CDAC Mumbai
PG-DAC Aug24 Batch (Juhu& Kharghar)
Lab Exam
Module 4: Database Technology

Time: 1:30hrs **Marks: 40**

Instructions:

- All queries must be written using MySQL.
- Ensure to use correct syntax and test your queries before submission.
- Marks are distributed as per the sub-questions.
- Mention the table structures, relationships, and logic in your queries.

Section A: (10 Marks)

1. (4 Marks)
   Create at least three tables (Patients, Doctors, Appointments) and insert a minimum of five records into each table. Define the column specifications appropriately.

Patients Table:

- PatientID (INT, PRIMARY KEY, AUTO_INCREMENT)
- FirstName (VARCHAR(50), NOT NULL)
- LastName (VARCHAR(50), NOT NULL)
- DOB (DATE, NOT NULL)

Doctors Table:

- DoctorID (INT, PRIMARY KEY, AUTO_INCREMENT)
- DoctorName (VARCHAR(100), NOT NULL)
- Specialization (VARCHAR(100))

Appointments Table:

- AppointmentID (INT, PRIMARY KEY, AUTO_INCREMENT)
- PatientID (INT, FOREIGN KEY REFERENCES Patients(PatientID))
- DoctorID (INT, FOREIGN KEY REFERENCES Doctors(DoctorID))
- AppointmentDate (DATE, NOT NULL)

2. (2 Marks)
   Write SQL queries to create foreign key relationships between Patients, Doctors, and Appointments tables.
3. (2 Marks)
   Write a query to fetch all patients who have an appointment scheduled after '2023-07-01' or with a cardiologist.
4. (2 Marks)
   Write a query to fetch the first 5 patients from the Patients table.

Section B: Intermediate SQL Queries (15 Marks)

5. (2 Marks)
   Write a query to display the number of appointments per doctor.

6. (2 Marks)
   Write a query to display the DoctorName and Specialization of the highest-paid doctor.
7. (3 Marks)
   Write a query to create a view named PatientAppointments that shows PatientID, DoctorID, and AppointmentDate. Update the appointment date for a patient through this view.
8. (2 Marks)
   Write a query to perform a LEFT JOIN between Patients and Appointments tables to retrieve all patients, including those who don't have any appointments.
9. (2 Marks)
   Create a temporary table with records of patients who are older than 60 years. Drop the temporary table after use.
10. (4 Marks)
    Write a stored procedure named GetDoctorDetails that takes a DoctorID as input and returns the DoctorName and total number of appointments for that doctor.

Section C: (15 Marks)

11. (4 Marks)
    Write a MySQL program using a WHILE loop to calculate appointment fees for all appointments scheduled after a specific date.
12. (4 Marks)
    Write a stored function named BILL_CALC that takes the total cost and discount percentage as input and returns the final medical bill.
13. (2 Marks)
    Write a query to find doctors who have more than 10 appointments using the HAVING clause.
14. (2 Marks)
    Write a trigger that logs changes to appointments whenever the appointment date is updated.
15. (3 Marks)
    Write a subquery to fetch the AppointmentDate and DoctorName of the second most recent appointment.