

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

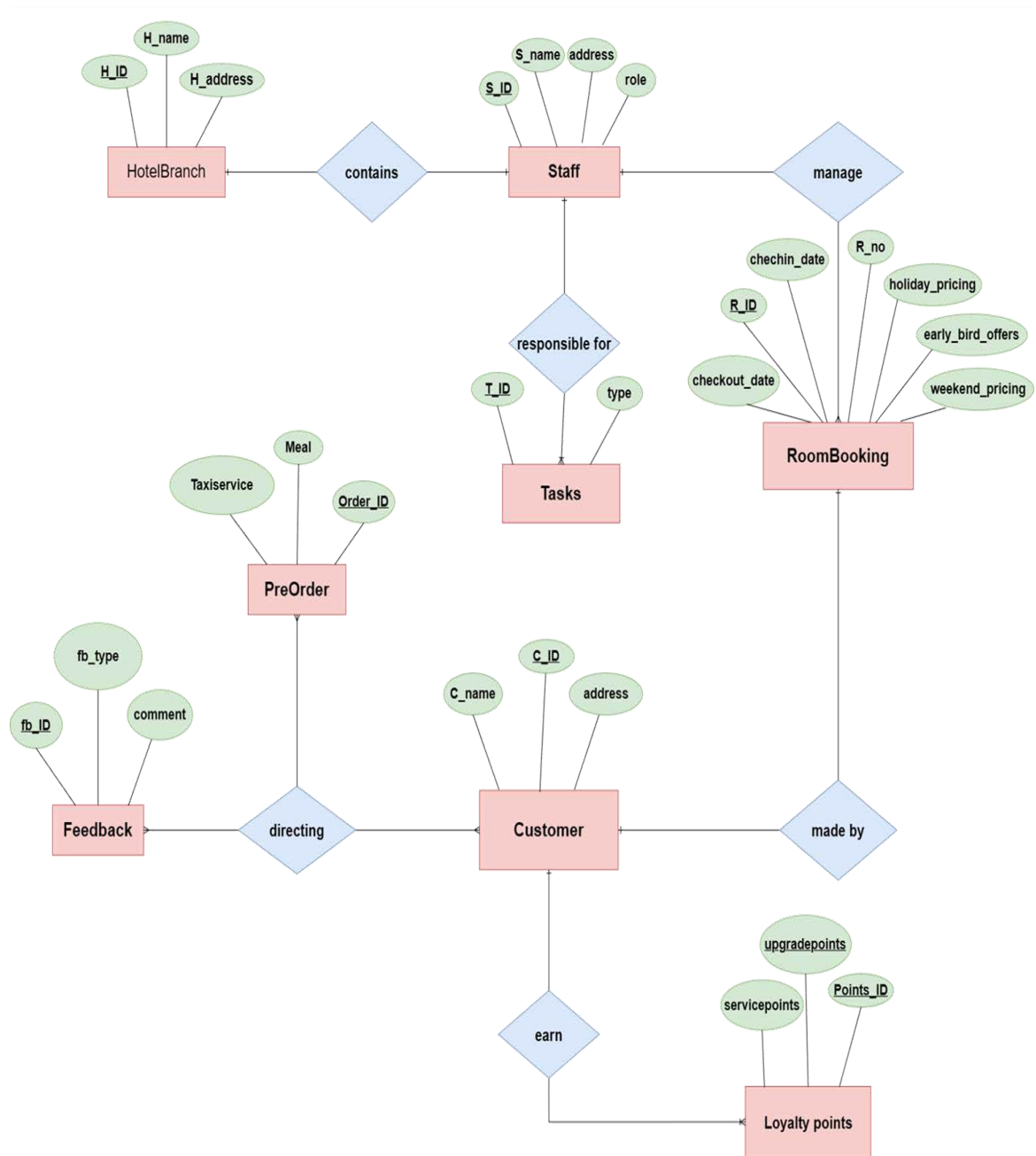
SQL stands for structure query language. It is a widely used, open source relational database management system.

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 CONSTRAINTS

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 aggregate_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 aggregate_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



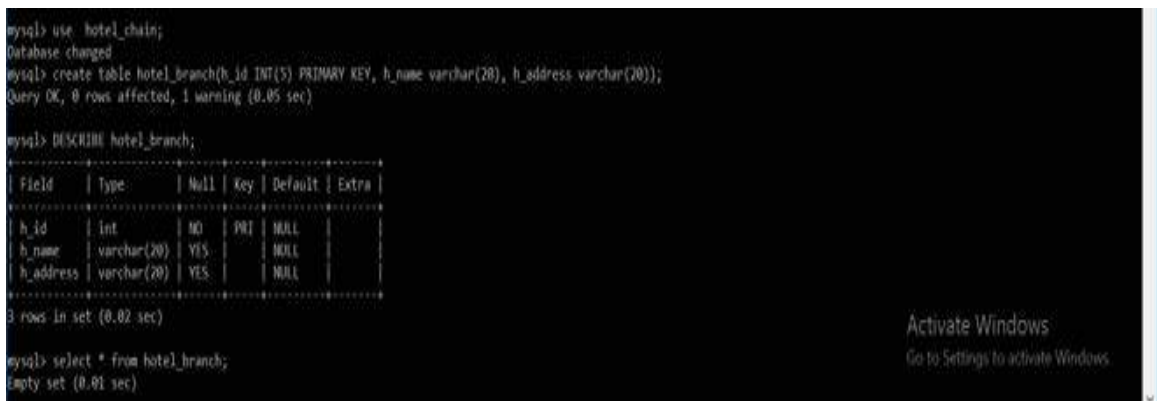
```
MySQL 8.0 Command Line Client
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database hotel_chain;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| college |
| hotel_chain |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
7 rows in set (0.04 sec)
```

CREATE Hotel_Branch Table



```
mysql> use hotel_chain;
Database changed
mysql> create table hotel_branch(h_id INT(5) PRIMARY KEY, h_name varchar(20), h_address varchar(20));
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> DESCRIBE hotel_branch;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| h_id | int | NO | PRI | NULL | |
| h_name | varchar(20) | YES | | NULL | |
| h_address | varchar(20) | YES | | NULL | |
+-----+
3 rows in set (0.02 sec)

mysql> select * from hotel_branch;
Empty set (0.01 sec)
```

INSERTING VALUES IN HOTEL_BRANCH



```
MySQL 8.0 Command Line Client - Unicode
+-----+
| h_address | varchar(30) | YES | | NULL | |
+-----+
3 rows in set (0.00 sec)

mysql> select * from hotel_branch;
+-----+
| h_id | h_name | h_address |
+-----+
| 1001 | unique | lahore |
| 1002 | five_star | multan |
| 1003 | Super | sheikhpura |
+-----+
3 rows in set (0.00 sec)
```

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    | ahead  | 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
-> -AC
mysql> select s_name,s_address,s_role from staff;
+-----+-----+-----+
| s_name | s_address | s_role |
+-----+-----+-----+
| haram  | multan    | manager |
| ahead  | sheikhpura | sweeper |
| sohail | lahore     | receptionist |
+-----+-----+-----+

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

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    |       |
| 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_pricing 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(29)	YES		NULL	
STAFF_ID	int	YES	MUL	NULL	

```
3 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,hoilday_pricing,early_bird_offers from Room_booking;
ERROR 1054 (42S22): unknown column 'hoilday_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, 1 rows affected (0.14 sec)
Records: 1 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','shanze', '10', '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	shanze	10	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	100

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 (cus_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 | |
| cus_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 | cus_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) UNSIGNED, type varchar(20), st_id INT(5));
Query OK, 0 rows affected (0.00 sec)

mysql> DESCRIBE task;
ERROR 1146 (42S02): Table 'hotel.task' 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 | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

ADD st_id FOREIGN KEY AND INSERTION

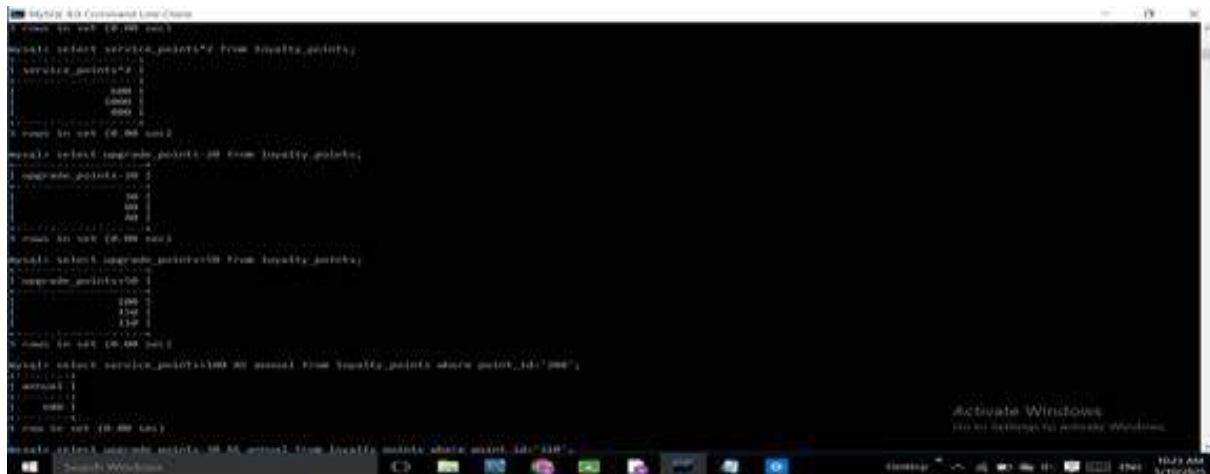
```
mysql> ALTER table task ADD FOREIGN KEY (st_id) REFERENCES staff(st_id);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

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 |
+-----+-----+-----+
| 100 | cleaning | 1 |
| 201 | maintenance | 2 |
| 400 | guard | 3 |
+-----+-----+-----+
```

PRACTICAL IMPLEMENTATION

ARITHMETIC & LOGICAL OPERATIONS ON LOYALTY_POINTS TABLE



```
mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select service_points*2 from loyalty_points;
+-----+
| service_points*2 |
+-----+
| 1000             |
| 1000             |
| 1000             |
+-----+

mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select upgrade_points*10 from loyalty_points;
+-----+
| upgrade_points*10 |
+-----+
| 100               |
| 100               |
| 100               |
+-----+

mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select service_points*100 as annual from loyalty_points where point_id=200;
+-----+
| annual |
+-----+
| 10000  |
+-----+

mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select upgrade_points*10 as annual from loyalty_points where point_id=210;
+-----+
| annual |
+-----+
| 100    |
+-----+
```

LOGICAL OPERATIONS ADDING ph_no COLUMN IN CUSTOMER



```
mysql> ALTER table Customer ADD VARCHAR(14) ph_no;
Query OK, 0 rows affected, 1 warning (0.00 sec)
Warnings: 0 Duplicates: 0 Warnings: 1

mysql> UPDATE Customer set ph_no='0223377781' where c_id='100';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE Customer set ph_no='0224078609' where c_id='200';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE Customer set ph_no='0119400061' where c_id='300';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Customer;
+----+-----+-----+-----+-----+
| c_id | c_name | c_id | c_address | ph_no |
+----+-----+-----+-----+-----+
| 100 | Shreyas | 10 | Jambhikar | 0223377781 |
| 200 | Nisha | 20 | Kulkarni | 0224078609 |
| 300 | Nisha | 30 | Shikhar | 0119400061 |
+----+-----+-----+-----+-----+
```



```
mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select service_points*100 as annual from loyalty_points where point_id=200;
+-----+
| annual |
+-----+
| 10000  |
+-----+

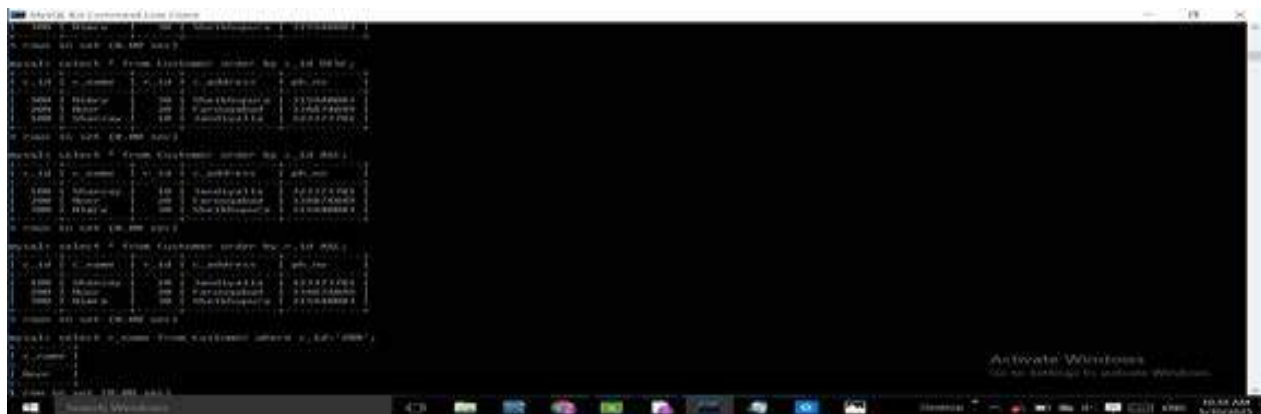
mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select upgrade_points*10 as annual from loyalty_points where point_id=210;
+-----+
| annual |
+-----+
| 100    |
+-----+

mysql> show tables;
+-----+
| tables_in_loyalty |
+-----+
| loyalty_points    |
+-----+

mysql> select upgrade_points*100 as annual from loyalty_points where point_id=220;
+-----+
| annual |
+-----+
| 10000  |
+-----+
```

ORDER BY CLAUSE ON CUSTOMER



GIVING TWO h_ids SAME NAME



DISTINCT CLAUSE AND ADDING SALARY COLUMN IN STAFF TABLE

```
mysql> select DISTINCT s_name from hotel_booking;
+-----+
| s_name |
+-----+
| Spring |
| Fall_Aut |
+-----+
2 rows in set (0.00 sec)

mysql> select * from staff;
+----+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role | s_sal |
+----+-----+-----+-----+-----+
| 1 | Haron | Sultan | manager | 10 |
| 2 | Premal | Shanthipore | manager | 20 |
| 3 | Subimal | Kothare | receptionist | 30 |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> ALTER TABLE staff ADD (new_salary DEC(10));
Empty set, 0 rows affected, 1 warning (0.00 sec)
Warning: 0. Duplicate: 0. Warnings: 1

mysql> DESCRIBE staff;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| s_id | int | YES | PRI | NULL | |
| s_name | varchar(20) | YES | | NULL | |
| s_address | varchar(20) | YES | | NULL | |
| s_role | varchar(20) | YES | | NULL | |
| s_sal | int | YES | | NULL | |
| salary | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE staff set salary='2000' where s_id='1';
Empty set, 1 row affected (0.00 sec)
```

BETWEEN, HAVING AND IN CLAUSES

```
mysql> select em_name from Employee where job='assistant' OR salary BETWEEN 15000 AND 30000;
+-----+
| em_name |
+-----+
| Ali |
| Ahmad |
| Arham |
+-----+
3 rows in set (0.00 sec)

mysql> select em_name from Employee where job='manager' OR salary BETWEEN 15000 AND 30000;
+-----+
| em_name |
+-----+
| Ali |
| Ahmad |
| Arham |
+-----+
3 rows in set (0.00 sec)

mysql> select em_name from Employee where salary IN (15000,30000);
+-----+
| em_name |
+-----+
| Ali |
| Ahmad |
| Arham |
+-----+
3 rows in set (0.00 sec)

mysql> select em_name from Employee where salary IN (15000,30000,20000);
+-----+
| em_name |
+-----+
| Ali |
| Ahmad |
| Arham |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select max(salary) from staff where s_role having count(s_id)>1;
+-----+
| max(salary) |
+-----+
| 13000 |
+-----+
1 row in set (0.00 sec)

mysql> select min(salary) from staff where s_role having count(s_id)>1;
+-----+
| min(salary) |
+-----+
| 12000 |
+-----+
1 row in set (0.00 sec)

mysql> select s_name from staff where salary<13000 between salary >8000;
Empty set (0.00 sec)
Warning: 0. Syntax error: 0. Warnings: 1
mysql> select s_name from staff where salary<13000 between salary<8000;
Empty set (0.00 sec)
Warning: 0. Syntax error: 0. Warnings: 1
mysql> select s_name from staff where salary<8000 as line 1;
Empty set (0.00 sec)
Warning: 0. Syntax error: 0. Warnings: 1
mysql> select s_name from staff where salary<8000 as salary<13000;
+-----+
| s_name |
+-----+
| Haron |
+-----+
1 row in set, 1 warning (0.00 sec)

mysql> select s_name from staff where s_role='sweeper' as salary<8000 as salary<13000;
Empty set, 2 warnings (0.00 sec)

mysql> select s_name from staff where s_role='sweeper' as salary<8000 as salary<13000;
+-----+
| s_name |
+-----+
| Ahmad |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select min(salary) from staff GROUP BY s_role where count(s_id)>1;
+-----+
| min(salary) |
+-----+
| 40000       |
+-----+
1 row in set (0.00 sec)

mysql> select s_name from staff where s_role='manager' and salary=40000;
+-----+
| s_name |
+-----+
| Heman |
+-----+
1 row in set (0.00 sec)

mysql> select s_name from staff where s_role='manager' and salary BETWEEN 30000 and 50000;
+-----+
| s_name |
+-----+
| Heman |
+-----+
1 row in set (0.00 sec)

mysql> select s_name from staff where s_address='Mumbai' and salary BETWEEN 30000 and 50000;
+-----+
| s_name |
+-----+
| Heman |
+-----+
1 row in set (0.00 sec)

mysql> select s_name from staff where s_address='Mumbai' and salary BETWEEN 30000 and 50000;
+-----+
| s_name |
+-----+
| Heman |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from room_booking;
+-----+-----+-----+-----+-----+-----+-----+-----+
| room_id | r_no | checkin_date | checkout_date | weekend_pricing | holiday_pricing | available_offers | sta_id |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10      | 200  | 2023-07-22   | 2023-07-26   | 4000           | 5000            | 5000            | 1      |
| 10      | 201  | 2023-07-22   | 2023-07-26   | 4000           | 5000            | 5000            | 2      |
| 10      | 202  | 2023-07-22   | 2023-07-26   | 4000           | 5000            | 5000            | 3      |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select room_id from room_booking where weekend_pricing IN(4000,5000);
+-----+
| room_id |
+-----+
| 10      |
| 10      |
| 10      |
+-----+
3 rows in set (0.00 sec)

mysql> select room_no from room_booking where holiday_pricing IN(52000,60000);
ERROR 1105 (HY000): unknown column 'room_no' in 'field list'
mysql> select r_no from room_booking where holiday_pricing IN(52000,60000);
+-----+
| r_no |
+-----+
| 200  |
| 201  |
| 202  |
+-----+
3 rows in set (0.00 sec)

mysql> select * from customer;
+-----+-----+-----+-----+-----+
| s_id | s_name | r_id | s_address | ph_no |
+-----+-----+-----+-----+-----+
| 200  | Heman | 10   | Mumbai   | 123456789 |
| 201  | Heman | 10   | Mumbai   | 123456789 |
| 202  | Heman | 10   | Mumbai   | 123456789 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

ARITHMETIC OPERATIONS ON STAFF

```
mysql> UPDATE staff SET salary=50000 where s_id=1;
Query OK, 1 row affected (0.01 sec)
Rows updated: 1 Changed: 1 Warnings: 0

mysql> UPDATE staff SET salary=50000 where s_id=2;
Query OK, 1 row affected (0.01 sec)
Rows updated: 1 Changed: 1 Warnings: 0

mysql> UPDATE staff SET salary=40000 where s_id=3;
Query OK, 1 row affected (0.01 sec)
Rows updated: 1 Changed: 1 Warnings: 0

mysql> select * from salary;
ERROR 1105 (HY000): table 'tbln1.tbln.salary' doesn't exist
mysql> select * from staff;
+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role |
+-----+-----+-----+-----+
| 1     | Heman | Mumbai   | manager |
| 2     | Heman | Mumbai   | manager |
| 3     | Heman | Mumbai   | manager |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select avg(salary) from staff;
+-----+
| avg(salary) |
+-----+
| 40000       |
+-----+
1 row in set (0.00 sec)

mysql> select max(salary) from staff;
+-----+
| max(salary) |
+-----+
| 50000       |
+-----+
1 row in set (0.00 sec)
```

COUNT & COUNT (*)

```
mysql> select * from feedback;
+----+-----+-----+-----+
| fb_id | fb_type | comment | customer_id |
+----+-----+-----+-----+
| 202 | complaint | late services | 200 |
| 201 | complaint | great food | 200 |
| 100 | complaint | good services | 100 |
+----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> select count(*) from feedback;
+-----+
| count(*) |
+-----+
| 3 |
+-----+
3 rows in set (0.00 sec)

mysql> select count(comment) from feedback;
ERROR 1146 (42002): Table 'local_chain.feedback' doesn't exist
mysql> select count(comment) from feedback;
+-----+
| count(comment) |
+-----+
| 3 |
+-----+
3 rows in set (0.00 sec)

mysql>
```

VIEW

```
mysql> create or replace view provider_view as select meal,meal_services from provider;
Query OK, 0 rows affected (0.02 sec)

mysql> create or replace view provider_view as select meal,meal_services from provider;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from provider_view;
+-----+-----+
| meal | meal_services |
+-----+-----+
| breakfast | yes |
| lunch | yes |
| dinner | yes |
+-----+-----+
3 rows in set (0.01 sec)

mysql> create or replace view customer_view as select c_address,c_name from customer;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from customer_view;
+-----+-----+
| c_address | c_name |
+-----+-----+
| Noida | Noida |
| Noida | Noida |
| Noida | Noida |
+-----+-----+
3 rows in set (0.02 sec)

mysql> select * from customer_view;
+-----+-----+
| c_address | c_name |
+-----+-----+
| Noida | Noida |
| Noida | Noida |
| Noida | Noida |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

LIKE CLAUSE ON CUSTOMER

```
mysql> select c_name from customer where c_name like 'N%';
+-----+
| c_name |
+-----+
| Noida |
| Noida |
+-----+
2 rows in set (0.00 sec)

mysql> select c_name from customer where c_name like 'N%';
+-----+
| c_name |
+-----+
| Noida |
| Noida |
+-----+
2 rows in set (0.00 sec)

mysql> select c_name from customer where c_name like 'N%';
+-----+
| c_name |
+-----+
| Noida |
| Noida |
+-----+
2 rows in set (0.00 sec)

mysql>
```

JOINS

```
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(255) | YES | | NULL | |
| staff_id | int | YES | MUL | NULL | |
| custom_id | int | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from room_booking;
+-----+-----+-----+-----+-----+-----+-----+-----+
| room_id | r_no | check_in_date | check_out_date | weekend_pricing | holiday_pricing | early_bird_offers | staff_id | custom_id |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10 | 202 | 2023-01-03 | 2023-01-23 | 8000 | 9000 | 40% | 1 | 100 |
| 20 | 504 | 2023-02-04 | 2023-02-28 | 7000 | 8000 | 30% | 2 | 200 |
| 30 | 301 | 2023-07-04 | 2023-07-28 | 8000 | 12000 | 80% | 3 | 300 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select room_booking.room_id,staff.s_name,customer.c_name from room_booking INNER JOIN staff ON room_booking.STAFF_ID = staff.s_id
INNER JOIN customer ON room_booking.custom_id=customer.c_id;
+-----+-----+-----+
| room_id | s_name | c_name |
+-----+-----+-----+
| 20 | haran | Nisha |
| 20 | ahead | shahid |
| 30 | sohai | shahary |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

INNER JOIN

```
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(255) | YES | | NULL | |
| staff_id | int | YES | MUL | NULL | |
| custom_id | int | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> describe customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| c_id | int | NO | PRI | NULL | |
| c_name | varchar(33) | YES | | NULL | |
| r_id | int | YES | MUL | NULL | |
| c_address | varchar(200) | YES | | NULL | |
| phone_no | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select room_booking.r_no , staff.s_name from room_booking inner join staff ON staff.s_id=room_booking.STAFF_ID;
+-----+-----+
| r_no | s_name |
+-----+-----+
| 202 | haran |
| 504 | ahead |
| 301 | sohai |
+-----+-----+
3 rows in set (0.05 sec)

mysql>
```

```
mysql> INSERT INTO staff(s_id,s_name,s_address) values(4,'hamza','muree');
Query OK, 1 row affected (0.02 sec)

mysql> select * from staff;
+-----+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role | R_id | salary |
+-----+-----+-----+-----+-----+
| 1 | haran | multan | manager | 10 | 12000 |
| 2 | ahead | sherkhpura | sweeper | 20 | 8000 |
| 3 | sohai | lahore | manager | 30 | 11000 |
| 4 | hamza | muree | NULL | NULL | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select room_booking.r_no , staff.s_name from room_booking INNER join staff ON staff.s_id=room_booking.STAFF_ID;
+-----+-----+
| r_no | s_name |
+-----+-----+
| 202 | haran |
| 504 | ahead |
| 301 | sohai |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

EQUI JOIN

```
mysql> describe customer;
+-----+-----+-----+-----+-----+
| Pb_id | Set  | NO | PRI | NULL |
| Pb_type | varchar(20) | YES | NULL |
| comment | varchar(10) | YES | NULL |
| customer_id | int | YES | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> describe room_booking;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| r_no | int | NO | PRI | NULL |
| c_name | varchar(35) | YES | NULL |
| r_id | int | YES | MUL | NULL |
| c_address | varchar(30) | YES | NULL |
| phone_no | int | YES | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

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.customer_id;
ERROR 1054 (42S22): Unknown column 'room_booking.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.customer_id;
+-----+-----+-----+-----+
| weekend_pricing | r_no | c_name | phone_no |
+-----+-----+-----+-----+
| 8000 | 202 | MOON | 3887237 |
| 7000 | 304 | ADWA | 3887237 |
| 8000 | 301 | shanay | 333447237 |
+-----+-----+-----+-----+
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(35) | YES | NULL |
| r_id | int | YES | MUL | NULL |
| c_address | varchar(30) | YES | NULL |
| phone_no | int | YES | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> describe room_booking;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| r_no | int | NO | PRI | NULL |
| c_name | varchar(35) | YES | NULL |
| r_id | int | YES | MUL | NULL |
| c_address | varchar(30) | YES | NULL |
| phone_no | int | YES | NULL |
+-----+-----+-----+-----+
3 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.customer_id;
+-----+-----+
| check_out_date | c_address |
+-----+-----+
| 2021-01-25 | farooqabad |
| 2021-02-04 | shanaypura |
| 2021-07-04 | shanaypura |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

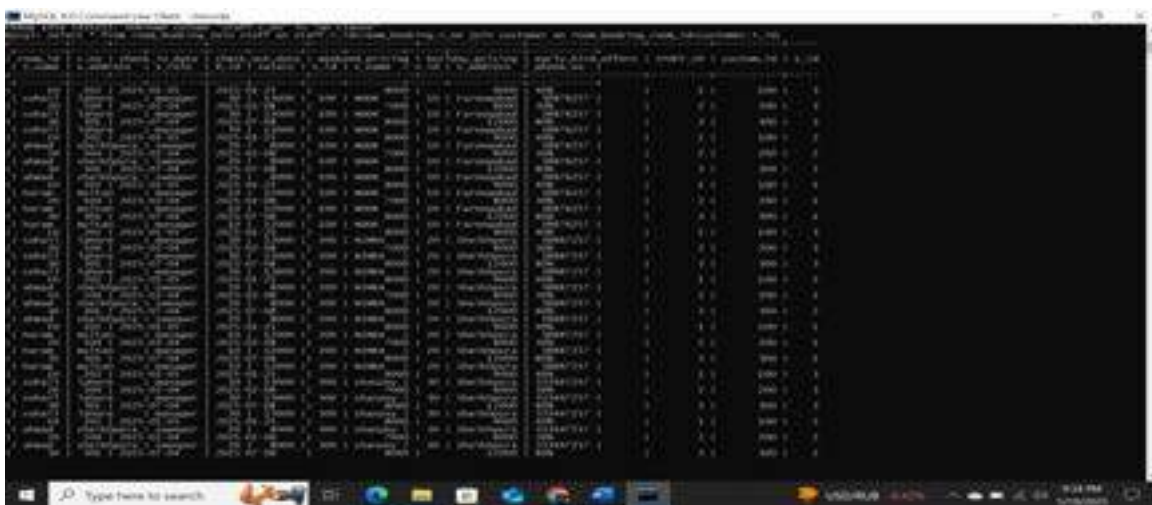
RIGHT JOIN

```
mysql> select room_booking.check_out_date, customer.c_address from room_booking RIGHT JOIN customer ON customer.c_id = room_booking.customer_id;
+-----+-----+
| check_out_date | c_address |
+-----+-----+
| 2021-01-25 | farooqabad |
| 2021-02-04 | shanaypura |
| 2021-07-04 | shanaypura |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```


CROSS JOIN

```
mysql> select distinct staff.s_id,room_booking.r_no from staff cross join room_booking on staff.s_id=room_booking.room_id;
+----+-----+
| s_id | r_no |
+----+-----+
| 10   | 202  |
| 20   | 204  |
| 30   | 301  |
+----+-----+
3 rows in set (0.01 sec)
```



The screenshot shows a MySQL command window with a large result set. The query is: `select distinct staff.s_id,room_booking.r_no from staff cross join room_booking on staff.s_id=room_booking.room_id;` The result set contains 3 rows, each with two columns: `s_id` and `r_no`. The rows are: (10, 202), (20, 204), and (30, 301). The status bar at the bottom indicates "3 rows in set (0.01 sec)".

DELETE ROW4 FROM STAFF

```
mysql> delete from staff where s_role = 'sweeper';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('hotel_chain', 'hotel_branch', CONSTRAINT 'hotel_branch_ibfk_1' FOREIGN KEY ('staff_id') REFERENCES 'staff' ('s_id'))
mysql> select * from staff;
+----+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role | s_id | salary |
+----+-----+-----+-----+-----+
| 1    | haran  | multan   | manager | 20   | 12000  |
| 2    | ahead  | sheikhpura | sweeper | 30   | 8000   |
| 3    | sohail | lahore    | manager | 30   | 13000  |
| 4    | hameed | muree    | NULL    | NULL  | NULL   |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM STAFF WHERE s_address= 'muree';
Query OK, 1 row affected (0.01 sec)

mysql> select * from staff;
+----+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role | s_id | salary |
+----+-----+-----+-----+-----+
| 1    | haran  | multan   | manager | 20   | 12000  |
| 3    | sohail | lahore    | manager | 30   | 13000  |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

DROP

```
mysql> drop task;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'task' at line 1
mysql> drop table task;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from task;
ERROR 1146 (42000): Table 'extra.task' doesn't exist
mysql> show the task;
ERROR 1146 (42000): Table 'extra.task' doesn't exist
mysql>
```

SUBQUERY

```
mysql> select * from staff;
+----+-----+-----+-----+-----+
| s_id | s_name | s_address | s_role | s_id | salary |
+----+-----+-----+-----+-----+
| 10 | haran | multan | manager | 10 | 12000 |
| 20 | ahmad | sheikhpura | sweeper | 20 | 8000 |
| 30 | sohail | lahore | manager | 30 | 13000 |
+----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

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

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

mysql> select s_name,s_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 5.6 Command-line Client - college

+-----+-----+
| s_name | s_id |
+-----+-----+
| haran | 10 |
| sohail | 30 |
+-----+-----+
2 rows in set (0.00 sec)

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

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

mysql>
```

```
mysql> create user 'identified by '1234';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' by '1234' at line 1
mysql> create user root identified by '1234';
Query OK, 0 rows affected (0.00 sec)

mysql> create role manager;
ERROR 1190 (HY000): Operation CREATE ROLE failed for 'manager'@'%'
mysql> create role manager;
ERROR 1190 (HY000): Operation CREATE ROLE failed for 'manager'@'%'
mysql> create role manager role;
Query OK, 0 rows affected (0.02 sec)

mysql> grant manager role to root;
Query OK, 0 rows affected (0.02 sec)

mysql> show grants;
+-----+
| grants for root@localhost |
+-----+
|
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER,
| TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER RO
| UTINE, CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO root@localhost WITH GRANT OPTION
+-----+
```

PRIVILEGES

[illegible]

MySQL 8.0 Command Line (Dev) - Oracle		
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
UNLOCK	Server Admin	No privileges - allow connect only
ENCRYPTION_KEY_ADMIN	Server Admin	
INNO_DB_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	
SWRM_ROUTINE	Server Admin	
BACKUP_ADMIN	Server Admin	
CONNECTION_ADMIN	Server Admin	
PERSISTENT_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	
INNO_DB_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	

