Database Schema

Tables

1. **Users**
    - `user_id` (Primary Key)
    - `name`
    - `email`
    - `birthdate`
    - `phone_number`
2. **Clinics**
    - `clinic_id` (Primary Key)
    - `name`
    - `address`
    - `phone_number`
3. **Doctors**
    - `doctor_id` (Primary Key)
    - `name`
    - `specialty`
4. **Doctor_Clinics**
    - `doctor_clinic_id` (Primary Key)
    - `doctor_id` (Foreign Key to Doctors)
    - `clinic_id` (Foreign Key to Clinics)
5. **Appointments**
    - `appointment_id` (Primary Key)
    - `user_id` (Foreign Key to Users)
    - `doctor_clinic_id` (Foreign Key to Doctor_Clinics)
    - `appointment_time`
    - `booking_time`
    - `status`

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.37-0ubuntu0.23.10.2 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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

```
mysql>
mysql> -- Users table
mysql> CREATE TABLE Users_New (
    ->     user_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     email VARCHAR(255) UNIQUE NOT NULL,
    ->     birthdate DATE NOT NULL,
    ->     phone_number VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> -- Clinics table
mysql> CREATE TABLE Clinics_New (
    ->     clinic_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     address VARCHAR(255) NOT NULL,
    ->     phone_number VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> -- Doctors table
mysql> CREATE TABLE Doctors_New (
    ->     doctor_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     specialty VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

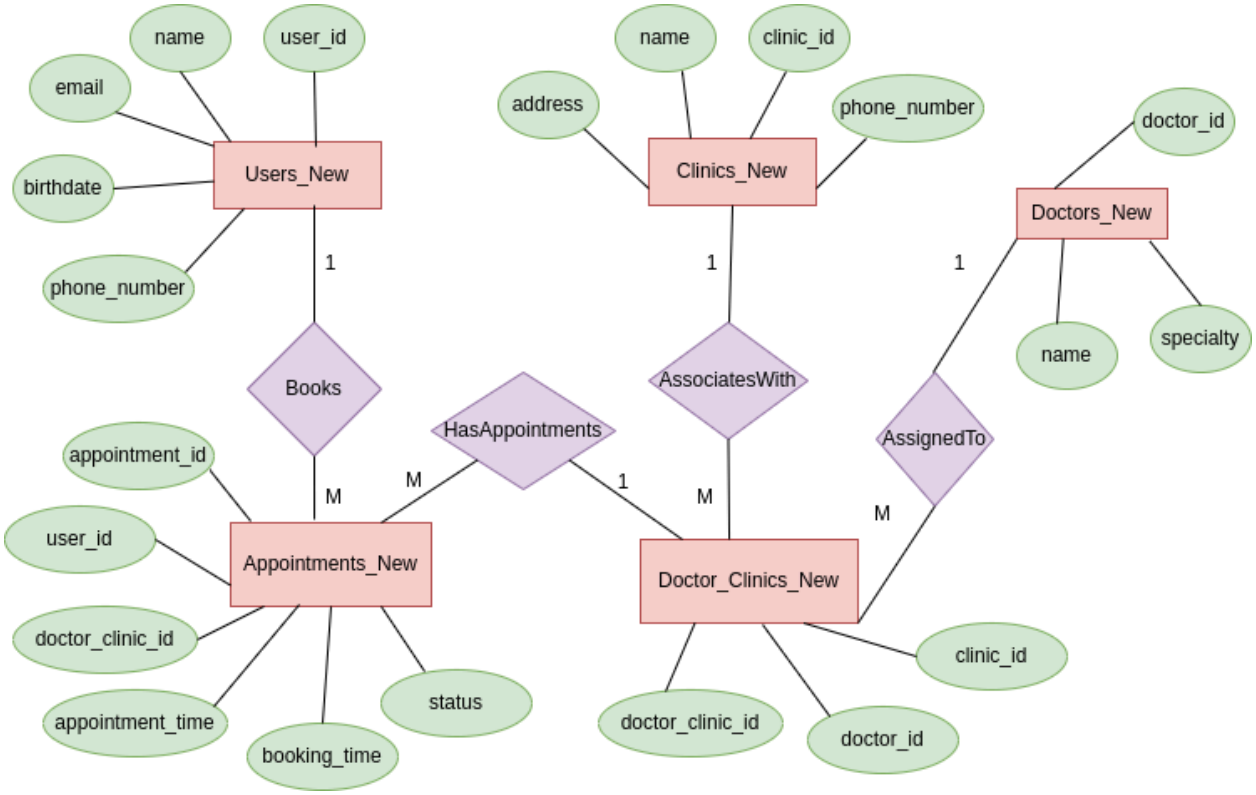
```
mysql>
mysql> -- Doctor_Clinics table
mysql> CREATE TABLE Doctor_Clinics_New (
    ->     doctor_clinic_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     doctor_id INT NOT NULL,
    ->     clinic_id INT NOT NULL,
    ->     FOREIGN KEY (doctor_id) REFERENCES Doctors_New(doctor_id),
    ->     FOREIGN KEY (clinic_id) REFERENCES Clinics_New(clinic_id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> -- Appointments table
mysql> CREATE TABLE Appointments_New (
    ->     appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     user_id INT NOT NULL,
    ->     doctor_clinic_id INT NOT NULL,
    ->     appointment_time DATETIME NOT NULL,
    ->     booking_time DATETIME NOT NULL,
    ->     status VARCHAR(20) NOT NULL,
    ->     FOREIGN KEY (user_id) REFERENCES Users_New(user_id),
    ->     FOREIGN KEY (doctor_clinic_id) REFERENCES Doctor_Clinics_New(doctor_clinic_id)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql>
mysql> -- Indexes for optimization
mysql> CREATE INDEX idx_booking_time_new ON Appointments_New (booking_time);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_appointment_time_new ON Appointments_New (appointment_time);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_doctor_id_new ON Doctor_Clinics_New (doctor_id);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_birthdate_new ON Users_New (birthdate);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_user_id_new ON Appointments_New (user_id);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql>
mysql> -- Insert values into Users_New table
mysql> INSERT INTO Users_New (name, email, birthdate, phone_number) VALUES
    -> ('John Doe', 'john_new@example.com', '1985-07-01', '1234567890'),
    -> ('Jane Smith', 'jane_new@example.com', '1990-07-01', '0987654321'),
    -> ('Alice Johnson', 'alice_new@example.com', '1995-07-01', '1122334455'),
    -> ('Bob Brown', 'bob_new@example.com', '1988-07-01', '2233445566');
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert values into Clinics_New table
mysql> INSERT INTO Clinics_New (name, address, phone_number) VALUES
    -> ('City Clinic', '123 Main St', '1112223333'),
    -> ('Downtown Clinic', '456 Elm St', '2223334444'),
    -> ('Uptown Clinic', '789 Oak St', '3334445555');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert values into Doctors_New table
mysql> INSERT INTO Doctors_New (name, specialty) VALUES
    -> ('Dr. Smith', 'Cardiology'),
    -> ('Dr. Johnson', 'Dermatology'),
    -> ('Dr. Brown', 'Pediatrics');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql>
mysql> -- Insert values into Doctor_Clinics_New table
mysql> INSERT INTO Doctor_Clinics_New (doctor_id, clinic_id) VALUES
    -> (1, 1),
    -> (1, 2),
    -> (2, 2),
    -> (3, 3);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert values into Appointments_New table
mysql> INSERT INTO Appointments_New (user_id, doctor_clinic_id, appointment_time, booking_time, status) VALUES
    -> (1, 1, '2024-06-26 10:00:00', '2024-06-20 09:00:00', 'booked'),
    -> (1, 2, '2024-06-26 11:00:00', '2024-06-20 10:00:00', 'cancelled'),
    -> (2, 2, '2024-06-27 12:00:00', '2024-06-21 11:00:00', 'completed'),
    -> (3, 3, '2024-06-28 13:00:00', '2024-06-22 12:00:00', 'booked'),
    -> (4, 1, '2024-06-29 14:00:00', '2024-06-23 13:00:00', 'booked'),
    -> (1, 2, '2024-06-30 15:00:00', '2024-06-24 14:00:00', 'completed'),
    -> (2, 3, '2024-07-01 16:00:00', '2024-06-25 15:00:00', 'booked');
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

ER DIAGRAM



QUERIES

1. All appointments booked in last 7 days for a doctor

SELECT a.appointment_id, a.user_id, a.doctor_clinic_id, a.appointment_time, a.booking_time, a.status
FROM Appointments_New a
JOIN Doctor_Clinics_New dc ON a.doctor_clinic_id = dc.doctor_clinic_id
WHERE dc.doctor_id = 1 AND a.booking_time >= NOW() - INTERVAL 7 DAY;

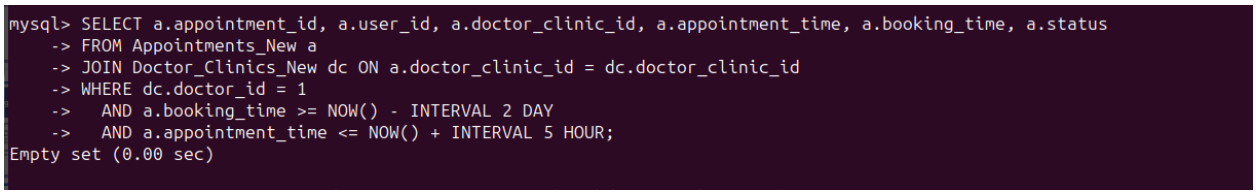
```
mysql> SELECT a.appointment_id, a.user_id, a.doctor_clinic_id, a.appointment_time, a.booking_time, a.status
    -> FROM Appointments_New a
    -> JOIN Doctor_Clinics_New dc ON a.doctor_clinic_id = dc.doctor_clinic_id
    -> WHERE dc.doctor_id = 1 AND a.booking_time >= NOW() - INTERVAL 7 DAY;
+----------------+---------+-----------------+---------------------+---------------------+-----------+
| appointment_id | user_id | doctor_clinic_id | appointment_time    | booking_time        | status    |
+----------------+---------+-----------------+---------------------+---------------------+-----------+
|              5 |       4 |               1 | 2024-06-29 14:00:00 | 2024-06-23 13:00:00 | booked    |
|              6 |       1 |               2 | 2024-06-30 15:00:00 | 2024-06-24 14:00:00 | completed |
+----------------+---------+-----------------+---------------------+---------------------+-----------+
2 rows in set (0.00 sec)
```

Explanation:

This query retrieves the appointment details for doctor ID 1 where the booking time was within the last 7 days. It joins the `Appointments_New` table with the `Doctor_Clinics_New` table to get the clinic details associated with the doctor.

2. All appointments booked in last 2 days n scheduled within next 5 hours for a doctor

```
SELECT a.appointment_id, a.user_id, a.doctor_clinic_id, a.appointment_time,
a.booking_time, a.status
FROM Appointments_New a
JOIN Doctor_Clinics_New dc ON a.doctor_clinic_id = dc.doctor_clinic_id
WHERE dc.doctor_id = 1
  AND a.booking_time >= NOW() - INTERVAL 2 DAY
  AND a.appointment_time <= NOW() + INTERVAL 5 HOUR;
```



Explanation:

This query fetches appointments for doctor ID 1 where the booking time was within the last 2 days and the appointment time is within the next 5 hours.

3. User who have atleast 1 appointment and have their birthday coming in next 5 days

```
SELECT DISTINCT u.user_id, u.name, u.email, u.birthdate, u.phone_number
FROM Users_New u
JOIN Appointments_New a ON u.user_id = a.user_id
WHERE u.birthdate BETWEEN CURDATE() AND CURDATE() + INTERVAL 5 DAY;
```

```
mysql> SELECT DISTINCT u.user_id, u.name, u.email, u.birthdate, u.phone_number
    -> FROM Users_New u
    -> JOIN Appointments_New a ON u.user_id = a.user_id
    -> WHERE u.birthdate BETWEEN CURDATE() AND CURDATE() + INTERVAL 5 DAY;
Empty set (0.00 sec)
```

Explanation:

This query lists users who have an appointment and whose birthday falls within the next 5 days.

4. Appointments for a particular patient in the last 7 days

SELECT a.appointment_id, a.doctor_clinic_id, a.appointment_time, a.booking_time, a.status
FROM Appointments_New a
WHERE a.user_id = 1 AND a.appointment_time >= NOW() - INTERVAL 7 DAY;

```
mysql> SELECT a.appointment_id, a.doctor_clinic_id, a.appointment_time, a.booking_time, a.status
    -> FROM Appointments_New a
    -> WHERE a.user_id = 1 AND a.appointment_time >= NOW() - INTERVAL 7 DAY;
+----------------+-----------------+---------------------+---------------------+-----------+
| appointment_id | doctor_clinic_id | appointment_time    | booking_time        | status    |
+----------------+-----------------+---------------------+---------------------+-----------+
|              1 |               1 | 2024-06-26 10:00:00 | 2024-06-20 09:00:00 | booked    |
|              2 |               2 | 2024-06-26 11:00:00 | 2024-06-20 10:00:00 | cancelled |
|              6 |               2 | 2024-06-30 15:00:00 | 2024-06-24 14:00:00 | completed |
+----------------+-----------------+---------------------+---------------------+-----------+
3 rows in set (0.00 sec)
```

Explanation:

This query retrieves the appointment details for user ID 1 where the appointment time was within the last 7 days.

5. Appointment cancellation percentage for a doctor by clinic

SELECT dc.clinic_id, c.name AS clinic_name,
COUNT(a.appointment_id) AS total_appointments,
SUM(CASE WHEN a.status = 'cancelled' THEN 1 ELSE 0 END) AS cancelled_appointments,(SUM(CASE WHEN a.status = 'cancelled' THEN 1 ELSE 0 END) / COUNT(a.appointment_id)) * 100 AS cancellation_percentage

FROM Appointments_New a

JOIN Doctor_Clinics_New dc ON a.doctor_clinic_id = dc.doctor_clinic_id

JOIN Clinics_New c ON dc.clinic_id = c.clinic_id

WHERE dc.doctor_id = 1

GROUP BY dc.clinic_id, c.name;

```
mysql> SELECT dc.clinic_id, c.name AS clinic_name,
    ->        COUNT(a.appointment_id) AS total_appointments,
    ->        SUM(CASE WHEN a.status = 'cancelled' THEN 1 ELSE 0 END) AS cancelled_appointments,
    ->        (SUM(CASE WHEN a.status = 'cancelled' THEN 1 ELSE 0 END) / COUNT(a.appointment_id)) * 100 AS cancellation_percenta
ge
    -> FROM Appointments_New a
    -> JOIN Doctor_Clinics_New dc ON a.doctor_clinic_id = dc.doctor_clinic_id
    -> JOIN Clinics_New c ON dc.clinic_id = c.clinic_id
    -> WHERE dc.doctor_id = 1
    -> GROUP BY dc.clinic_id, c.name;
+-----------+----------------+--------------------+----------------------+-------------------------+
| clinic_id | clinic_name    | total_appointments | cancelled_appointments | cancellation_percentage |
+-----------+----------------+--------------------+----------------------+-------------------------+
|         1 | City Clinic    |                  2 |                    0 |                  0.0000 |
|         2 | Downtown Clinic |                 3 |                    1 |                 33.3333 |
+-----------+----------------+--------------------+----------------------+-------------------------+
2 rows in set (0.00 sec)
```

Explanation:

This query calculates the total number of appointments and the percentage of canceled appointments for doctor ID 1, grouped by clinic.