Appointment System

The assignment involves designing a database schema for an appointment system, implementing it with sample data, and writing SQL queries to retrieve specific information from the database.

STEP-1

Firstly, we created the database as an appointment system then created the 4 tables in the database as users,doctor,clinics and appointments.

```
mysql> CREATE DATABASE appointment_system;
Query OK, 1 row affected (0.00 sec)

mysql> USE appointment_system;
Database changed
mysql> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     birthdate DATE NOT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE doctors (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE clinics (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE clinics (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL
    -> );
ERROR 1050 (42S01): Table 'clinics' already exists
mysql> CREATE TABLE appointments (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     user_id INT NOT NULL,
    ->     doctor_id INT NOT NULL,
    ->     clinic_id INT NOT NULL,
    ->     appointment_time DATETIME NOT NULL,
    ->     status ENUM('booked', 'cancelled') NOT NULL,
    ->     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ->     FOREIGN KEY (user_id) REFERENCES users(id),
    ->     FOREIGN KEY (doctor_id) REFERENCES doctors(id),
    ->     FOREIGN KEY (clinic_id) REFERENCES clinics(id)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

In the above image, I created the database and tables in the database.

STEP-2

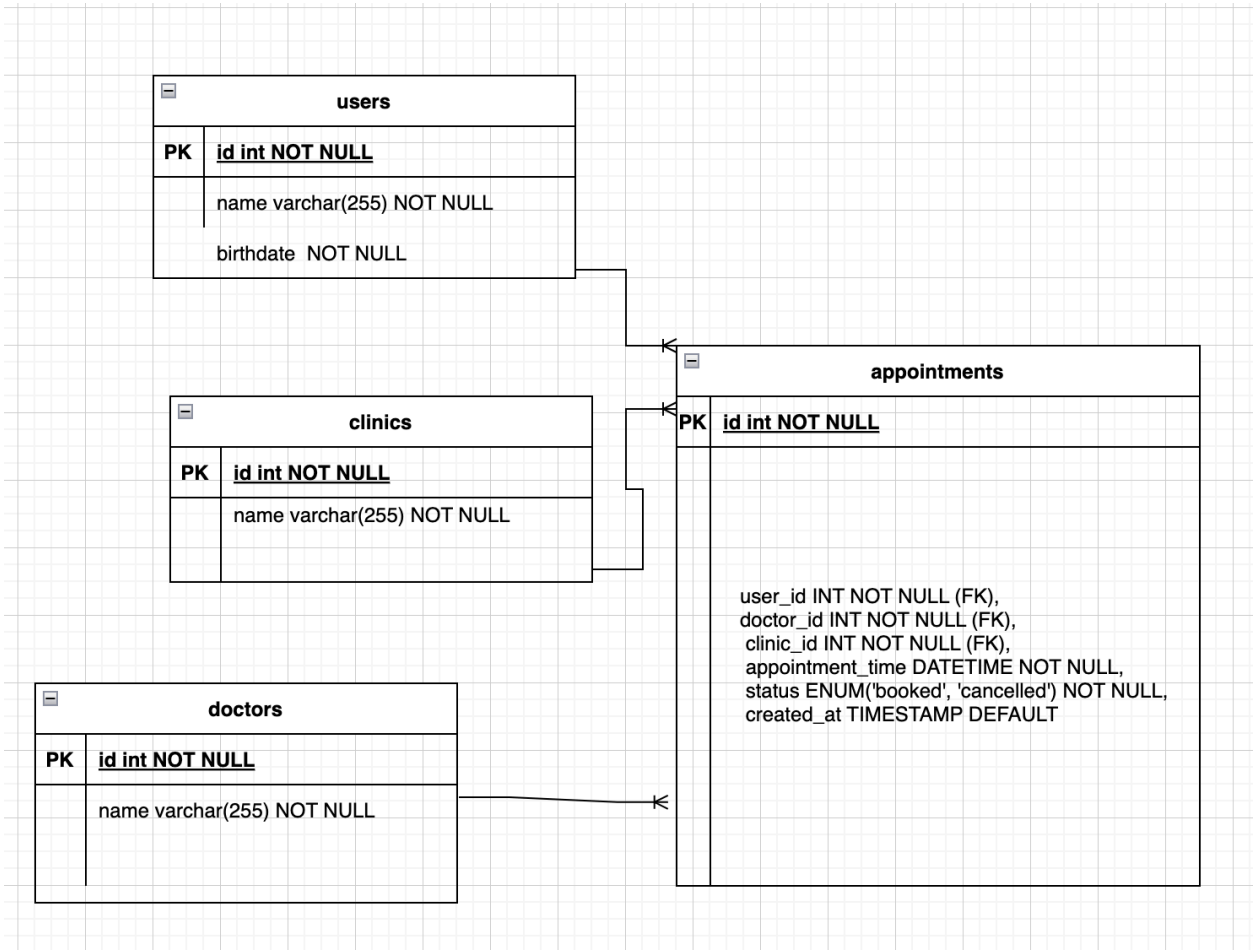
Inserted the values in the tables.

```
mysql> INSERT INTO users (name, birthdate)
    -> VALUES
    -> ('Jane Doe', '1990-06-28'),
    -> ('Bob Smith', '1985-06-29'),
    -> ('Samuel Johnson', '1978-06-30'),
    -> ('Linda Brown', '1995-07-01'),
    -> ('George Wilson', '1980-07-02'),
    -> ('Anna Lee', '1992-07-03');
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> INSERT INTO doctors (name)
    -> VALUES
    -> ('Dr. Alice'),
    -> ('Dr. Bob'),
    -> ('Dr. Charlie'),
    -> ('Dr. David'),
    -> ('Dr. Eve');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO clinics (name)
    -> VALUES
    -> ('Clinic A'),
    -> ('Clinic B'),
    -> ('Clinic C'),
    -> ('Clinic D'),
    -> ('Clinic E');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO appointments (user_id, doctor_id, clinic_id, appointment_time, status)
    -> VALUES
    -> (1, 1, 1, '2024-06-28 09:00:00', 'booked'),
    -> (2, 2, 2, '2024-06-29 10:30:00', 'booked'),
    -> (3, 3, 3, '2024-06-30 14:00:00', 'booked'),
    -> (4, 4, 4, '2024-07-01 11:00:00', 'booked'),
    -> (5, 5, 5, '2024-07-02 15:30:00', 'booked'),
    -> (6, 1, 1, '2024-07-03 16:00:00', 'booked'),
    -> (2, 3, 2, '2024-06-26 09:00:00', 'cancelled'),
    -> (3, 2, 3, '2024-06-27 10:30:00', 'booked'),
    -> (4, 1, 4, '2024-06-28 14:00:00', 'cancelled'),
    -> (5, 3, 5, '2024-06-29 11:00:00', 'cancelled'),
    -> (6, 2, 1, '2024-06-30 15:30:00', 'booked'),
    -> (1, 4, 2, '2024-07-01 16:00:00', 'booked'),
    -> (2, 5, 3, '2024-07-02 08:30:00', 'cancelled'),
    -> (3, 4, 4, '2024-07-03 09:00:00', 'booked'),
    -> (4, 5, 5, '2024-07-04 10:30:00', 'booked');
Query OK, 15 rows affected (0.01 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

Relationships:

- Many-to-One relationship between **users** and **appointments** (via user_id).
- Many-to-One relationship between **doctors** and **appointments** (via doctor_id).
- Many-to-One relationship between **clinics** and **appointments** (via clinic_id).

ER-Diagram



STEP-3

Query-1:All appointments booked in the last 7 days for a doctor

```
mysql> SELECT *
    -> FROM appointments
    -> WHERE doctor_id = 1
    ->   AND created_at >= NOW() - INTERVAL 7 DAY;
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
| id | user_id | doctor_id | clinic_id | appointment_time    | status    | created_at          |
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
|  1 |       1 |         1 |         1 | 2024-06-28 09:00:00 | booked    | 2024-06-28 19:28:37 |
|  6 |       6 |         1 |         1 | 2024-07-03 16:00:00 | booked    | 2024-06-28 19:28:37 |
|  9 |       4 |         1 |         4 | 2024-06-28 14:00:00 | cancelled | 2024-06-28 19:28:37 |
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
3 rows in set (0.00 sec)
```

EXPLAIN Image:

```
mysql> Explain SELECT *
    -> FROM appointments
    -> WHERE doctor_id = 1
    ->   AND created_at >= CURDATE() - INTERVAL 7 DAY;
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
| id | select_type | table        | partitions | type | possible_keys | key       | key_len | ref   | rows | filtered | Extra       |
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
|  1 | SIMPLE      | appointments | NULL       | ref  | doctor_id     | doctor_id | 4       | const |    4 |    33.33 | Using where |
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
1 row in set, 1 warning (0.00 sec)
```

Query-2:All appointments booked in the last 2 days and scheduled within the next 5 hours for a doctor

```
mysql> SELECT *
    -> FROM appointments
    -> WHERE doctor_id = 1
    ->   AND created_at >= NOW() - INTERVAL 2 DAY
    ->   AND appointment_time <= NOW() + INTERVAL 5 HOUR;
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
| id | user_id | doctor_id | clinic_id | appointment_time    | status    | created_at          |
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
|  1 |       1 |         1 |         1 | 2024-06-28 09:00:00 | booked    | 2024-06-28 19:28:37 |
|  9 |       4 |         1 |         4 | 2024-06-28 14:00:00 | cancelled | 2024-06-28 19:28:37 |
+----+---------+-----------+-----------+---------------------+-----------+---------------------+
2 rows in set (0.00 sec)
```

EXPLAIN Image:

```
mysql> EXPLAIN SELECT *
    -> FROM appointments
    -> WHERE doctor_id = 1
    ->   AND created_at >= CURDATE() - INTERVAL 2 DAY
    ->   AND appointment_time <= NOW() + INTERVAL 5 HOUR;
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
| id | select_type | table        | partitions | type | possible_keys | key       | key_len | ref   | rows | filtered | Extra       |
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
|  1 | SIMPLE      | appointments | NULL       | ref  | doctor_id     | doctor_id | 4       | const |    4 |    11.11 | Using where |
+----+-------------+--------------+------------+------+---------------+-----------+---------+-------+------+----------+-------------+
1 row in set, 1 warning (0.00 sec)
```

Query-3:Users who have at least 1 appointment and have their birthday coming in the next 5 days

```
mysql> SELECT DISTINCT u.*
    -> FROM users u
    -> JOIN appointments a ON u.id = a.user_id
    -> WHERE DATE_FORMAT(u.birthdate, '%m-%d') BETWEEN DATE_FORMAT(CURDATE(), '%m-%d') AND DATE_FORMAT(CURDATE() + INTERVAL 5 DAY, '%m-%d');
+----+----------------+------------+
| id | name           | birthdate  |
+----+----------------+------------+
|  1 | Jane Doe       | 1990-06-28 |
|  2 | Bob Smith      | 1985-06-29 |
|  3 | Samuel Johnson | 1978-06-30 |
|  4 | Linda Brown    | 1995-07-01 |
|  5 | George Wilson  | 1980-07-02 |
|  6 | Anna Lee       | 1992-07-03 |
|  7 | Yogi           | 2024-07-01 |
+----+----------------+------------+
7 rows in set (0.00 sec)
```

EXPLAIN Image:

```
mysql> EXPLAIN SELECT DISTINCT u.*
    -> FROM users u
    -> JOIN appointments a ON u.id = a.user_id
    -> WHERE DATE_FORMAT(u.birthdate, '%m-%d') BETWEEN DATE_FORMAT(CURDATE(), '%m-%d') AND DATE_FORMAT(CURDATE() + INTERVAL 5 DAY, '%m-%d');
+----+-------------+-------+------------+------+---------------+---------+---------+-------------------------+------+----------+-----------------------------+
| id | select_type | table | partitions | type | possible_keys | key     | key_len | ref                     | rows | filtered | Extra                       |
+----+-------------+-------+------------+------+---------------+---------+---------+-------------------------+------+----------+-----------------------------+
|  1 | SIMPLE      | u     | NULL       | ALL  | PRIMARY       | NULL    | NULL    | NULL                    |    7 |   100.00 | Using where; Using temporary |
|  1 | SIMPLE      | a     | NULL       | ref  | user_id       | user_id | 4       | appointment_system.u.id |    1 |   100.00 | Using index; Distinct       |
+----+-------------+-------+------------+------+---------------+---------+---------+-------------------------+------+----------+-----------------------------+
2 rows in set, 1 warning (0.00 sec)
```

Query-4:Appointments for a particular patient in the last 7 days

```
mysql> SELECT *
    -> FROM appointments
    -> WHERE user_id = 1
    ->   AND created_at >= NOW() - INTERVAL 7 DAY;
+----+---------+-----------+-----------+---------------------+---------+---------------------+
| id | user_id | doctor_id | clinic_id | appointment_time    | status  | created_at          |
+----+---------+-----------+-----------+---------------------+---------+---------------------+
|  1 |       1 |         1 |         1 | 2024-06-28 09:00:00 | booked  | 2024-06-28 19:28:37 |
| 12 |       1 |         4 |         2 | 2024-07-01 16:00:00 | booked  | 2024-06-28 19:28:37 |
+----+---------+-----------+-----------+---------------------+---------+---------------------+
2 rows in set (0.00 sec)
```

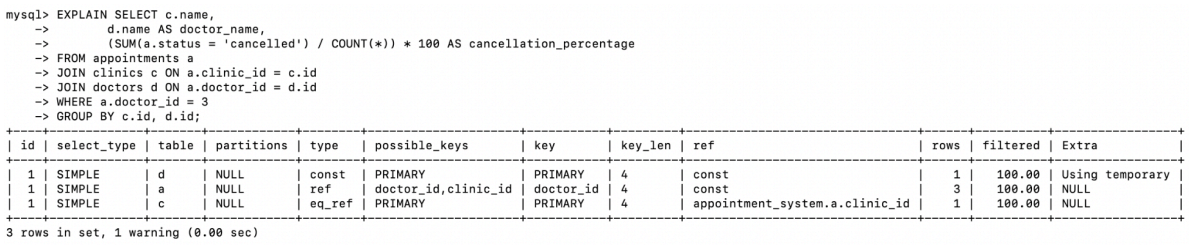
EXPLAIN Image:

```
mysql> EXPLAIN SELECT *
    -> FROM appointments
    -> WHERE user_id = 1
    ->   AND created_at >= CURDATE() - INTERVAL 7 DAY;
+----+-------------+--------------+------------+------+---------------+---------+---------+-------+------+----------+-------------+
| id | select_type | table        | partitions | type | possible_keys | key     | key_len | ref   | rows | filtered | Extra       |
+----+-------------+--------------+------------+------+---------------+---------+---------+-------+------+----------+-------------+
|  1 | SIMPLE      | appointments | NULL       | ref  | user_id       | user_id | 4       | const |    2 |    33.33 | Using where |
+----+-------------+--------------+------------+------+---------------+---------+---------+-------+------+----------+-------------+
1 row in set, 1 warning (0.00 sec)
```

Query-5:Appointment cancellation percentage for a doctor by clinic

```
mysql> SELECT c.name AS clinic_name,
    ->        d.name AS doctor_name,
    ->        (SUM(a.status = 'cancelled') / COUNT(*)) * 100 AS cancellation_percentage
    -> FROM appointments a
    -> JOIN clinics c ON a.clinic_id = c.id
    -> JOIN doctors d ON a.doctor_id = d.id
    -> WHERE a.doctor_id = 3
    -> GROUP BY c.id, d.id;
+-------------+-------------+--------------------------+
| clinic_name | doctor_name | cancellation_percentage |
+-------------+-------------+--------------------------+
| Clinic C    | Dr. Charlie |                   0.0000 |
| Clinic B    | Dr. Charlie |                 100.0000 |
| Clinic E    | Dr. Charlie |                 100.0000 |
+-------------+-------------+--------------------------+
3 rows in set (0.00 sec)
```

EXPLAIN Image:



```
mysql> EXPLAIN SELECT c.name,
    ->        d.name AS doctor_name,
    ->        (SUM(a.status = 'cancelled') / COUNT(*)) * 100 AS cancellation_percentage
    -> FROM appointments a
    -> JOIN clinics c ON a.clinic_id = c.id
    -> JOIN doctors d ON a.doctor_id = d.id
    -> WHERE a.doctor_id = 3
    -> GROUP BY c.id, d.id;
+----+-------------+-------+------------+--------+--------------------+-----------+---------+--------------------------------+------+----------+-----------------+
| id | select_type | table | partitions | type   | possible_keys      | key       | key_len | ref                            | rows | filtered | Extra           |
+----+-------------+-------+------------+--------+--------------------+-----------+---------+--------------------------------+------+----------+-----------------+
|  1 | SIMPLE      | d     | NULL       | const  | PRIMARY            | PRIMARY   | 4       | const                          |    1 |   100.00 | Using temporary |
|  1 | SIMPLE      | a     | NULL       | ref    | doctor_id,clinic_id | doctor_id | 4       | const                          |    3 |   100.00 | NULL            |
|  1 | SIMPLE      | c     | NULL       | eq_ref | PRIMARY            | PRIMARY   | 4       | appointment_system.a.clinic_id |    1 |   100.00 | NULL            |
+----+-------------+-------+------------+--------+--------------------+-----------+---------+--------------------------------+------+----------+-----------------+
3 rows in set, 1 warning (0.00 sec)
```