CREATE TABLE Students (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

birthdate DATE,

gender CHAR(1),

city VARCHAR(50)

);

CREATE TABLE Courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

credits INT

);

CREATE TABLE Instructors (

instructor\_id INT PRIMARY KEY,

full\_name VARCHAR(100),

hire\_date DATE

);

CREATE TABLE Enrollments (

enrollment\_id INT PRIMARY KEY,

student\_id INT,

course\_id INT,

enrollment\_date DATE,

grade DECIMAL(4,2),

FOREIGN KEY (student\_id) REFERENCES Students(student\_id),

FOREIGN KEY (course\_id) REFERENCES Courses(course\_id)

);

INSERT INTO Students VALUES

(1, 'Alice', 'Roy', '2000-05-12', 'F', 'Delhi'),

(2, 'Bob', 'Khan', '1999-11-23', 'M', 'Mumbai'),

(3, 'Charlie', 'Patel', '2001-03-30', 'M', 'Bangalore'),

(4, 'Diana', 'Singh', '2002-07-08', 'F', 'Mumbai'),

(5, 'Evan', 'Mishra', '2003-01-14', 'M', 'Delhi'),

(6, 'Fiona', 'Das', '2000-12-20', 'F', 'Kolkata'),

(7, 'George', 'Rao', '2002-02-25', 'M', 'Hyderabad'),

(8, 'Hannah', 'Gupta', '2001-06-11', 'F', 'Mumbai'),

(9, 'Ian', 'Verma', '1998-09-17', 'M', 'Delhi'),

(10, 'Jaya', 'Joshi', '2004-04-22', 'F', 'Chennai');

INSERT INTO Courses VALUES

(101, 'Database Systems', 4),

(102, 'Operating Systems', 3),

(103, 'Computer Networks', 4),

(104, 'Software Engineering', 3),

(105, 'Data Structures', 4);

INSERT INTO Instructors VALUES

(1, 'Dr. Anil Mehta', '2015-06-01'),

(2, 'Dr. Neha Jain', '2018-01-15'),

(3, 'Dr. Rakesh Sharma', '2017-09-20');

INSERT INTO Enrollments VALUES

(1, 1, 101, '2022-01-10', 85.5),

(2, 2, 101, '2022-01-10', 78.0),

(3, 3, 102, '2022-01-15', 68.2),

(4, 1, 103, '2022-01-17', 92.0),

(5, 4, 104, '2022-02-10', 73.5),

(6, 5, 105, '2022-02-15', 59.0),

(7, 6, 101, '2022-02-18', 60.2),

(8, 7, 103, '2022-03-01', 88.0),

(9, 8, 102, '2022-03-05', 91.3),

(10, 9, 101, '2022-03-10', 76.0),

(11, 10, 105, '2022-03-20', 45.5),

(12, 2, 105, '2022-04-01', 80.0),

(13, 3, 105, '2022-04-05', 77.5),

(14, 4, 101, '2022-04-10', 83.2),

(15, 5, 103, '2022-04-15', 71.0),

(16, 6, 104, '2022-04-20', 66.0),

(17, 7, 105, '2022-04-25', 74.5),

(18, 8, 101, '2022-05-01', 88.0),

(19, 9, 102, '2022-05-05', 81.0),

(20, 10, 103, '2022-05-10', 69.5);

1. Why is it important to use a data model when designing a system like UniversityDB? Describe two benefits.

2. In our schema, identify and explain the entities, their attributes, and the relationships between them.

3. Classify each attribute in the Students table as single‑valued, multi‑valued, or derived (and suggest a derived attribute).

4. What kinds of entities (weak, strong, associative) are present in this schema? Justify your answer.

5. List two advantages of using an ER diagram before implementing SQL tables.

6. What is SQL, and why is it essential for relational databases? Provide a brief history.

7. Explain differences between RDBMS and NoSQL. Give one example of each.

8. Define “database,” “table,” and “row” in context of this schema.

9. What SQL client tools would you install to connect to the database and run queries?

10. Give examples of data types used in this schema and explain appropriate use.

11. Write a query to list all Female students (gender = 'F') born after January 1, 2001, whose city is not 'Mumbai'. Use AND, OR, and NOT.

12. Select students whose first\_name contains exactly 5 letters and ends with 'a', using wildcard operators % and \_ correctly.

13. Find students born between March 1, 2000 and February 28, 2002 inclusive.

14. Compute the number of characters in each student’s full name (first + last) using CONCAT, LENGTH, and alias the result as name\_length.

15. Show full names and find position of substring “an” in their first\_name using INSTR().

16. Calculate each student’s age in years (using birthdate and CURDATE() or CURRENT\_TIMESTAMP()), rounded down.

17. For each enrollment, display student\_id, course\_id, and the day of the week when they enrolled. Use appropriate date functions.

18. Find average grade per course\_id, but only include courses with more than 3 enrollments.

19. List cities having at least 2 students, along with the maximum grade among enrollments for each city.

20. Determine how many students have enrolled in all courses offered. Use grouping and filtering logic.