

HOTEL CHAIN MANAGEMENT SYSTEM



MYSQL QUERIES ON
NORMALIZATION
PROJECT

GROUP 16

Nimra Ashraf & Noor Fatima

110829 & 110830

MORNING

Submitted to:

Mam Sehrish Khan

11/05/2025

TABLE OF CONTENTS

1. DESCRIPTION

2. ENTITY RELATIONSHIP DIAGRAM

3. RELATIONAL SCHEMA

4. SQL QUERIES

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

5. NORMALIZATION TABLES

6. PRACTICAL IMPLEMENTATION OF QUERIES

DESCRIPTION

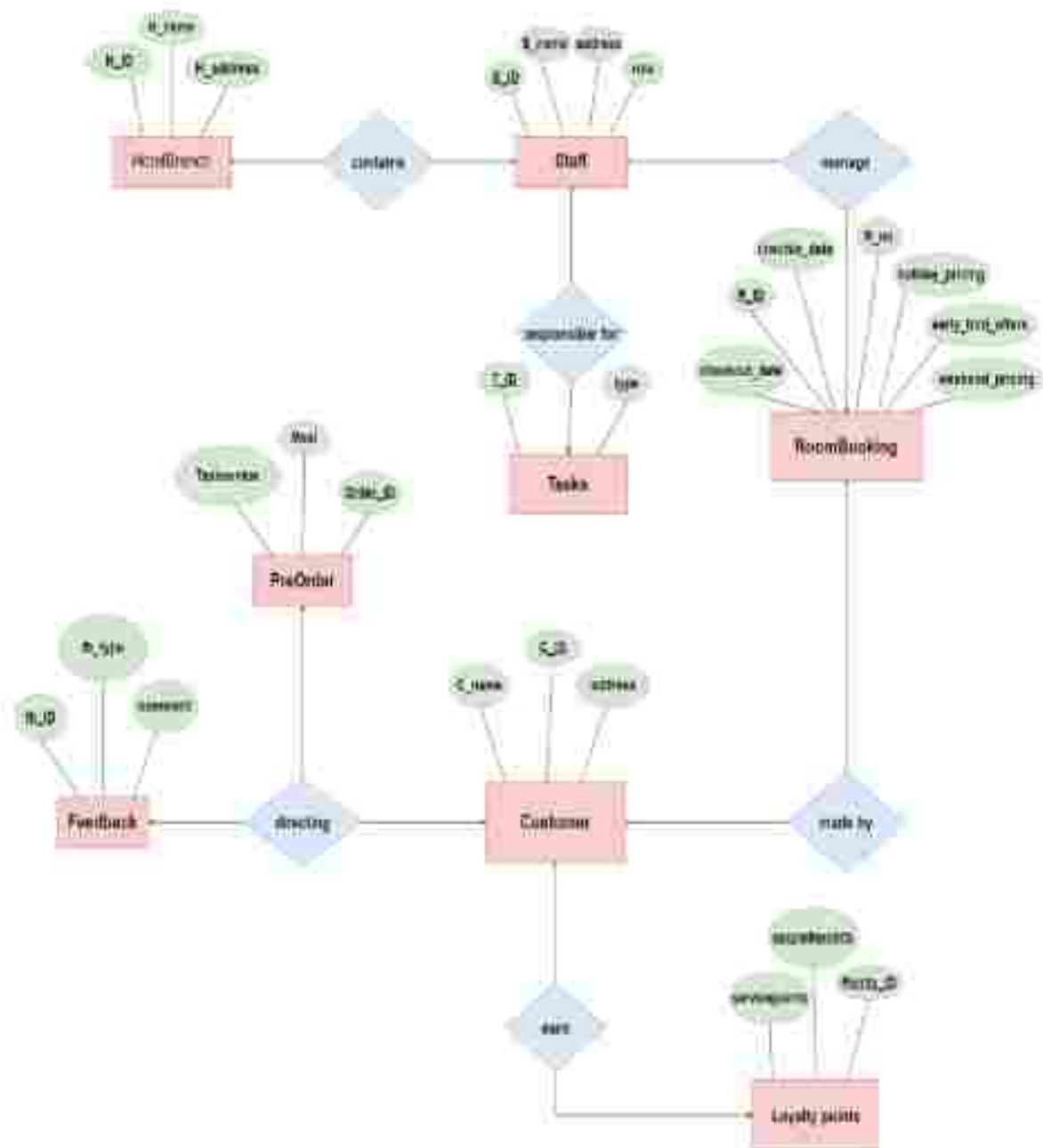
SQL stands for structure query language. It is a widely used, open source relational database management system.

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. to perform addition, subtraction, multiplication and division of two attributes.

⇒ **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 set. Otherwise for sorting the records in descending order DESC order is used. For ascending ASC and descending DESC is used.

Syntaxes of above two orders are:

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

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

GROUP BY

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

Syntax:

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

AS

The AS keyword in MySQL is used to assign an alias to a table or column, making it easier to reference or improving readability. 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. It ensures that the query returns only unique values in the specified columns.

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(*)

Used to count the number of rows in a table. This function counts all rows 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, allowing you to filter on the results of

aggregate functions like COUNT, SUM, AVG, and others.

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 OR operator displays a record if any of the conditions separated by OR 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 are 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. There are multiple types of joins such as inner, right, left etc.

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 where  
    condition;
```

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. It is embedded inside another query and acts as input or output for that query.

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

```
mysql> insert into hotel_branch values(1,101,'unique',1001,null);
1 row in set (0.00 sec)

mysql> insert into hotel_branch values(2,102,'free_star',1002,null);
1 row in set (0.00 sec)

mysql> insert into hotel_branch values(3,103,'super',1003,null);
1 row in set (0.00 sec)
```

| h_id | h_name | h_address |
|------|-----------|-----------|
| 101 | unique | 1001 |
| 102 | free_star | 1002 |
| 103 | super | 1003 |

3 NF OF HOTEL_BRANCH

```
mysql> select h_name, h_address from hotel_branch;
```

| h_name | h_address |
|-----------|-----------|
| unique | 1001 |
| free_star | 1002 |
| super | 1003 |

3 rows in set (0.00 sec)

CREATE STAFF TABLE

```
mysql> create table staff(h_id int(5) primary key, s_name varchar(20), s_address varchar(20), s_role varchar(10), s_id int(5));
Query OK, 0 rows affected, 3 warnings (0.04 sec)

mysql> describe staff;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| h_id | int | NO | PK | NULL | |
| s_name | varchar(20) | YES | | NULL | |
| s_address | varchar(20) | YES | | NULL | |
| s_role | varchar(10) | YES | | NULL | |
| s_id | int | YES | | NULL | |

1 row in set (0.00 sec)

INSERTING VALUES IN STAFF

[illegible]

usually select = from staff

| id | name | country | city | ch |
|----|-------|---------|------------|----|
| 1 | John | USA | London | 10 |
| 2 | David | Belgium | London | 20 |
| 3 | James | Japan | Washington | 30 |

time in sec (0.00 sec)

3RD NORMAL FORM OF STAFF

```

def infect(target, ip_address):
    # ...

```

```
mysql> select p_name,s_address,s_phone from staff;
```

| s_rate | s_address | s_title |
|--------|-----------|-------------|
| harsh | william | manager |
| afraid | shankar | worker |
| cohort | tanore | recognition |

POAM: (11/11/2017) 10:00

ADD s_id COLUMN IN HOTEL BRANCH TABLE

100% WITH table base) branch was a. in DPC(1)
 50% in 0 time affected, 1 warning (0.02 sec)
 100% in 0 time affected, 0 warnings:

WINTER: DISCOUNT WITH LUNCH!

| Field | Type | Key | Index | Default | Extra |
|----------------------|---------------------------|-----|-------|---------|-------|
| <code>id</code> | <code>int</code> | NO | YES | NULL | |
| <code>name</code> | <code>varchar(255)</code> | YES | | NULL | |
| <code>address</code> | <code>varchar(255)</code> | YES | | NULL | |
| <code>sex</code> | <code>int</code> | YES | | NULL | |

4:20PM '11 WED (0:00 AM)

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 (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe hotel_branch;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| s_id | int | YES | PRI | NULL | |
| s_name | varchar(25) | YES | | NULL | |
| s_address | varchar(100) | YES | | NULL | |
| staff_id | int | YES | | NULL | |
+-----+
1 row 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, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe hotel_branch;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| s_id | int | YES | PRI | NULL | |
| s_name | varchar(25) | YES | | NULL | |
| s_address | varchar(100) | YES | | NULL | |
| staff_id | int | YES | YES | NULL | |
+-----+
1 row in set (0.00 sec)
```

INSERTING VALUES IN FOREIGN KEY

```
mysql> insert into hotel_branch set staff_id=1 where s_id=1001;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> insert into hotel_branch set staff_id=2 where s_id=1002;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> insert into hotel_branch set staff_id=3 where s_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 | Tadash | 1 |
| 1002 | Five_star | Wafan | 2 |
| 1003 | Star | Shikharu | 3 |

```
3 rows in set (0.10 sec)
```

CREATE ROOM_BOOKING TABLE

```
mysql> create table room_booking(room_id INT(3) PRIMARY KEY, r_no INT(3), check_in_date DATETIME, check_out_date DATETIME, weekend_pricing INT(20), holiday_pricing INT(20), early_bird_offers INT(20));
```

```
Query OK, 0 rows affected, 1 warnings (0.04 sec)
```

```
mysql> describe room_booking;
```

| Field | Type | Null | Key | Default | Extra |
|-------------------|----------|------|-----|---------|-------|
| room_id | int | NO | PK | NULL | |
| r_no | int | YES | | NULL | |
| check_in_date | datetime | YES | | NULL | |
| check_out_date | datetime | YES | | NULL | |
| weekend_pricing | int | YES | | NULL | |
| holiday_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, 3 rows affected (0.11 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE room_booking MODIFY COLUMN check_out_date DATE;  
Query OK, 3 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE room_booking MODIFY COLUMN check_out_date DATE;  
Query OK, 3 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 | PK | 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 | int | YES | | NULL | |

```
3 rows in set (0.00 sec)
```

MODIFY EARLY_BIRD_OFFERS & INSERTING VALUES

```
mysql> ALTER table room_booking MODIFY COLUMN early_bird_offers varchar(20);
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> INSERT INTO room_booking values('10', '302', '2025-01-25', '2025-01-25', '8000', '8000', '40%');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO room_booking values('10', '304', '2025-02-24', '2025-02-24', '7000', '8000', '30%');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO room_booking values('10', '301', '2025-02-24', '2025-02-24', '8000', '12000', '30%');
```

```
Query OK, 1 row affected (0.00 sec)
```

SELECT FROM ROOM_BOOKING

```
mysql> select * from room_booking;
```

| room_id | r_no | check_in_date | check_out_date | weekend_pricing | holiday_pricing | early_bird_offers |
|---------|------|---------------|----------------|-----------------|-----------------|-------------------|
| 10 | 302 | 2025-01-25 | 2025-01-25 | 8000 | 8000 | 40% |
| 10 | 304 | 2025-02-24 | 2025-02-24 | 7000 | 8000 | 30% |
| 10 | 301 | 2025-02-24 | 2025-02-24 | 8000 | 12000 | 30% |

```
3 rows in set (0.00 sec)
```

ADD STAFF_ID FOREIGN KEY IN ROOM_BOOKING

```
mysql> ALTER table room_booking ADD FOREIGN KEY (staff_id) REFERENCES staffs(id);
Query OK, 1 row affected (0.12 sec)
Records: 1 Duplicates: 0 Warnings: 0
```

```
mysql> describe room_booking;
```

| Field | Type | Null | Key | Default | Extra |
|--------------------|--------------|------|-----|---------|-------|
| room_id | int | NO | PRY | 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_check_offers | varchar(255) | YES | | NULL | |
| staff_id | int | YES | FK | NULL | |

```
1 row 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 | holiday_pricing | early_check_offers | staff_id |
|---------|------|---------------|----------------|-----------------|-----------------|--------------------|----------|
| 10 | 332 | 2025-01-01 | 2025-01-25 | 1000 | 9000 | 45 | 1 |
| 20 | 324 | 2025-02-04 | 2025-02-20 | 7000 | 8000 | 120 | 2 |
| 30 | 331 | 2025-03-04 | 2025-03-24 | 5000 | 12000 | 100 | 3 |

```
3 rows in set (0.20 sec)
```

3 NF OF ROOM_BOOKING

```
mysql> select r.rm, weekend_pricing, holiday_pricing, early_bird_offers from room_booking;
ERROR 1054 (42S22): Unknown column 'holiday_pricing' in 'field list'
mysql> select r.rm, weekend_pricing, holiday_pricing, early_bird_offers from room_booking;
```

| r_rm | weekend_pricing | holiday_pricing | early_bird_offers |
|------|-----------------|-----------------|-------------------|
| 222 | 8000 | 9000 | 40% |
| 322 | 7000 | 8000 | 30% |
| 121 | 8000 | 12000 | 30% |

```
1 row in set (0.01 sec)
```

MAKE ROOM_ID FOREIGN KEY IN STAFF TABLE

```
mysql> ALTER table staff add foreign key (r_id) references room_booking (room_id);
Query OK, 0 rows affected (0.14 sec)
Warnings: 1: Ignored: 0, warnings: 0
```

```
mysql> describe staff;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| r_id | int | NO | YES | NULL | |
| s_name | varchar(20) | YES | | NULL | |
| s_address | varchar(20) | YES | | NULL | |
| s_role | varchar(20) | YES | | NULL | |
| s_id | int | YES | NO | NULL | |

```
1 row in set (0.00 sec)
```

```
mysql>
```

CREATE CUSTOMER TABLE

```
mysql> create table customer(c_id INT(10) primary key, c_name varchar(30), c_dof DATE);
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> describe customer;
```

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|-------|
| c_id | int | NO | YES | NULL | |
| c_name | varchar(30) | YES | | NULL | |
| c_dof | date | YES | | NULL | |

```
1 row in set (0.00 sec)
```

ADD MISSING C_ADDRESS COLUMN IN CUSTOMER TABLE

```
mysql> ALTER table customer ADD c_address varchar(30);
Query OK, 2 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe customer;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| c_id | int | NO | YES | NULL | |
| c_name | varchar(15) | YES | | NULL | |
| c_id | int | YES | | NULL | |
| c_address | varchar(30) | YES | | NULL | |

```
1 row in set (0.00 sec)
```

INSERTING VALUES

```
mysql> INSERT INTO customer VALUES (10, 'sonu', '12', 'Ramnagar');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO customer VALUES (20, 'sonu', '20', 'Ramnagar'), (30, 'sonu', '30', 'Ramnagar');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> select * from student;
```

```
ERROR 1146 (4200): Table 'test/student' doesn't exist
```

```
mysql> select * from customer;
```

| c_id | c_name | c_id | c_address |
|------|--------|------|-----------|
| 10 | sonu | 12 | Ramnagar |
| 20 | sonu | 20 | Ramnagar |
| 30 | sonu | 30 | Ramnagar |

```
3 rows in set (0.00 sec)
```

MAKE r_id FOREIGN KEY IN CUSTOMER TABLE

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

```
mysql> describe customer;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| c_id | int | NO | YES | NULL | |
| c_name | varchar(15) | YES | | NULL | |
| c_id | int | YES | NO | NULL | |
| c_address | varchar(30) | YES | | NULL | |

```
1 row in set (0.00 sec)
```


CREATE LOYALTY POINTS TABLE

```
mysql> create table loyalty_points(point_id int(10) PRIMARY KEY, service_points int(5), upgrade_points int(5), cus_id int(5));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> describe loyalty_points;
```

| Field | Type | Null | Key | Default | Extra |
|----------------|------|------|-----|---------|-------|
| point_id | int | NO | PRY | 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 (cus_id);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe loyalty_points;
```

| Field | Type | Null | Key | Default | Extra |
|----------------|------|------|-----|---------|-------|
| point_id | int | NO | PRY | NULL | |
| service_points | int | YES | | NULL | |
| upgrade_points | int | YES | | NULL | |
| cus_id | int | YES | FK | 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('258','500','100','200'),('310','400','100','350');
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 |
| 258 | 500 | 100 | 200 |
| 310 | 400 | 50 | 350 |

```
3 rows in set (0.00 sec)
```

CREATE FEEDBACK TABLE

```
mysql> create table feedback(fb_id int(11) primary key, fb_type varchar(20), comment varchar(200), customer_id int(11));
Query OK, 0 rows affected, 2 warnings (0.04 sec)
```

```
mysql> describe feedback;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|--------------|------|-----|---------|-------|
| fb_id | int | NO | PK | NUL | |
| fb_type | varchar(20) | YES | | NUL | |
| comment | varchar(200) | YES | | NUL | |
| customer_id | int | YES | | NUL | |

```
1 row in set (0.00 sec)
```

MAKE customer_id FOREIGN KEY IN FEEDBACK TABLE

```
mysql> INSERT INTO feedback values('101','complaint','great food','100');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO feedback values('102','complaint','late services','100');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO feedback values('103','complaint','good services','100');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from feedback;
```

| fb_id | fb_type | comment | customer_id |
|-------|-----------|---------------|-------------|
| 101 | complaint | late services | 100 |
| 102 | complaint | great food | 100 |
| 103 | complaint | 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)
```

```
Warnings: 0 Duplicated: 0 Warnings: 0
```

```
mysql> describe feedback;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|--------------|------|-----|---------|-------|
| fb_id | int | NO | PK | NUL | |
| fb_type | varchar(20) | YES | | NUL | |
| comment | varchar(200) | YES | | NUL | |
| customer_id | int | YES | FK | NUL | |

```
1 row in set (0.00 sec)
```


CREATE PRE_ORDER TABLE

```
mysql> create table pre_order(order_id int(50) 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 pre_order;
```

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| order_id | int | NO | PR | NULL | |
| meal | varchar(40) | YES | | NULL | |
| taxi_services | varchar(10) | YES | | NULL | |
| cus_id | int | YES | | NULL | |

```
4 rows in set (0.00 sec)
```

MADE cus_id FOREIGN KEY

```
mysql> alter table pre_order add FOREIGN KEY (cus_id) REFERENCES customer (c_id);
```

```
Query OK, 0 rows affected (0.13 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe pre_order;
```

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| order_id | int | NO | PR | NULL | |
| meal | varchar(40) | YES | | NULL | |
| taxi_services | varchar(10) | YES | | NULL | |
| cus_id | int | YES | FK | NULL | |

```
4 rows in set (0.00 sec)
```

INSERTING VALUES IN ORDER

```
mysql> insert into pre_order values(110,'breakfast','yes',100);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into pre_order values(120,'dinner','no',100);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into pre_order values(130,'lunch','yes',100);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from pre_order;
```

| order_id | meal | taxi_services | cus_id |
|----------|-----------|---------------|--------|
| 110 | breakfast | yes | 100 |
| 120 | dinner | no | 100 |
| 130 | lunch | yes | 100 |

```
3 rows in set (0.00 sec)
```


PRACTICAL IMPLEMENTATION OF MYSQL QUERIES

ORDER BY CLAUSE ON CUSTOMER

```
SQL> select * from emp;
EMPID EMPNAME EMPJOB EMPDEPTNO EMPMGR EMPHIRE EMPSAL
-----
1001 SCOTT SCOTT 10 1000 1980 5000
1002 JAMES JAMES 10 1000 1981 3000
1003 ALLEN ALLEN 10 1000 1982 3500
1004 WATSON WATSON 10 1000 1983 2800
1005 MARTIN MARTIN 10 1000 1984 4000
1006 MANSOUR MANSOUR 10 1000 1985 3200
1007 JESSIE JESSIE 10 1000 1986 2500
1008 DEAN DEAN 10 1000 1987 3800
1009 FORD FORD 10 1000 1988 4500
1010 SMITH SMITH 10 1000 1989 2000
1011 COOPER COOPER 10 1000 1990 2200
1012 BATES BATES 10 1000 1991 2400
1013 JONES JONES 10 1000 1992 2600
1014 GARCIA GARCIA 10 1000 1993 2800
1015 ROY ROY 10 1000 1994 3000
1016 LEE LEE 10 1000 1995 3200
1017 PERKINS PERKINS 10 1000 1996 3400
1018 TAYLOR TAYLOR 10 1000 1997 3600
1019 WILSON WILSON 10 1000 1998 3800
1020 GIBSON GIBSON 10 1000 1999 4000
1021 HARRIS HARRIS 10 1000 2000 4200
1022 KING KING 10 1000 2001 4400
1023 QUINN QUINN 10 1000 2002 4600
1024 STEVEN STEVEN 10 1000 2003 4800
1025 JORDAN JORDAN 10 1000 2004 5000
1026 COLEMAN COLEMAN 10 1000 2005 5200
1027 BAKER BAKER 10 1000 2006 5400
1028 NELSON NELSON 10 1000 2007 5600
1029 HILL HILL 10 1000 2008 5800
1030 GAY GAY 10 1000 2009 6000
1031 FLOYD FLOYD 10 1000 2010 6200
1032 WALKER WALKER 10 1000 2011 6400
1033 LYNN LYNN 10 1000 2012 6600
1034 TOLSON TOLSON 10 1000 2013 6800
1035 GORDON GORDON 10 1000 2014 7000
1036 ROSS ROSS 10 1000 2015 7200
1037 PATE PATE 10 1000 2016 7400
1038 HUGHES HUGHES 10 1000 2017 7600
1039 FLETCHER FLETCHER 10 1000 2018 7800
1040 SIMS SIMS 10 1000 2019 8000
1041 STEVENSON STEVENSON 10 1000 2020 8200
1042 HARRISON HARRISON 10 1000 2021 8400
1043 GIBSON GIBSON 10 1000 2022 8600
1044 ROY ROY 10 1000 2023 8800
1045 LEE LEE 10 1000 2024 9000
1046 PERKINS PERKINS 10 1000 2025 9200
1047 TAYLOR TAYLOR 10 1000 2026 9400
1048 WILSON WILSON 10 1000 2027 9600
1049 GIBSON GIBSON 10 1000 2028 9800
1050 HARRIS HARRIS 10 1000 2029 10000
1051 KING KING 10 1000 2030 10200
1052 QUINN QUINN 10 1000 2031 10400
1053 STEVEN STEVEN 10 1000 2032 10600
1054 JORDAN JORDAN 10 1000 2033 10800
1055 COLEMAN COLEMAN 10 1000 2034 11000
1056 BAKER BAKER 10 1000 2035 11200
1057 NELSON NELSON 10 1000 2036 11400
1058 HILL HILL 10 1000 2037 11600
1059 GAY GAY 10 1000 2038 11800
1060 FLOYD FLOYD 10 1000 2039 12000
1061 WALKER WALKER 10 1000 2040 12200
1062 LYNN LYNN 10 1000 2041 12400
1063 TOLSON TOLSON 10 1000 2042 12600
1064 GORDON GORDON 10 1000 2043 12800
1065 ROSS ROSS 10 1000 2044 13000
1066 PATE PATE 10 1000 2045 13200
1067 HUGHES HUGHES 10 1000 2046 13400
1068 FLETCHER FLETCHER 10 1000 2047 13600
1069 SIMS SIMS 10 1000 2048 13800
1070 STEVENSON STEVENSON 10 1000 2049 14000
1071 HARRISON HARRISON 10 1000 2050 14200
1072 GIBSON GIBSON 10 1000 2051 14400
1073 ROY ROY 10 1000 2052 14600
1074 LEE LEE 10 1000 2053 14800
1075 PERKINS PERKINS 10 1000 2054 15000
1076 TAYLOR TAYLOR 10 1000 2055 15200
1077 WILSON WILSON 10 1000 2056 15400
1078 GIBSON GIBSON 10 1000 2057 15600
1079 HARRIS HARRIS 10 1000 2058 15800
1080 KING KING 10 1000 2059 16000
1081 QUINN QUINN 10 1000 2060 16200
1082 STEVEN STEVEN 10 1000 2061 16400
1083 JORDAN JORDAN 10 1000 2062 16600
1084 COLEMAN COLEMAN 10 1000 2063 16800
1085 BAKER BAKER 10 1000 2064 17000
1086 NELSON NELSON 10 1000 2065 17200
1087 HILL HILL 10 1000 2066 17400
1088 GAY GAY 10 1000 2067 17600
1089 FLOYD FLOYD 10 1000 2068 17800
1090 WALKER WALKER 10 1000 2069 18000
1091 LYNN LYNN 10 1000 2070 18200
1092 TOLSON TOLSON 10 1000 2071 18400
1093 GORDON GORDON 10 1000 2072 18600
1094 ROSS ROSS 10 1000 2073 18800
1095 PATE PATE 10 1000 2074 19000
1096 HUGHES HUGHES 10 1000 2075 19200
1097 FLETCHER FLETCHER 10 1000 2076 19400
1098 SIMS SIMS 10 1000 2077 19600
1099 STEVENSON STEVENSON 10 1000 2078 19800
1100 HARRISON HARRISON 10 1000 2079 20000
1101 GIBSON GIBSON 10 1000 2080 20200
1102 ROY ROY 10 1000 2081 20400
1103 LEE LEE 10 1000 2082 20600
1104 PERKINS PERKINS 10 1000 2083 20800
1105 TAYLOR TAYLOR 10 1000 2084 21000
1106 WILSON WILSON 10 1000 2085 21200
1107 GIBSON GIBSON 10 1000 2086 21400
1108 HARRIS HARRIS 10 1000 2087 21600
1109 KING KING 10 1000 2088 21800
1110 QUINN QUINN 10 1000 2089 22000
1111 STEVEN STEVEN 10 1000 2090 22200
1112 JORDAN JORDAN 10 1000 2091 22400
1113 COLEMAN COLEMAN 10 1000 2092 22600
1114 BAKER BAKER 10 1000 2093 22800
1115 NELSON NELSON 10 1000 2094 23000
1116 HILL HILL 10 1000 2095 23200
1117 GAY GAY 10 1000 2096 23400
1118 FLOYD FLOYD 10 1000 2097 23600
1119 WALKER WALKER 10 1000 2098 23800
1120 LYNN LYNN 10 1000 2099 24000
1121 TOLSON TOLSON 10 1000 2100 24200
1122 GORDON GORDON 10 1000 2101 24400
1123 ROSS ROSS 10 1000 2102 24600
1124 PATE PATE 10 1000 2103 24800
1125 HUGHES HUGHES 10 1000 2104 25000
1126 FLETCHER FLETCHER 10 1000 2105 25200
1127 SIMS SIMS 10 1000 2106 25400
1128 STEVENSON STEVENSON 10 1000 2107 25600
1129 HARRISON HARRISON 10 1000 2108 25800
1130 GIBSON GIBSON 10 1000 2109 26000
1131 ROY ROY 10 1000 2110 26200
1132 LEE LEE 10 1000 2111 26400
1133 PERKINS PERKINS 10 1000 2112 26600
1134 TAYLOR TAYLOR 10 1000 2113 26800
1135 WILSON WILSON 10 1000 2114 27000
1136 GIBSON GIBSON 10 1000 2115 27200
1137 HARRIS HARRIS 10 1000 2116 27400
1138 KING KING 10 1000 2117 27600
1139 QUINN QUINN 10 1000 2118 27800
1140 STEVEN STEVEN 10 1000 2119 28000
1141 JORDAN JORDAN 10 1000 2120 28200
1142 COLEMAN COLEMAN 10 1000 2121 28400
1143 BAKER BAKER 10 1000 2122 28600
1144 NELSON NELSON 10 1000 2123 28800
1145 HILL HILL 10 1000 2124 29000
1146 GAY GAY 10 1000 2125 29200
1147 FLOYD FLOYD 10 1000 2126 29400
1148 WALKER WALKER 10 1000 2127 29600
1149 LYNN LYNN 10 1000 2128 29800
1150 TOLSON TOLSON 10 1000 2129 30000
1151 GORDON GORDON 10 1000 2130 30200
1152 ROSS ROSS 10 1000 2131 30400
1153 PATE PATE 10 1000 2132 30600
1154 HUGHES HUGHES 10 1000 2133 30800
1155 FLETCHER FLETCHER 10 1000 2134 31000
1156 SIMS SIMS 10 1000 2135 31200
1157 STEVENSON STEVENSON 10 1000 2136 31400
1158 HARRISON HARRISON 10 1000 2137 31600
1159 GIBSON GIBSON 10 1000 2138 31800
1160 ROY ROY 10 1000 2139 32000
1161 LEE LEE 10 1000 2140 32200
1162 PERKINS PERKINS 10 1000 2141 32400
1163 TAYLOR TAYLOR 10 1000 2142 32600
1164 WILSON WILSON 10 1000 2143 32800
1165 GIBSON GIBSON 10 1000 2144 33000
1166 HARRIS HARRIS 10 1000 2145 33200
1167 KING KING 10 1000 2146 33400
1168 QUINN QUINN 10 1000 2147 33600
1169 STEVEN STEVEN 10 1000 2148 33800
1170 JORDAN JORDAN 10 1000 2149 34000
1171 COLEMAN COLEMAN 10 1000 2150 34200
1172 BAKER BAKER 10 1000 2151 34400
1173 NELSON NELSON 10 1000 2152 34600
1174 HILL HILL 10 1000 2153 34800
1175 GAY GAY 10 1000 2154 35000
1176 FLOYD FLOYD 10 1000 2155 35200
1177 WALKER WALKER 10 1000 2156 35400
1178 LYNN LYNN 10 1000 2157 35600
1179 TOLSON TOLSON 10 1000 2158 35800
1180 GORDON GORDON 10 1000 2159 36000
1181 ROSS ROSS 10 1000 2160 36200
1182 PATE PATE 10 1000 2161 36400
1183 HUGHES HUGHES 10 1000 2162 36600
1184 FLETCHER FLETCHER 10 1000 2163 36800
1185 SIMS SIMS 10 1000 2164 37000
1186 STEVENSON STEVENSON 10 1000 2165 37200
1187 HARRISON HARRISON 10 1000 2166 37400
1188 GIBSON GIBSON 10 1000 2167 37600
1189 ROY ROY 10 1000 2168 37800
1190 LEE LEE 10 1000 2169 38000
1191 PERKINS PERKINS 10 1000 2170 38200
1192 TAYLOR TAYLOR 10 1000 2171 38400
1193 WILSON WILSON 10 1000 2172 38600
1194 GIBSON GIBSON 10 1000 2173 38800
1195 HARRIS HARRIS 10 1000 2174 39000
1196 KING KING 10 1000 2175 39200
1197 QUINN QUINN 10 1000 2176 39400
1198 STEVEN STEVEN 10 1000 2177 39600
1199 JORDAN JORDAN 10 1000 2178 39800
1200 COLEMAN COLEMAN 10 1000 2179 40000
1201 BAKER BAKER 10 1000 2180 40200
1202 NELSON NELSON 10 1000 2181 40400
1203 HILL HILL 10 1000 2182 40600
1204 GAY GAY 10 1000 2183 40800
1205 FLOYD FLOYD 10 1000 2184 41000
1206 WALKER WALKER 10 1000 2185 41200
1207 LYNN LYNN 10 1000 2186 41400
1208 TOLSON TOLSON 10 1000 2187 41600
1209 GORDON GORDON 10 1000 2188 41800
1210 ROSS ROSS 10 1000 2189 42000
1211 PATE PATE 10 1000 2190 42200
1212 HUGHES HUGHES 10 1000 2191 42400
1213 FLETCHER FLETCHER 10 1000 2192 42600
1214 SIMS SIMS 10 1000 2193 42800
1215 STEVENSON STEVENSON 10 1000 2194 43000
1216 HARRISON HARRISON 10 1000 2195 43200
1217 GIBSON GIBSON 10 1000 2196 43400
1218 ROY ROY 10 1000 2197 43600
1219 LEE LEE 10 1000 2198 43800
1220 PERKINS PERKINS 10 1000 2199 44000
1221 TAYLOR TAYLOR 10 1000 2200 44200
1222 WILSON WILSON 10 1000 2201 44400
1223 GIBSON GIBSON 10 1000 2202 44600
1224 HARRIS HARRIS 10 1000 2203 44800
1225 KING KING 10 1000 2204 45000
1226 QUINN QUINN 10 1000 2205 45200
1227 STEVEN STEVEN 10 1000 2206 45400
1228 JORDAN JORDAN 10 1000 2207 45600
1229 COLEMAN COLEMAN 10 1000 2208 45800
1230 BAKER BAKER 10 1000 2209 46000
1231 NELSON NELSON 10 1000 2210 46200
1232 HILL HILL 10 1000 2211 46400
1233 GAY GAY 10 1000 2212 46600
1234 FLOYD FLOYD 10 1000 2213 46800
1235 WALKER WALKER 10 1000 2214 47000
1236 LYNN LYNN 10 1000 2215 47200
1237 TOLSON TOLSON 10 1000 2216 47400
1238 GORDON GORDON 10 1000 2217 47600
1239 ROSS ROSS 10 1000 2218 47800
1240 PATE PATE 10 1000 2219 48000
1241 HUGHES HUGHES 10 1000 2220 48200
1242 FLETCHER FLETCHER 10 1000 2221 48400
1243 SIMS SIMS 10 1000 2222 48600
1244 STEVENSON STEVENSON 10 1000 2223 48800
1245 HARRISON HARRISON 10 1000 2224 49000
1246 GIBSON GIBSON 10 1000 2225 49200
1247 ROY ROY 10 1000 2226 49400
1248 LEE LEE 10 1000 2227 49600
1249 PERKINS PERKINS 10 1000 2228 49800
1250 TAYLOR TAYLOR 10 1000 2229 50000
1251 WILSON WILSON 10 1000 2230 50200
1252 GIBSON GIBSON 10 1000 2231 50400
1253 HARRIS HARRIS 10 1000 2232 50600
1254 KING KING 10 1000 2233 50800
1255 QUINN QUINN 10 1000 2234 51000
1256 STEVEN STEVEN 10 1000 2235 51200
1257 JORDAN JORDAN 10 1000 2236 51400
1258 COLEMAN COLEMAN 10 1000 2237 51600
1259 BAKER BAKER 10 1000 2238 51800
1260 NELSON NELSON 10 1000 2239 52000
1261 HILL HILL 10 1000 2240 52200
1262 GAY GAY 10 1000 2241 52400
1263 FLOYD FLOYD 10 1000 2242 52600
1264 WALKER WALKER 10 1000 2243 52800
1265 LYNN LYNN 10 1000 2244 53000
1266 TOLSON TOLSON 10 1000 2245 53200
1267 GORDON GORDON 10 1000 2246 53400
1268 ROSS ROSS 10 1000 2247 53600
1269 PATE PATE 10 1000 2248 53800
1270 HUGHES HUGHES 10 1000 2249 54000
1271 FLETCHER FLETCHER 10 1000 2250 54200
1272 SIMS SIMS 10 1000 2251 54400
1273 STEVENSON STEVENSON 10 1000 2252 54600
1274 HARRISON HARRISON 10 1000 2253 54800
1275 GIBSON GIBSON 10 1000 2254 55000
1276 ROY ROY 10 1000 2255 55200
1277 LEE LEE 10 1000 2256 55400
1278 PERKINS PERKINS 10 1000 2257 55600
1279 TAYLOR TAYLOR 10 1000 2258 55800
1280 WILSON WILSON 10 1000 2259 56000
1281 GIBSON GIBSON 10 1000 2260 56200
1282 HARRIS HARRIS 10 1000 2261 56400
1283 KING KING 10 1000 2262 56600
1284 QUINN QUINN 10 1000 2263 56800
1285 STEVEN STEVEN 10 1000 2264 57000
1286 JORDAN JORDAN 10 1000 2265 57200
1287 COLEMAN COLEMAN 10 1000 2266 57400
1288 BAKER BAKER 10 1000 2267 57600
1289 NELSON NELSON 10 1000 2268 57800
1290 HILL HILL 10 1000 2269 58000
1291 GAY GAY 10 1000 2270 58200
1292 FLOYD FLOYD 10 1000 2271 58400
1293 WALKER WALKER 10 1000 2272 58600
1294 LYNN LYNN 10 1000 2273 58800
1295 TOLSON TOLSON 10 1000 2274 59000
1296 GORDON GORDON 10 1000 2275 59200
1297 ROSS ROSS 10 1000 2276 59400
1298 PATE PATE 10 1000 2277 59600
1299 HUGHES HUGHES 10 1000 2278 59800
1300 FLETCHER FLETCHER 10 1000 2279 60000
1301 SIMS SIMS 10 1000 2280 60200
1302 STEVENSON STEVENSON 10 1000 2281 60400
1303 HARRISON HARRISON 10 1000 2282 60600
1304 GIBSON GIBSON 10 1000 2283 60800
1305 ROY ROY 10 1000 2284 61000
1306 LEE LEE 10 1000 2285 61200
1307 PERKINS PERKINS 10 1000 2286 61400
1308 TAYLOR TAYLOR 10 1000 2287 61600
1309 WILSON WILSON 10 1000 2288 61800
1310 GIBSON GIBSON 10 1000 2289 62000
1311 HARRIS HARRIS 10 1000 2290 62200
1312 KING KING 10 1000 2291 62400
1313 QUINN QUINN 10 1000 2292 62600
1314 STEVEN STEVEN 10 1000 2293 62800
1315 JORDAN JORDAN 10 1000 2294 63000
1316 COLEMAN COLEMAN 10 1000 2295 63200
1317 BAKER BAKER 10 1000 2296 63400
1318 NELSON NELSON 10 1000 2297 63600
1319 HILL HILL 10 1000 2298 63800
1320 GAY GAY 10 1000 2299 64000
1321 FLOYD FLOYD 10 1000 2300 64200
1322 WALKER WALKER 10 1000 2301 64400
1323 LYNN LYNN 10 1000 2302 64600
1324 TOLSON TOLSON 10 1000 2303 64800
1325 GORDON GORDON 10 1000 2304 65000
1326 ROSS ROSS 10 1000 2305 65200
1327 PATE PATE 10 1000 2306 65400
1328 HUGHES HUGHES 10 1000 2307 65600
1329 FLETCHER FLETCHER 10 1000 2308 65800
1330 SIMS SIMS 10 1000 2309 66000
1331 STEVENSON STEVENSON 10 1000 2310 66200
1332 HARRISON HARRISON 10 1000 2311 66400
1333 GIBSON GIBSON 10 1000 2312 66600
1334 ROY ROY 10 1000 2313 66800
1335 LEE LEE 10 1000 2314 67000
1336 PERKINS PERKINS 10 1000 2315 67200
1337 TAYLOR TAYLOR 10 1000 2316 67400
1338 WILSON WILSON 10 1000 2317 67600
1339 GIBSON GIBSON 10 1000 2318 67800
1340 HARRIS HARRIS 10 1000 2319 68000
1341 KING KING 10 1000 2320 68200
1342 QUINN QUINN 10 1000 2321 68400
1343 STEVEN STEVEN 10 1000 2322 68600
1344 JORDAN JORDAN 10 1000 2323 68800
1345 COLEMAN COLEMAN 10 1000 2324 69000
1346 BAKER BAKER 10 1000 2325 69200
1347 NELSON NELSON 10 1000 2326 69400
1348 HILL HILL 10 1000 2327 69600
1349 GAY GAY 10 1000 2328 69800
1350 FLOYD FLOYD 10 1000 2329 70000
1351 WALKER WALKER 10 1000 2330 70200
1352 LYNN LYNN 10 1000 2331 70400
1353 TOLSON TOLSON 10 1000 2332 70600
1354 GORDON GORDON 10 1000 2333 70800
1355 ROSS ROSS 10 1000 2334 71000
1356 PATE PATE 10 1000 2335 71200
1357 HUGHES HUGHES 10 1000 2336 71400
1358 FLETCHER FLETCHER 10 1000 2337 71600
1359 SIMS SIMS 10 1000 2338 71800
1360 STEVENSON STEVENSON 10 1000 2339 72000
1361 HARRISON HARRISON 10 1000 2340 72200
1362 GIBSON GIBSON 10 1000 2341 72400
1363 ROY ROY 10 1000 2342 72600
1364 LEE LEE 10 1000 2343 72800
1365 PERKINS PERKINS 10 1000 2344 73000
1366 TAYLOR TAYLOR 10 1000 2345 73200
1367 WILSON WILSON 10 1000 2346 73400
1368 GIBSON GIBSON 10 1000 2347 73600
1369 HARRIS HARRIS 10 1000 2348 73800
1370 KING KING 10 1000 2349 74000
1371 QUINN QUINN 10 1000 2350 74200
1372 STEVEN STEVEN 10 1000 2351 74400
1373 JORDAN JORDAN 10 1000 2352 74600
1374 COLEMAN COLEMAN 10 1000 2353 74800
1375 BAKER BAKER 10 1000 2354 75000
1376 NELSON NELSON 10 1000 2355 75200
1377 HILL HILL 10 1000 2356 75400
1378 GAY GAY 10 1000 2357 75600
1379 FLOYD FLOYD 10 1000 2358 75800
1380 WALKER WALKER 10 1000 2359 76000
1381 LYNN LYNN 10 1000 2360 76200
1382 TOLSON TOLSON 10 1000 2361 76400
1383 GORDON GORDON 10 1000 2362 76600
1384 ROSS ROSS 10 1000 2363 76800
1385 PATE PATE 10 1000 2364 77000
1386 HUGHES HUGHES 10 1000 2365 77200
1387 FLETCHER FLETCHER 10 1000 2366 77400
1388 SIMS SIMS 10 1000 2367 77600
1389 STEVENSON STEVENSON 10 1000 2368 77800
1390 HARRISON HARRISON 10 1000 2369 78000
1391 GIBSON GIBSON 10 1000 2370 78200
1392 ROY ROY 10 1000 2371 78400
1393 LEE LEE 10 1000 2372 78600
1394 PERKINS PERKINS 10 1000 2373 78800
1395 TAYLOR TAYLOR 10 1000 2374 79000
1396 WILSON WILSON 10 1000 2375 79200
1397 GIBSON GIBSON 10 1000 2376 79400
1398 HARRIS HARRIS 10 1000 2377 79600
1399 KING KING 10 1000 2378 79800
1400 QUINN QUINN 10 1000 2379 80000
1401 STEVEN STEVEN 10 1000 2380 80200
1402 JORDAN JORDAN 10 1000 2381 80400
1403 COLEMAN COLEMAN 10 1000 2382 80600
1404 BAKER BAKER 10 1000 2383 80800
1405 NELSON NELSON 10 1000 2384 81000
1406 HILL HILL 10 1000 2385 81200
1407 GAY GAY 10 1000 2386 81400
1408 FLOYD FLOYD 10 1000 2387 81600
1409 WALKER WALKER 10 1000 2388 81800
1410 LYNN LYNN 10 1000 2389 82000
1411 TOLSON TOLSON 10 1000 2390 82200
1412 GORDON GORDON 10 1000 2391 82400
1413 ROSS ROSS 10 1000 2392 82600
1414 PATE PATE 10 1000 2393 82800
1415 HUGHES HUGHES 10 1000 2394 83000
1416 FLETCHER FLETCHER 10 1000 2395 83200
1417 SIMS SIMS 10 1000 2396 83400
1418 STEVENSON STEVENSON 10 1000 2397 83600
1419 HARRISON HARRISON 10 1000 2398 83800
1420 GIBSON GIBSON 10 1000 2399 84000
1421 ROY ROY 10 1000 2400 84200
1422 LEE LEE 10 1000 2401 84400
1423 PERKINS PERKINS 10 1000 2402 84600

```

DISTINCT CLAUSE AND ADDING SALARY COLUMN IN STAFF TABLE

The screenshot shows a Windows 10 desktop environment. A Notepad++ window is open, displaying a C++ program. The program defines a 2D array named 'matrix' with 3 rows and 4 columns. It initializes the array with the following values:

| Row | Col 1 | Col 2 | Col 3 | Col 4 |
|-----|-------|-------|-------|-------|
| 1 | 100 | 200 | 300 | 400 |
| 2 | 500 | 600 | 700 | 800 |
| 3 | 900 | 800 | 700 | 600 |

The program then prints the matrix, and the output is visible in the console window at the bottom of the Notepad++ interface. The taskbar at the bottom shows various application icons, including the Start button, File Explorer, and several open applications. The system clock in the bottom right corner shows the date as 10/10/2023 and the time as 10:10 AM.

BETWEEN, HAVING AND IN CLAUSES

```

1 create table emp (emp_id number(4) primary key, emp_name varchar2(30), emp_salary number(8,2), emp_dept varchar2(10));
2 insert into emp values (1, 'John', 12000, 'IT');
3 insert into emp values (2, 'James', 9000, 'IT');
4 insert into emp values (3, 'Alex', 15000, 'IT');
5 insert into emp values (4, 'Ravi', 8000, 'IT');
6 insert into emp values (5, 'Priya', 10000, 'HR');
7 insert into emp values (6, 'Neha', 11000, 'HR');
8 insert into emp values (7, 'Anil', 7000, 'HR');
9 insert into emp values (8, 'Smita', 9500, 'HR');
10 insert into emp values (9, 'Ajay', 6000, 'HR');
11 insert into emp values (10, 'Amit', 5000, 'HR');
12 insert into emp values (11, 'Vijay', 4000, 'HR');
13 insert into emp values (12, 'Adarsh', 3000, 'HR');
14 insert into emp values (13, 'Rishi', 2000, 'HR');
15 insert into emp values (14, 'Siddhi', 1000, 'HR');
16 insert into emp values (15, 'Arjun', 1500, 'HR');
17 insert into emp values (16, 'Isha', 1200, 'HR');
18 insert into emp values (17, 'Nisha', 1100, 'HR');
19 insert into emp values (18, 'Harsh', 1000, 'HR');
20 insert into emp values (19, 'Rohit', 900, 'HR');
21 insert into emp values (20, 'Sanya', 800, 'HR');
22 insert into emp values (21, 'Nikhil', 700, 'HR');
23 insert into emp values (22, 'Anshu', 600, 'HR');
24 insert into emp values (23, 'Geeta', 500, 'HR');
25 insert into emp values (24, 'Divya', 400, 'HR');
26 insert into emp values (25, 'Aditya', 300, 'HR');
27 insert into emp values (26, 'Nisha', 200, 'HR');
28 insert into emp values (27, 'Harsh', 100, 'HR');
29 insert into emp values (28, 'Rohit', 50, 'HR');
30 insert into emp values (29, 'Sanya', 25, 'HR');
31 insert into emp values (30, 'Nikhil', 12, 'HR');
32 insert into emp values (31, 'Anshu', 6, 'HR');
33 insert into emp values (32, 'Geeta', 3, 'HR');
34 insert into emp values (33, 'Divya', 1, 'HR');
35 insert into emp values (34, 'Aditya', 0.5, 'HR');
36 insert into emp values (35, 'Nisha', 0.2, 'HR');
37 insert into emp values (36, 'Harsh', 0.1, 'HR');
38 insert into emp values (37, 'Rohit', 0.05, 'HR');
39 insert into emp values (38, 'Sanya', 0.02, 'HR');
40 insert into emp values (39, 'Nikhil', 0.01, 'HR');
41 insert into emp values (40, 'Anshu', 0.005, 'HR');
42 insert into emp values (41, 'Geeta', 0.002, 'HR');
43 insert into emp values (42, 'Divya', 0.001, 'HR');
44 insert into emp values (43, 'Aditya', 0.0005, 'HR');
45 insert into emp values (44, 'Nisha', 0.0002, 'HR');
46 insert into emp values (45, 'Harsh', 0.0001, 'HR');
47 insert into emp values (46, 'Rohit', 5e-05, 'HR');
48 insert into emp values (47, 'Sanya', 2e-05, 'HR');
49 insert into emp values (48, 'Nikhil', 1e-05, 'HR');
50 insert into emp values (49, 'Anshu', 5e-06, 'HR');
51 insert into emp values (50, 'Geeta', 2e-06, 'HR');
52 insert into emp values (51, 'Divya', 1e-06, 'HR');
53 insert into emp values (52, 'Aditya', 5e-07, 'HR');
54 insert into emp values (53, 'Nisha', 2e-07, 'HR');
55 insert into emp values (54, 'Harsh', 1e-07, 'HR');
56 insert into emp values (55, 'Rohit', 5e-08, 'HR');
57 insert into emp values (56, 'Sanya', 2e-08, 'HR');
58 insert into emp values (57, 'Nikhil', 1e-08, 'HR');
59 insert into emp values (58, 'Anshu', 5e-09, 'HR');
60 insert into emp values (59, 'Geeta', 2e-09, 'HR');
61 insert into emp values (60, 'Divya', 1e-09, 'HR');
62 insert into emp values (61, 'Aditya', 5e-10, 'HR');
63 insert into emp values (62, 'Nisha', 2e-10, 'HR');
64 insert into emp values (63, 'Harsh', 1e-10, 'HR');
65 insert into emp values (64, 'Rohit', 5e-11, 'HR');
66 insert into emp values (65, 'Sanya', 2e-11, 'HR');
67 insert into emp values (66, 'Nikhil', 1e-11, 'HR');
68 insert into emp values (67, 'Anshu', 5e-12, 'HR');
69 insert into emp values (68, 'Geeta', 2e-12, 'HR');
70 insert into emp values (69, 'Divya', 1e-12, 'HR');
71 insert into emp values (70, 'Aditya', 5e-13, 'HR');
72 insert into emp values (71, 'Nisha', 2e-13, 'HR');
73 insert into emp values (72, 'Harsh', 1e-13, 'HR');
74 insert into emp values (73, 'Rohit', 5e-14, 'HR');
75 insert into emp values (74, 'Sanya', 2e-14, 'HR');
76 insert into emp values (75, 'Nikhil', 1e-14, 'HR');
77 insert into emp values (76, 'Anshu', 5e-15, 'HR');
78 insert into emp values (77, 'Geeta', 2e-15, 'HR');
79 insert into emp values (78, 'Divya', 1e-15, 'HR');
80 insert into emp values (79, 'Aditya', 5e-16, 'HR');
81 insert into emp values (80, 'Nisha', 2e-16, 'HR');
82 insert into emp values (81, 'Harsh', 1e-16, 'HR');
83 insert into emp values (82, 'Rohit', 5e-17, 'HR');
84 insert into emp values (83, 'Sanya', 2e-17, 'HR');
85 insert into emp values (84, 'Nikhil', 1e-17, 'HR');
86 insert into emp values (85, 'Anshu', 5e-18, 'HR');
87 insert into emp values (86, 'Geeta', 2e-18, 'HR');
88 insert into emp values (87, 'Divya', 1e-18, 'HR');
89 insert into emp values (88, 'Aditya', 5e-19, 'HR');
90 insert into emp values (89, 'Nisha', 2e-19, 'HR');
91 insert into emp values (90, 'Harsh', 1e-19, 'HR');
92 insert into emp values (91, 'Rohit', 5e-20, 'HR');
93 insert into emp values (92, 'Sanya', 2e-20, 'HR');
94 insert into emp values (93, 'Nikhil', 1e-20, 'HR');
95 insert into emp values (94, 'Anshu', 5e-21, 'HR');
96 insert into emp values (95, 'Geeta', 2e-21, 'HR');
97 insert into emp values (96, 'Divya', 1e-21, 'HR');
98 insert into emp values (97, 'Aditya', 5e-22, 'HR');
99 insert into emp values (98, 'Nisha', 2e-22, 'HR');
100 insert into emp values (99, 'Harsh', 1e-22, 'HR');
101 insert into emp values (100, 'Rohit', 5e-23, 'HR');
102 insert into emp values (101, 'Sanya', 2e-23, 'HR');
103 insert into emp values (102, 'Nikhil', 1e-23, 'HR');
104 insert into emp values (103, 'Anshu', 5e-24, 'HR');
105 insert into emp values (104, 'Geeta', 2e-24, 'HR');
106 insert into emp values (105, 'Divya', 1e-24, 'HR');
107 insert into emp values (106, 'Aditya', 5e-25, 'HR');
108 insert into emp values (107, 'Nisha', 2e-25, 'HR');
109 insert into emp values (108, 'Harsh', 1e-25, 'HR');
110 insert into emp values (109, 'Rohit', 5e-26, 'HR');
111 insert into emp values (110, 'Sanya', 2e-26, 'HR');
112 insert into emp values (111, 'Nikhil', 1e-26, 'HR');
113 insert into emp values (112, 'Anshu', 5e-27, 'HR');
114 insert into emp values (113, 'Geeta', 2e-27, 'HR');
115 insert into emp values (114, 'Divya', 1e-27, 'HR');
116 insert into emp values (115, 'Aditya', 5e-28, 'HR');
117 insert into emp values (116, 'Nisha', 2e-28, 'HR');
118 insert into emp values (117, 'Harsh', 1e-28, 'HR');
119 insert into emp values (118, 'Rohit', 5e-29, 'HR');
120 insert into emp values (119, 'Sanya', 2e-29, 'HR');
121 insert into emp values (120, 'Nikhil', 1e-29, 'HR');
122 insert into emp values (121, 'Anshu', 5e-30, 'HR');
123 insert into emp values (122, 'Geeta', 2e-30, 'HR');
124 insert into emp values (123, 'Divya', 1e-30, 'HR');
125 insert into emp values (124, 'Aditya', 5e-31, 'HR');
126 insert into emp values (125, 'Nisha', 2e-31, 'HR');
127 insert into emp values (126, 'Harsh', 1e-31, 'HR');
128 insert into emp values (127, 'Rohit', 5e-32, 'HR');
129 insert into emp values (128, 'Sanya', 2e-32, 'HR');
130 insert into emp values (129, 'Nikhil', 1e-32, 'HR');
131 insert into emp values (130, 'Anshu', 5e-33, 'HR');
132 insert into emp values (131, 'Geeta', 2e-33, 'HR');
133 insert into emp values (132, 'Divya', 1e-33, 'HR');
134 insert into emp values (133, 'Aditya', 5e-34, 'HR');
135 insert into emp values (134, 'Nisha', 2e-34, 'HR');
136 insert into emp values (135, 'Harsh', 1e-34, 'HR');
137 insert into emp values (136, 'Rohit', 5e-35, 'HR');
138 insert into emp values (137, 'Sanya', 2e-35, 'HR');
139 insert into emp values (138, 'Nikhil', 1e-35, 'HR');
140 insert into emp values (139, 'Anshu', 5e-36, 'HR');
141 insert into emp values (140, 'Geeta', 2e-36, 'HR');
142 insert into emp values (141, 'Divya', 1e-36, 'HR');
143 insert into emp values (142, 'Aditya', 5e-37, 'HR');
144 insert into emp values (143, 'Nisha', 2e-37, 'HR');
145 insert into emp values (144, 'Harsh', 1e-37, 'HR');
146 insert into emp values (145, 'Rohit', 5e-38, 'HR');
147 insert into emp values (146, 'Sanya', 2e-38, 'HR');
148 insert into emp values (147, 'Nikhil', 1e-38, 'HR');
149 insert into emp values (148, 'Anshu', 5e-39, 'HR');
150 insert into emp values (149, 'Geeta', 2e-39, 'HR');
151 insert into emp values (150, 'Divya', 1e-39, 'HR');
152 insert into emp values (151, 'Aditya', 5e-40, 'HR');
153 insert into emp values (152, 'Nisha', 2e-40, 'HR');
154 insert into emp values (153, 'Harsh', 1e-40, 'HR');
155 insert into emp values (154, 'Rohit', 5e-41, 'HR');
156 insert into emp values (155, 'Sanya', 2e-41, 'HR');
157 insert into emp values (156, 'Nikhil', 1e-41, 'HR');
158 insert into emp values (157, 'Anshu', 5e-42, 'HR');
159 insert into emp values (158, 'Geeta', 2e-42,
```


COUNT & COUNT(*)



VIEW



LIKE CLAUSE ON CUSTOMER



```

C:\Users\user> java HelloWorld.java
Hello World!
Hello Java!

```

JOINS

[illegible]

INNER JOIN

[illegible]

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 391–397

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 369–375

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 2007 | 2006 | 2005 | 2004 | 2003 | 2002 | 2001 | 2000 | 1999 | 1998 | 1997 | 1996 | 1995 | 1994 | 1993 | 1992 | 1991 | 1990 | 1989 | 1988 | 1987 | 1986 | 1985 | 1984 | 1983 | 1982 | 1981 | 1980 | 1979 | 1978 | 1977 | 1976 | 1975 | 1974 | 1973 | 1972 | 1971 | 1970 | 1969 | 1968 | 1967 | 1966 | 1965 | 1964 | 1963 | 1962 | 1961 | 1960 | 1959 | 1958 | 1957 | 1956 | 1955 | 1954 | 1953 | 1952 | 1951 | 1950 | 1949 | 1948 | 1947 | 1946 | 1945 | 1944 | 1943 | 1942 | 1941 | 1940 | 1939 | 1938 | 1937 | 1936 | 1935 | 1934 | 1933 | 1932 | 1931 | 1930 | 1929 | 1928 | 1927 | 1926 | 1925 | 1924 | 1923 | 1922 | 1921 | 1920 | 1919 | 1918 | 1917 | 1916 | 1915 | 1914 | 1913 | 1912 | 1911 | 1910 | 1909 | 1908 | 1907 | 1906 | 1905 | 1904 | 1903 | 1902 | 1901 | 1900 | 1899 | 1898 | 1897 | 1896 | 1895 | 1894 | 1893 | 1892 | 1891 | 1890 | 1889 | 1888 | 1887 | 1886 | 1885 | 1884 | 1883 | 1882 | 1881 | 1880 | 1879 | 1878 | 1877 | 1876 | 1875 | 1874 | 1873 | 1872 | 1871 | 1870 | 1869 | 1868 | 1867 | 1866 | 1865 | 1864 | 1863 | 1862 | 1861 | 1860 | 1859 | 1858 | 1857 | 1856 | 1855 | 1854 | 1853 | 1852 | 1851 | 1850 | 1849 | 1848 | 1847 | 1846 | 1845 | 1844 | 1843 | 1842 | 1841 | 1840 | 1839 | 1838 | 1837 | 1836 | 1835 | 1834 | 1833 | 1832 | 1831 | 1830 | 1829 | 1828 | 1827 | 1826 | 1825 | 1824 | 1823 | 1822 | 1821 | 1820 | 1819 | 1818 | 1817 | 1816 | 1815 | 1814 | 1813 | 1812 | 1811 | 1810 | 1809 | 1808 | 1807 | 1806 | 1805 | 1804 | 1803 | 1802 | 1801 | 1800 | 1799 | 1798 | 1797 | 1796 | 1795 | 1794 | 1793 | 1792 | 1791 | 1790 | 1789 | 1788 | 1787 | 1786 | 1785 | 1784 | 1783 | 1782 | 1781 | 1780 | 1779 | 1778 | 1777 | 1776 | 1775 | 1774 | 1773 | 1772 | 1771 | 1770 | 1769 | 1768 | 1767 | 1766 | 1765 | 1764 | 1763 | 1762 | 1761 | 1760 | 1759 | 1758 | 1757 | 1756 | 1755 | 1754 | 1753 | 1752 | 1751 | 1750 | 1749 | 1748 | 1747 | 1746 | 1745 | 1744 | 1743 | 1742 | 1741 | 1740 | 1739 | 1738 | 1737 | 1736 | 1735 | 1734 | 1733 | 1732 | 1731 | 1730 | 1729 | 1728 | 1727 | 1726 | 1725 | 1724 | 1723 | 1722 | 1721 | 1720 | 1719 | 1718 | 1717 | 1716 | 1715 | 1714 | 1713 | 1712 | 1711 | 1710 | 1709 | 1708 | 1707 | 1706 | 1705 | 1704 | 1703 | 1702 | 1701 | 1700 | 1699 | 1698 | 1697 | 1696 | 1695 | 1694 | 1693 | 1692 | 1691 | 1690 | 1689 | 1688 | 1687 | 1686 | 1685 | 1684 | 1683 | 1682 | 1681 | 1680 | 1679 | 1678 | 1677 | 1676 | 1675 | 1674 | 1673 | 1672 | 1671 | 1670 | 1669 | 1668 | 1667 | 1666 | 1665 | 1664 | 1663 | 1662 | 1661 | 1660 | 1659 | 1658 | 1657 | 1656 | 1655 | 1654 | 1653 | 1652 | 1651 | 1650 | 1649 | 1648 | 1647 | 1646 | 1645 | 1644 | 1643 | 1642 | 1641 | 1640 | 1639 | 1638 | 1637 | 1636 | 1635 | 1634 | 1633 | 1632 | 1631 | 1630 | 1629 | 1628 | 1627 | 1626 | 1625 | 1624 | 1623 | 1622 | 1621 | 1620 | 1619 | 1618 | 1617 | 1616 | 1615 | 1614 | 1613 | 1612 | 1611 | 1610 | 1609 | 1608 | 1607 | 1606 | 1605 | 1604 | 1603 | 1602 | 1601 | 1600 | 1599 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|

www.elsevier.com/locate/jmb

© 2006 by American Psychological Association or one of its allied publishers. This article is intended solely for the personal use of the individual user and is not to be disseminated broadly.

| | |
|---|---|
| 1 | 1 |
| 2 | 2 |

CHEN 2007

1998

```

# This function will take an IP address and return the name that corresponds to your IP.
# It will return the name of the host that corresponds to the IP address.
def get_hostname(ip):
    """Returns the hostname of the IP address"""
    return socket.gethostbyaddr(ip)

# This function will take a domain name and return the IP address that corresponds to it.
# It will return the IP address of the host that corresponds to the domain name.
def get_ip(hostname):
    """Returns the IP address of the hostname"""
    return socket.getaddrinfo(hostname, None)[0][4][0]

# This function will take a domain name and return the IP address that corresponds to it.
# It will return the IP address of the host that corresponds to the domain name.
def get_ip_by_hostname(hostname):
    """Returns the IP address of the hostname"""
    return get_ip(hostname)

# This function will take a domain name and return the IP address that corresponds to it.
# It will return the IP address of the host that corresponds to the domain name.
def get_ip_by_hostname(hostname):
    """Returns the IP address of the hostname"""
    return get_ip(hostname)

```

Received 12 July 2006; accepted 12 March 2007

| Case | Location | Year | Age | Sex | Weight |
|------|----------|----------|----------|----------|----------|
| 1 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 |
| 2 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 |
| 3 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 |
| 4 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 | 10/10/00 |

1. *Journal of Management Studies*, 1990, 27, 1, 1-10.

https://doi.org/10.1016/j.jm.2019.05.001

| | |
|-----|--------|
| LOW | 2,500+ |
| 25% | 16,667 |
| 50% | 8,333 |
| 75% | 5,556 |

© 1999 Blackwell Science Ltd, *Journal of Internal Medicine* 245: 395–402

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

EQUI JOIN

[illegible]

LEFT JOIN

[illegible]

RIGHT JOIN

```

root@kali:~/Documents# docker exec -it kali_1 bash
kali_1:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
kubepki:x:1000:1000:kube:/kubernetes:/bin/bash
root@kali:~/Documents# docker exec -it kali_1 bash
kali_1:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
kubepki:x:1000:1000:kube:/kubernetes:/bin/bash
root@kali:~/Documents#

```

CROSS JOIN

```

# Create the data frame
df = pd.DataFrame({
    'Year': [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030],
    'Category': ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U'],
    'Value': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000]
})

# Print the data frame
print(df)

```

The screenshot shows a Windows 10 desktop environment. A terminal window is open, displaying a directory listing of files and folders. The terminal title is "C:\Users\user>". The listing is organized into columns, likely representing file type, name, and size. The files include various system files, user files, and application files. The terminal is running on a Windows 10 desktop with a taskbar at the bottom showing the Start button, search bar, and several application icons.

DELETE ROW4 FROM STAFF

```
SQL> DELETE FROM STAFF WHERE S_ID=4;
SQL> SELECT * FROM STAFF;
SQL> DELETE FROM STAFF WHERE S_ID=5;
SQL> SELECT * FROM STAFF;
```

| S_ID | S_NAME | S_ADDRESS | S_CITY | S_ZIP | S_SALARY |
|------|--------|-----------|-------------|-------|----------|
| 1 | James | 12345 | New York | 10001 | 12000 |
| 2 | John | 67890 | Los Angeles | 90001 | 8000 |
| 3 | David | 11111 | Chicago | 60601 | 15000 |
| 6 | James | 12345 | New York | 10001 | 12000 |

| S_ID | S_NAME | S_ADDRESS | S_CITY | S_ZIP | S_SALARY |
|------|--------|-----------|-------------|-------|----------|
| 1 | James | 12345 | New York | 10001 | 12000 |
| 2 | John | 67890 | Los Angeles | 90001 | 8000 |
| 3 | David | 11111 | Chicago | 60601 | 15000 |

DROP

```
SQL> DROP TABLE STAFF;
SQL> SELECT * FROM STAFF;
```

TRUNCATE

```
SQL> TRUNCATE TABLE STAFF;
SQL> SELECT * FROM STAFF;
```

| S_ID | S_NAME | S_ADDRESS | S_CITY | S_ZIP | S_SALARY |
|------|--------|-----------|--------|-------|----------|
|------|--------|-----------|--------|-------|----------|

GRANT

```
mysql> create user identified by '1234'
ERROR 1045 (42002): User 'root' as 'root' is not the daemon; check the account that corresponds to your mysql socket socket for the right
privilege to use user 'root' on '127.0.0.1' at 1794.1
mysql> create user user identified by '1234'
Query OK, 0 rows affected (0.00 sec)

mysql> create role example
ERROR 1105 (HY000): operation cannot now be performed for 'example'@'%'
mysql> create role example
ERROR 1105 (HY000): operation cannot now be performed for 'example'@'%'
mysql> create user example user
Query OK, 0 rows affected (0.00 sec)

mysql> grant manager role to user
Query OK, 0 rows affected (0.00 sec)

mysql> show grants;
+-----+
| Grants for user@localhost |
+-----+
|
+-----+

mysql>
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, EXECUTE, GRANT OPTION,
CREATE TEMPORARY TABLES, LOCK TABLES, CREATE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE USER, DROP USER, GRANT OPTION, ALTER
USER, CREATE VIEW, TRIGGER, CREATE TABLESPACE, CREATE ROUTE, SUPER, BACKUP, > TO user@localhost WITH GRANT OPTION;
```

```
mysql>
mysql> GRANT FOR example@localhost
+-----+
| Grants for example@localhost |
+-----+
|
+-----+

mysql>
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, EXECUTE, GRANT OPTION,
CREATE TEMPORARY TABLES, LOCK TABLES, CREATE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE USER, DROP USER, GRANT OPTION, ALTER
USER, CREATE VIEW, TRIGGER, CREATE TABLESPACE, CREATE ROUTE, SUPER, BACKUP, > TO user@localhost WITH GRANT OPTION;
```

PRIVILEGES

[illegible][illegible]

[illegible]