Course Code	**:**	**BCSL-034**
Title	**:**	**DBMS Lab**
Assignment Number	**:**	**BCA(III)/L-034/Assignment/2026-27**
Maximum Marks	**:**	**50**
Weightage	**:**	**25%**
Last Date of Submission	**:**	**31st October, 2026 (for July 2026 Session)**
	:	**30thApril, 2027 (for January 2027 Session)**

This assignment has only one question. Answer the question. This question carries 40 marks. Rest 10 marks are for viva voce. You may use illustrations and diagrams to enhance the explanation. Assumptions can be made wherever required. Please go through the guidelines regarding the assignments given in the programme guide for the format of presentation.

Q1: **(15 Marks)**

Design and implement a database system using any relational DBMS for an *Online Hospital Appointment Management System*, demonstrating fundamental DBMS concepts such as database design, entity relationships, CRUD operations and SQL queries for data retrieval and analysis.

Create Database Schema

Create the following tables.

Patients Table

- patient_id (Primary Key, INT, Auto Increment)
- patient_name (VARCHAR)
- gender (VARCHAR)
- date_of_birth (DATE)
- phone_number (VARCHAR)
- city (VARCHAR)

Doctors Table

- doctor_id (Primary Key, INT, Auto Increment)
- doctor_name (VARCHAR)
- specialization (VARCHAR)
- consultation_fee (DECIMAL)
- experience (INT)

Appointments Table

- appointment_id (Primary Key, INT, Auto Increment)
- patient_id (Foreign Key)
- doctor_id (Foreign Key)
- appointment_date (DATE)

- appointment_time (TIME)
- status (VARCHAR)

Relationships

- A patient can book multiple appointments.
- A doctor can attend multiple patients.
- Each appointment is associated with exactly one patient and one doctor.

Draw an ER Diagram **(5 Marks)**

Draw a neat ER diagram showing:

- Entities
- Attributes
- Primary Keys
- Foreign Keys
- Relationships with appropriate cardinality

Perform the Following CRUD Operations **(5 Marks)**

- Insert sample records into all tables.
- Display records from all tables.
- Update patient contact information.
- Update appointment status.
- Delete a cancelled appointment.
- Delete a doctor record after ensuring referential integrity.

SQL Queries **(15 Marks)**

Write and execute the following SQL queries.

1. Display all appointments along with patient and doctor details.
2. Retrieve all appointments scheduled for a particular date.
3. List doctors belonging to a specific specialization.
4. Display patients who have visited more than one doctor.
5. Find the doctor having the highest consultation fee.
6. Display the total number of appointments handled by each doctor.
7. Retrieve all cancelled appointments.
8. Find patients who have not booked any appointment.
9. Display appointments ordered by appointment date and time.
10. Find the total consultation fees generated by each doctor.
11. Display the average consultation fee for each specialization.
12. Retrieve the doctor(s) having more than five appointments.

13. Display the city-wise count of registered patients.
14. Find the oldest patient registered in the database.
15. Display specialization-wise number of doctors available.

Documentation Checklist:

Include the following in your submission:

- Database schema
- Table creation scripts
- ER Diagram
- Sample data inserted into all tables
- SQL commands
- Output of all SQL queries
- Screenshots of database tables
- Screenshots of query execution
- Brief explanation of each SQL query
- Summary of observations and learning outcomes

Q1: (15 Marks)

Design and implement a database system using any relational DBMS for an Online Hospital Appointment Management System, demonstrating fundamental DBMS concepts such as database design, entity relationships, CRUD operations and SQL queries for data retrieval and analysis.

Create Database Schema

Create the following tables.

Patients Table

- **patient_id (Primary Key, INT, Auto Increment)**
- **patient_name (VARCHAR)**
- **gender (VARCHAR)**
- **date_of_birth (DATE)**
- **phone_number (VARCHAR)**
- **city (VARCHAR)**

Doctors Table

- **doctor_id (Primary Key, INT, Auto Increment)**
- **doctor_name (VARCHAR)**
- **specialization (VARCHAR)**
- **consultation_fee (DECIMAL)**
- **experience (INT)**

Appointments Table

- **appointment_id (Primary Key, INT, Auto Increment)**
- **patient_id (Foreign Key)**
- **doctor_id (Foreign Key)**
- **appointment_date (DATE)**
- **appointment_time (TIME)**
- **status (VARCHAR)**