
GIVING TWO h_ids SAME NAME

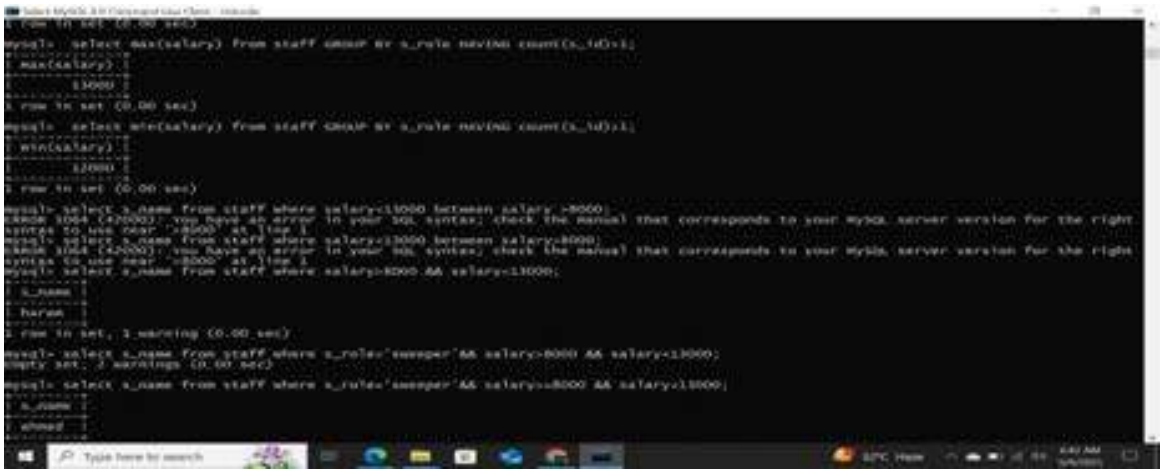


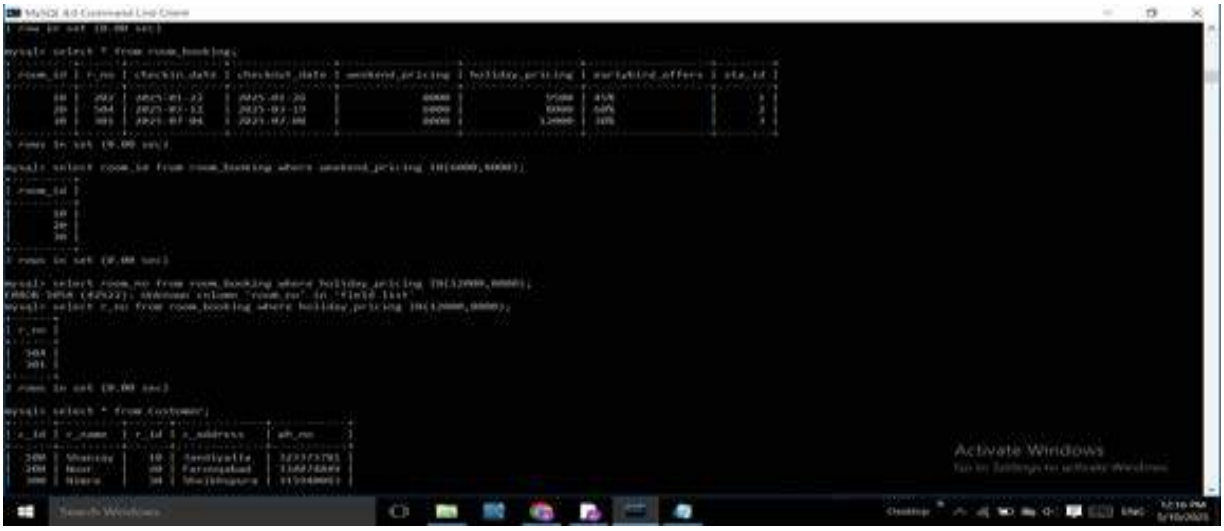
DISTINCT CLAUSE AND ADDING SALARY COLUMN IN STAFF TABLE



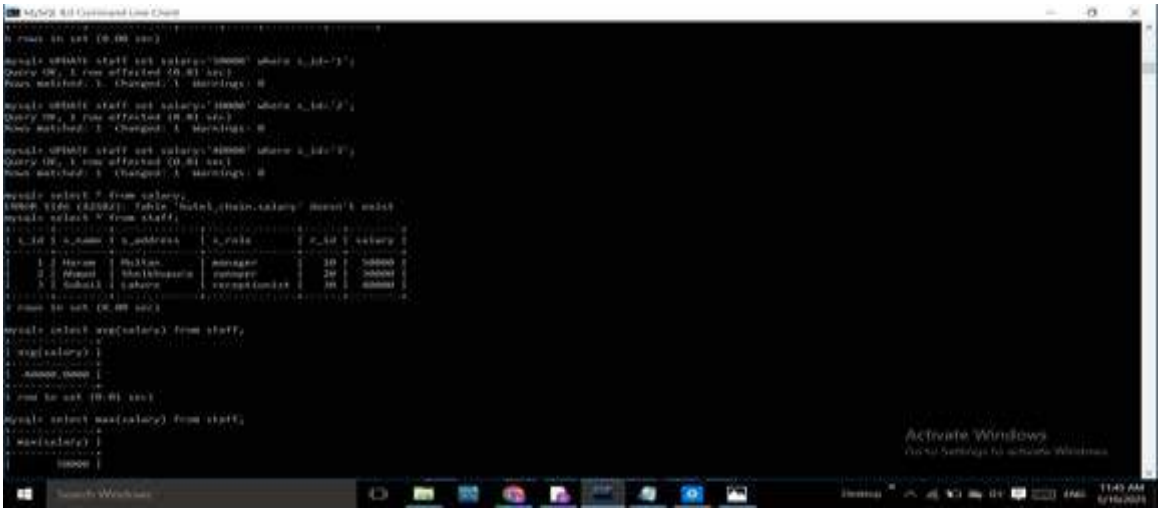
BETWEEN, HAVING AND IN CLAUSES





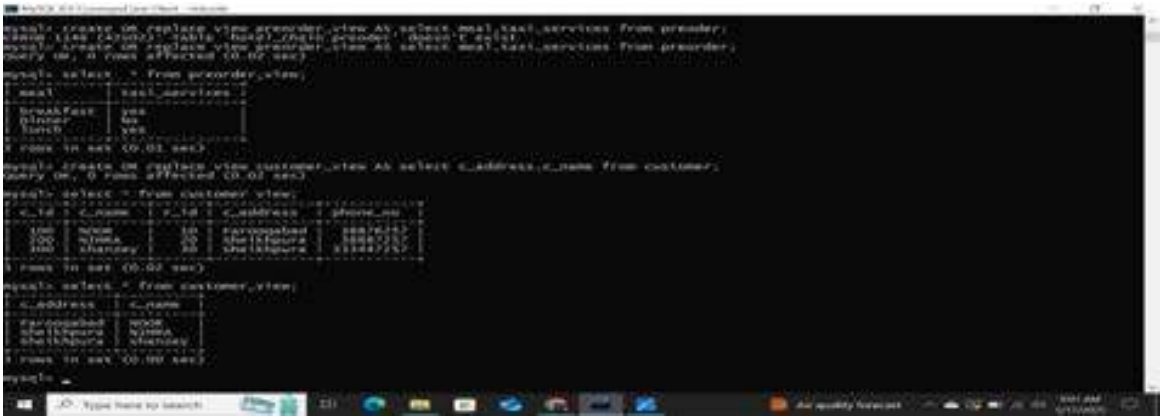


ARITHMETIC OPERATIONS ON STAFF



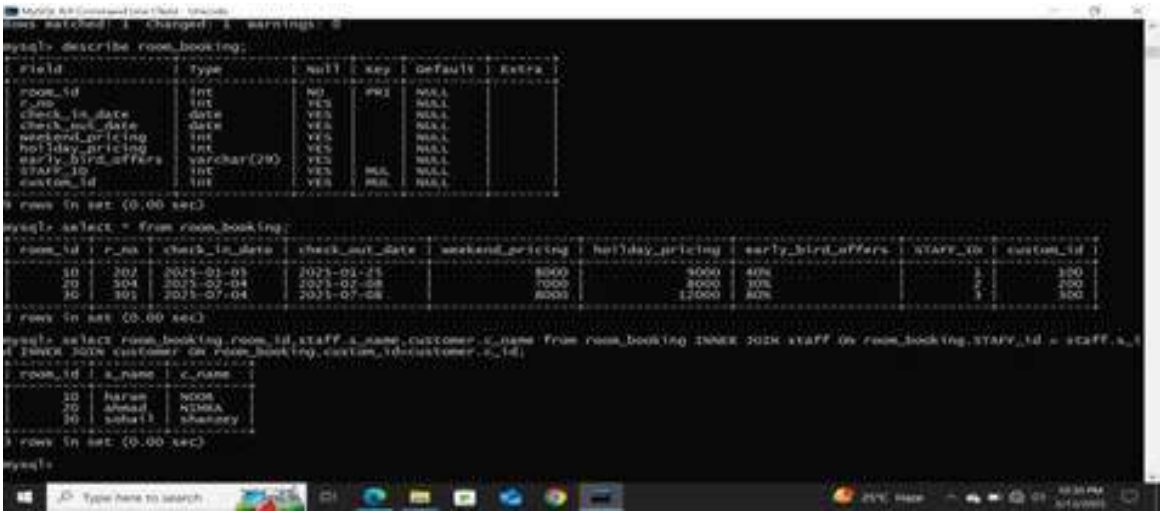
COUNT & COUNT (*)

VIEW

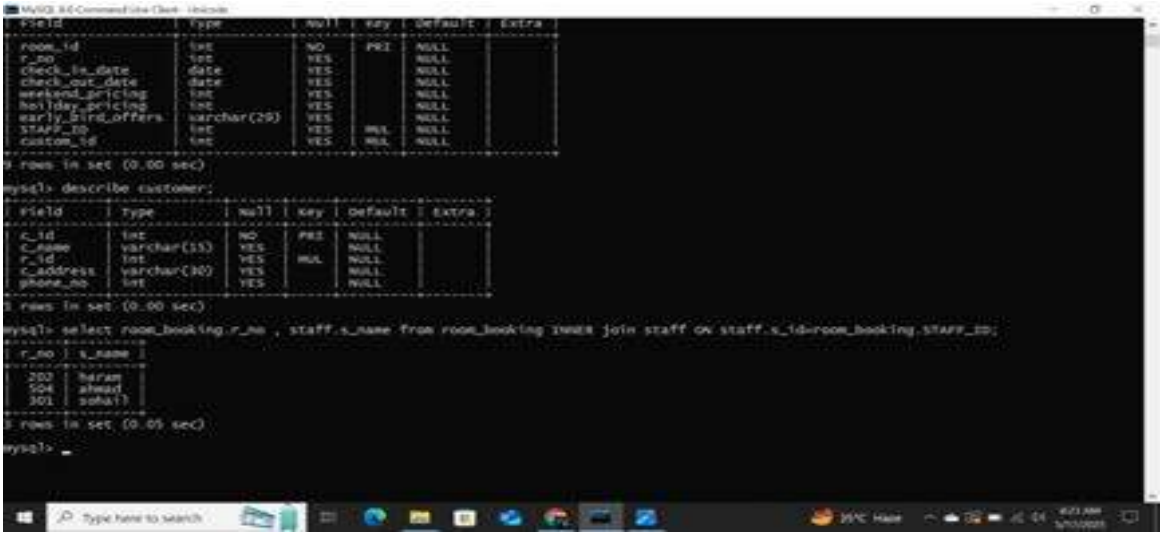


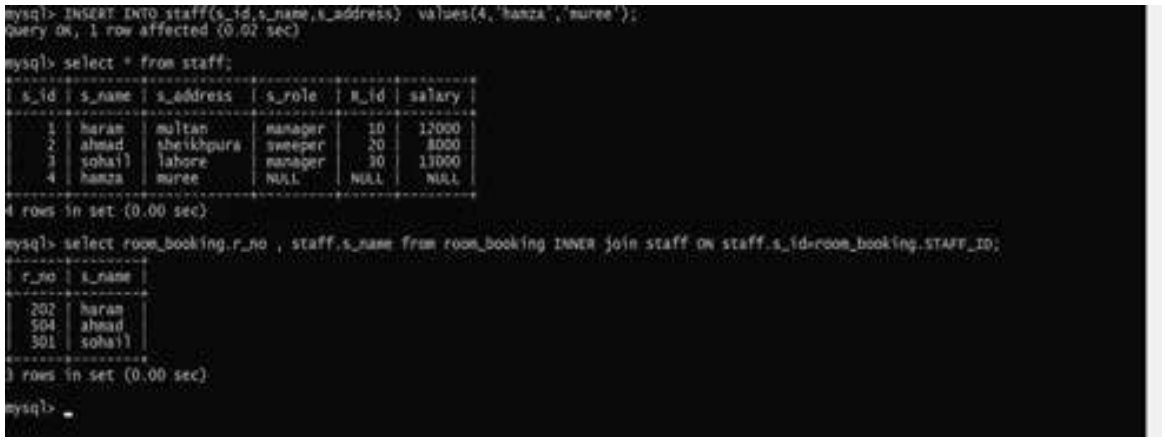
LIKE CLAUSE ON CUSTOMER

JOINS

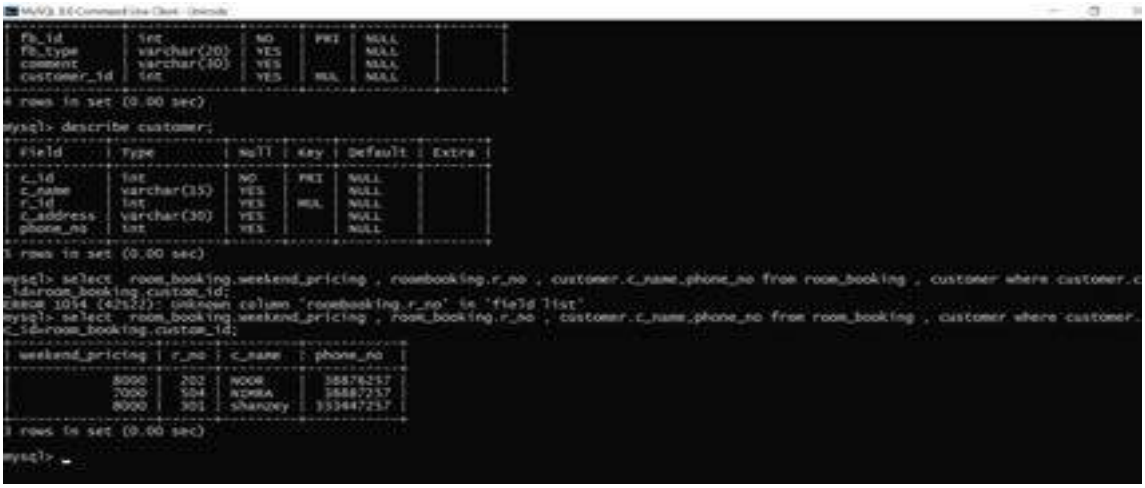


INNER JOIN



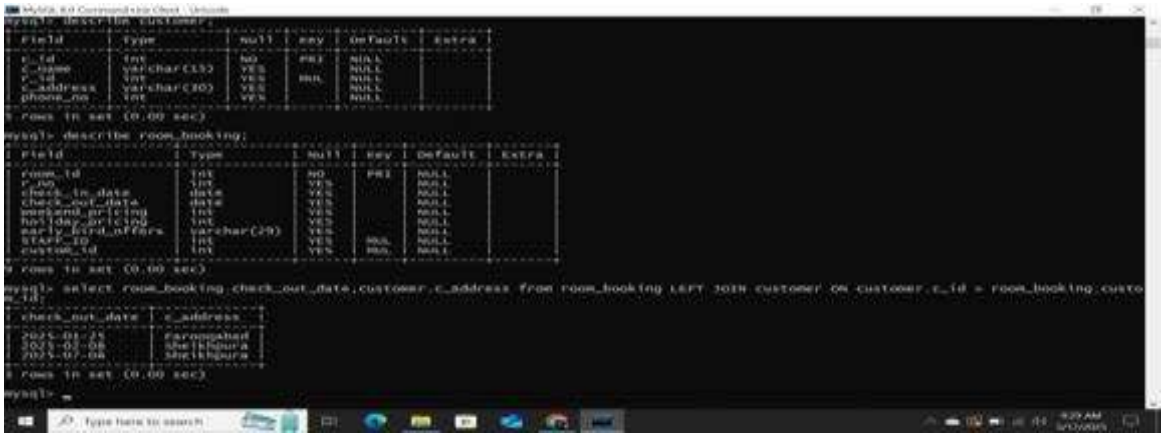


EQUI JOIN



```
| fb_id       | int         | NO   | PRI | NULL |       |
| fb_type     | varchar(20) | YES  |     | NULL |       |
| comment     | varchar(30) | YES  |     | NULL |       |
| customer_id | int         | YES  | MUL | NULL |       |
+-------------+-------------+------+-----+------+-------+
4 rows in set (0.00 sec)

mysql> describe customer;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| c_id      | int         | NO   | PRI | NULL    |       |
| c_name    | varchar(15) | YES  |     | NULL    |       |
| r_id      | int         | YES  | MUL | NULL    |       |
| c_address | varchar(30) | YES  |     | NULL    |       |
| phone_no  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> select  room_booking.weekend_pricing , roombooking.r_no , customer.c_name,phone_no from room_booking , customer where customer.c
_id=room_booking.custom_id;
ERROR 1054 (42S22): Unknown column 'roombooking.r_no' in 'field list'
mysql> select  room_booking.weekend_pricing , room_booking.r_no , customer.c_name,phone_no from room_booking , customer where customer.
c_id=room_booking.custom_id;
+-----------------+------+--------+----------+
| weekend_pricing | r_no | c_name | phone_no |
+-----------------+------+--------+----------+
|            8000 |  202 | NOOR   | 38876257 |
|            7000 |  504 | NIMRA  | 38887257 |
|            8000 |  301 | shanzey| 33344727 |
+-----------------+------+--------+----------+
3 rows in set (0.00 sec)

mysql>
```

LEFT JOIN

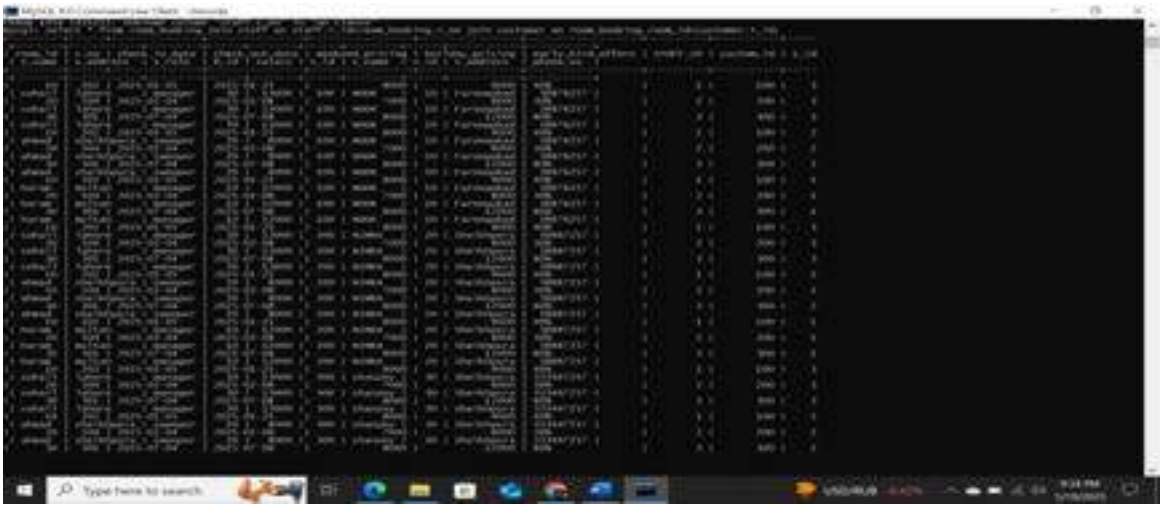


```
mysql> describe customer;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| c_id      | int         | NO   | PRI | NULL    |       |
| c_name    | varchar(15) | YES  |     | NULL    |       |
| r_id      | int         | YES  | MUL | NULL    |       |
| c_address | varchar(30) | YES  |     | NULL    |       |
| phone_no  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> describe room_booking;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| room_id         | int         | NO   | PRI | NULL    |       |
| r_no            | int         | YES  |     | NULL    |       |
| check_in_date   | date        | YES  |     | NULL    |       |
| check_out_date  | date        | YES  |     | NULL    |       |
| weekend_pricing | int         | YES  |     | NULL    |       |
| holiday_pricing | int         | YES  |     | NULL    |       |
| early_bird_offers | varchar(20) | YES  |     | NULL    |       |
| staff_id        | int         | YES  | MUL | NULL    |       |
| custom_id       | int         | YES  | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> select room_booking.check_out_date,customer.c_address from room_booking LEFT JOIN customer ON customer.c_id = room_booking.custo
m_id;
+----------------+-----------+
| check_out_date | c_address |
+----------------+-----------+
| 2025-01-25     | Farooqabad |
| 2025-02-08     | Sheikhpura |
| 2025-07-08     | Sheikhpura |
+----------------+-----------+
3 rows in set (0.00 sec)

mysql>
```

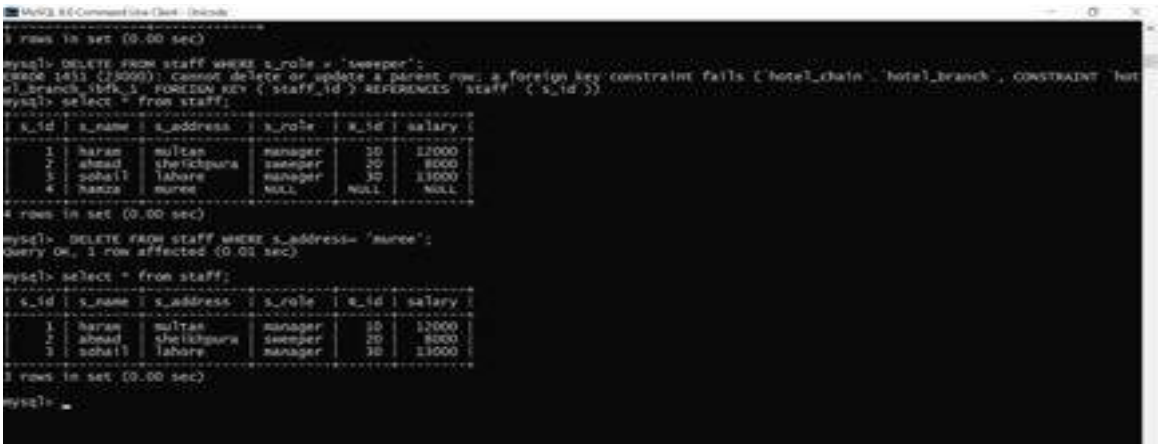
RIGHT JOIN

```
| shanzey |   30 | Sheikhpura | 33344727 |
+---------+------+------------+----------+
3 rows in set (0.00 sec)

mysql> select room_booking.check_out_date,customer.c_address from room_booking RIGHT JOIN customer ON customer.c_id = room_booking.cust
om_id;
+----------------+-----------+
| check_out_date | c_address |
+----------------+-----------+
| 2025-01-25     | Farooqabad |
| 2025-02-08     | Sheikhpura |
| 2025-07-08     | Sheikhpura |
+----------------+-----------+
3 rows in set (0.00 sec)
```

CROSS JOIN

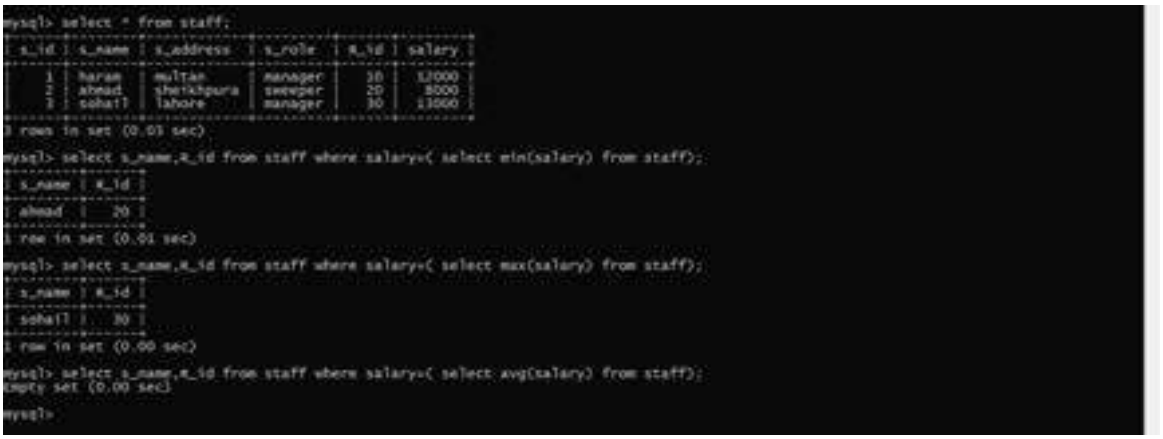


DELETE ROW4 FROM STAFF



DROP

SUBQUERY



```
mysql> select * from staff;
+------+--------+-----------+---------+------+--------+
| s_id | s_name | s_address | s_role  | R_id | salary |
+------+--------+-----------+---------+------+--------+
|    1 | haram  | multan    | manager |   10 |  12000 |
|    2 | ahmad  | sherkhpura| sweeper |   20 |   8000 |
|    3 | sohail | lahore    | manager |   30 |  13000 |
+------+--------+-----------+---------+------+--------+
3 rows in set (0.01 sec)

mysql> select s_name,R_id from staff where salary=( select min(salary) from staff);
+--------+------+
| s_name | R_id |
+--------+------+
| ahmad  |   20 |
+--------+------+
1 row in set (0.01 sec)

mysql> select s_name,R_id from staff where salary=( select max(salary) from staff);
+--------+------+
| s_name | R_id |
+--------+------+
| sohail |   30 |
+--------+------+
1 row in set (0.00 sec)

mysql> select s_name,R_id from staff where salary=( select avg(salary) from staff);
Empty set (0.00 sec)

mysql>
```

TRUNCATE

```
mysql> truncate table course;
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+-------------------+
| Tables_in_college |
+-------------------+
| course            |
+-------------------+
1 row in set (0.00 sec)

mysql> select * from course;
Empty set (0.00 sec)
```

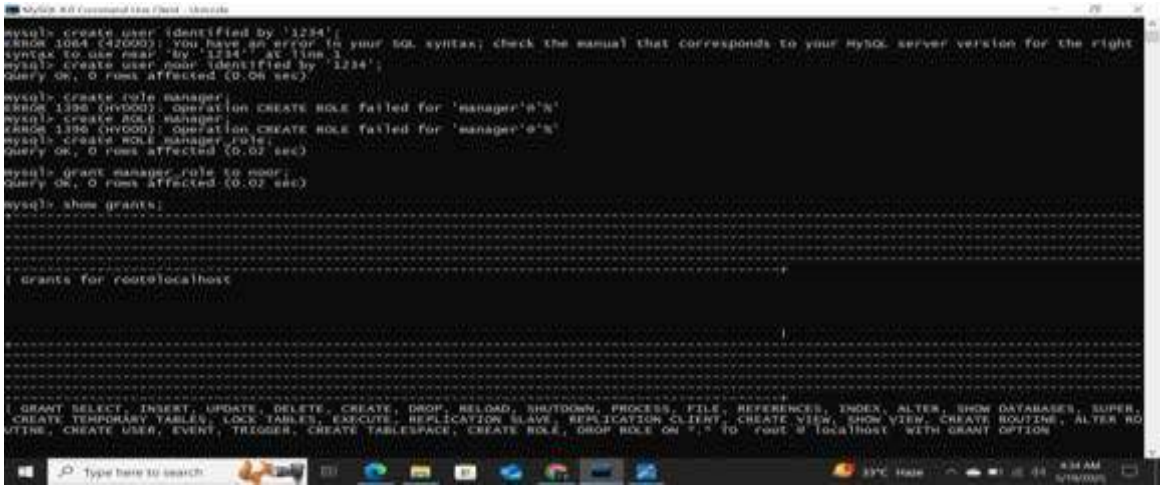
LOWEST SALARY PRINTED

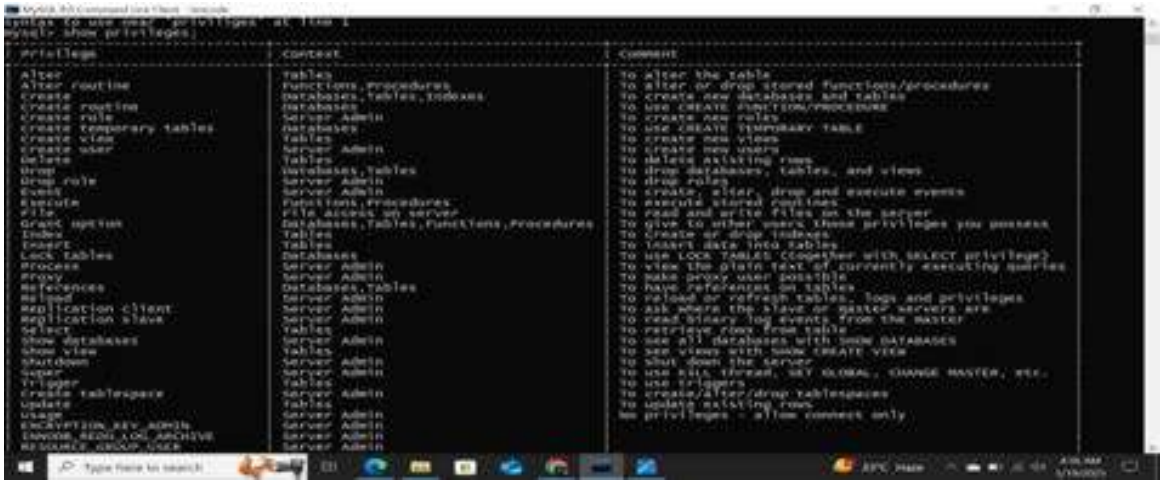
```
MySQL 8.0 Command Line Client - Unicode
| haram  |   10 |
| sohail |   30 |
+--------+------+
2 rows in set (0.00 sec)

mysql>  select s_name,R_id , salary from staff where salary in( select min(salary) from staff group by s_role);
+--------+------+--------+
| s_name | R_id | salary |
+--------+------+--------+
| haram  |   10 |  12000 |
| ahmad  |   20 |   8000 |
+--------+------+--------+
2 rows in set (0.00 sec)

mysql>
```

MIN SALARY USING GROUP BY FROM S-ID PRINTED

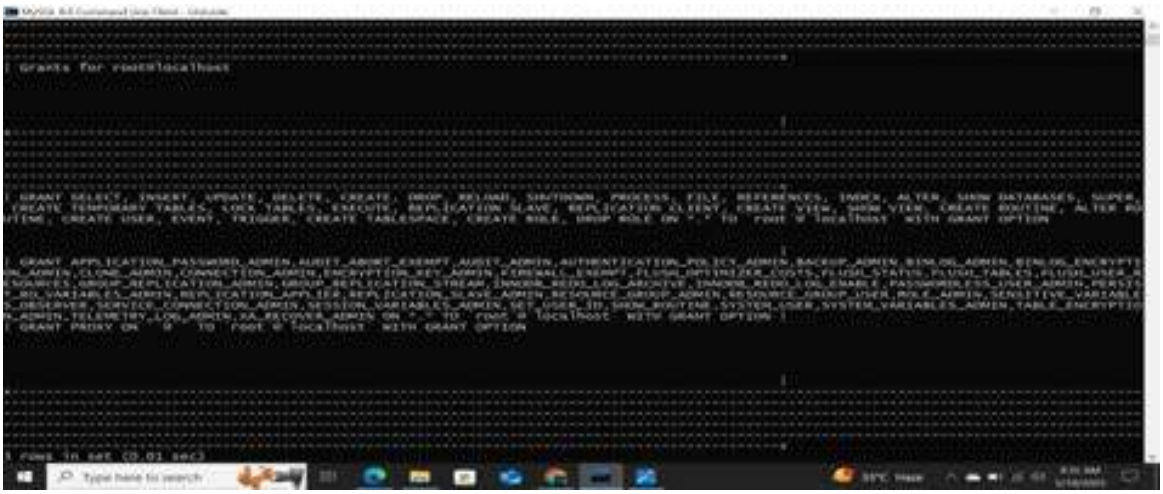
```
mysql> select s_name,R_id,salary from staff where salary>any( select min(salary) from staff group by s_role);
+--------+------+--------+
| s_name | R_id | salary |
+--------+------+--------+
| haram  |   10 |  12000 |
| sohail |   30 |  13000 |
+--------+------+--------+
2 rows in set (0.00 sec)

mysql> select s_name,R_id,salary from staff where salary>ALL( select min(salary) from staff group by s_role);
+--------+------+--------+
| s_name | R_id | salary |
+--------+------+--------+
| sohail |   30 |  13000 |
+--------+------+--------+
1 row in set (0.00 sec)

mysql>
```

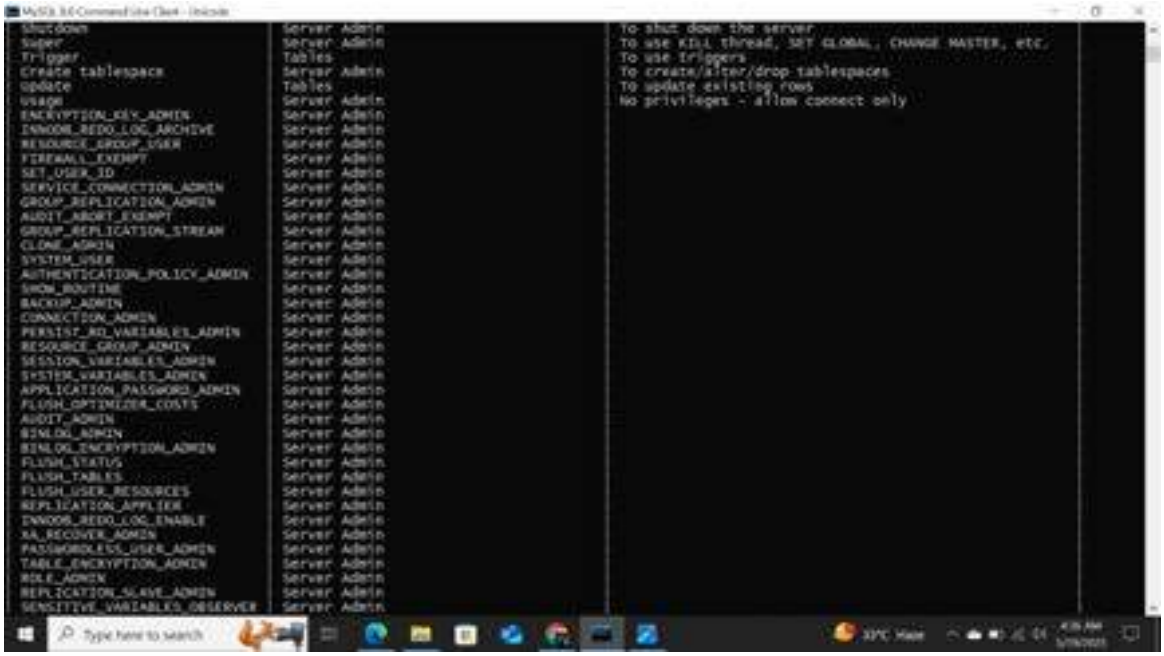
GRANT

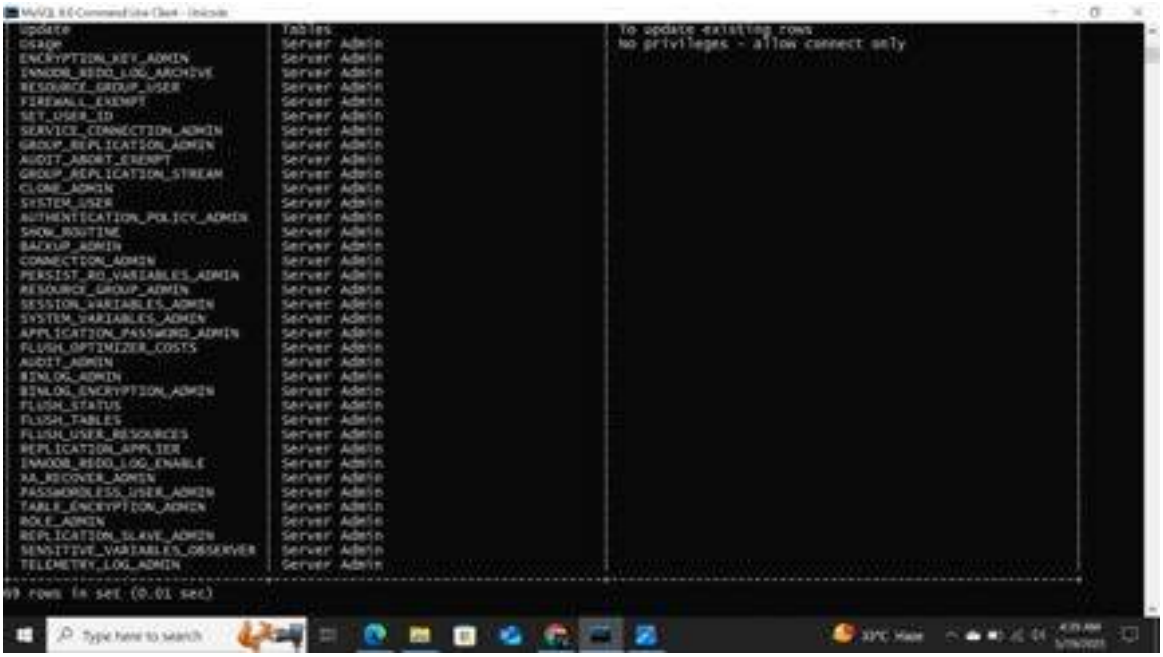




PRIVILEGES




MySQL 8.0 Command Line Client - Unicode
SHUTDOWN Server Admin To shut down the server
SUPER Server Admin To use KILL thread, SET GLOBAL, CHANGE MASTER, etc.
TRIGGER Tables To use triggers
CREATE TABLESPACE Server Admin To create/alter/drop tablespaces
UPDATE Tables To update existing rows
USAGE Server Admin No privileges - allow connect only
ENCRYPTION_KEY_ADMIN Server Admin
INNODB_REDO_LOG_ARCHIVE Server Admin
RESOURCE_GROUP_USER Server Admin
FIREWALL_EXEMPT Server Admin
SET_USER_ID Server Admin
SERVICE_CONNECTION_ADMIN Server Admin
GROUP_REPLICATION_ADMIN Server Admin
AUDIT_ABORT_EXEMPT Server Admin
GROUP_REPLICATION_STREAM Server Admin
CLONE_ADMIN Server Admin
SYSTEM_USER Server Admin
AUTHENTICATION_POLICY_ADMIN Server Admin
SHOW_ROUTINE Server Admin
BACKUP_ADMIN Server Admin
CONNECTION_ADMIN Server Admin
PERSIST_RO_VARIABLES_ADMIN Server Admin
RESOURCE_GROUP_ADMIN Server Admin
SESSION_VARIABLES_ADMIN Server Admin
SYSTEM_VARIABLES_ADMIN Server Admin
APPLICATION_PASSWORD_ADMIN Server Admin
FLUSH_OPTIMIZER_COSTS Server Admin
AUDIT_ADMIN Server Admin
BINLOG_ADMIN Server Admin
BINLOG_ENCRYPTION_ADMIN Server Admin
FLUSH_STATUS Server Admin
FLUSH_TABLES Server Admin
FLUSH_USER_RESOURCES Server Admin
REPLICATION_APPLIER Server Admin
INNODB_REDO_LOG_ENABLE Server Admin
XA_RECOVER_ADMIN Server Admin
PASSWORDLESS_USER_ADMIN Server Admin
TABLE_ENCRYPTION_ADMIN Server Admin
ROLE_ADMIN Server Admin
REPLICATION_SLAVE_ADMIN Server Admin
SENSITIVE_VARIABLES_OBSERVER Server Admin
Type here to search
33°C Haze


MySQL 8.0 Command Line Client - Unicode
update	Tables	To update existing rows
usage	Server Admin	No privileges - allow connect only
ENCRYPTION_KEY_ADMIN	Server Admin
INNODB_REDO_LOG_ARCHIVE	Server Admin
RESOURCE_GROUP_USER	Server Admin
FIREWALL_EXEMPT	Server Admin
SET_USER_ID	Server Admin
SERVICE_CONNECTION_ADMIN	Server Admin
GROUP_REPLICATION_ADMIN	Server Admin
AUDIT_ABORT_EXEMPT	Server Admin
GROUP_REPLICATION_STREAM	Server Admin
CLONE_ADMIN	Server Admin
SYSTEM_USER	Server Admin
AUTHENTICATION_POLICY_ADMIN	Server Admin
SHOW_ROUTINE	Server Admin
BACKUP_ADMIN	Server Admin
CONNECTION_ADMIN	Server Admin
PERSIST_RO_VARIABLES_ADMIN	Server Admin
RESOURCE_GROUP_ADMIN	Server Admin
SESSION_VARIABLES_ADMIN	Server Admin
SYSTEM_VARIABLES_ADMIN	Server Admin
APPLICATION_PASSWORD_ADMIN	Server Admin
FLUSH_OPTIMIZER_COSTS	Server Admin
AUDIT_ADMIN	Server Admin
BINLOG_ADMIN	Server Admin
BINLOG_ENCRYPTION_ADMIN	Server Admin
FLUSH_STATUS	Server Admin
FLUSH_TABLES	Server Admin
FLUSH_USER_RESOURCES	Server Admin
REPLICATION_APPLIER	Server Admin
INNODB_REDO_LOG_ENABLE	Server Admin
XA_RECOVER_ADMIN	Server Admin
PASSWORDLESS_USER_ADMIN	Server Admin
TABLE_ENCRYPTION_ADMIN	Server Admin
ROLE_ADMIN	Server Admin
REPLICATION_SLAVE_ADMIN	Server Admin
SENSITIVE_VARIABLES_OBSERVER	Server Admin
TELEMETRY_LOG_ADMIN	Server Admin
89 rows in set (0.01 sec)
Type here to search
33°C Haze