1. Convert temperature from Celsius to Fahrenheit.
2. Calculate simple interest given principal, rate, and time.
3. Calculate volume and surface area of a cube.
4. Convert kilometers to miles.
5. Convert string to uppercase and lowercase.
6. Calculate total price including GST.
7. Find the average of three numbers.
8. Convert hours into minutes and seconds.
9. Extract year, month, and day from a given date.
10. Calculate power of a number using INOUT (base^exponent).
11. Convert amount in rupees to dollars.
12. Join two strings with a space between them.
13. Calculate discounted price of an item.
14. Extract domain name from an email address.
15. Find minimum, maximum, and average of two numbers.
16. Compute the perimeter of a triangle.
17. Reverse a string using OUT parameter.
18. Format a name as “Last, First” using INOUT.
19. Get the current timestamp in an OUT parameter.
20. Return the first 5 characters of a string.

These are **problems** based on the schemas given towards the end of this file.

1. Calculate the **average marks** of a student across all enrolled courses.
2. Enroll a student into a course only if the student belongs to the same department as the course.
3. Generate a **report card** for a student showing course-wise marks and grades.
4. Promote a student to the next academic year if their average marks are above a threshold.
5. Send a warning to students (log in a table) who failed more than 2 subjects.
6. Automatically assign a **grade** (A, B, C, D, F) based on marks.
7. Get a list of **top 3 students** department-wise based on average marks.
8. Calculate **pass percentage** of a course across all students.
9. Find students **eligible for scholarships** (average marks > 85 and no backlogs).
10. Show all courses a student has **not enrolled in** yet.
11. Detect and prevent duplicate enrollments using procedure logic.
12. Assign a **mentor** to students based on their department and year.
13. Create a backlog tracker — list courses where a student scored below passing marks.
14. Calculate the **total number of students** in each course.
15. Generate a **consolidated result** sheet for all students in a specific course.
16. Show list of **students failing multiple years** for academic review.
17. Calculate **department-wise average marks** per semester.
18. Auto-drop students who haven't cleared minimum required credits in 2 years.
19. Insert and notify **marks entry errors** if marks > 100 or < 0.
20. Insert attendance data and flag students with attendance < 75%.

CREATE TABLE students (

student\_id INT PRIMARY KEY,

name VARCHAR(100),

department VARCHAR(50),

admission\_year INT

);

INSERT INTO students VALUES

(1, 'Aditi Sharma', 'Computer Science', 2022),

(2, 'Ravi Kumar', 'Electronics', 2021),

(3, 'Sneha Das', 'Mathematics', 2022);

CREATE TABLE courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

department VARCHAR(50)

);

INSERT INTO courses VALUES

(101, 'Data Structures', 'Computer Science'),

(102, 'Digital Electronics', 'Electronics'),

(103, 'Calculus', 'Mathematics');

CREATE TABLE enrollments (

student\_id INT,

course\_id INT,

PRIMARY KEY (student\_id, course\_id),

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

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

);

INSERT INTO enrollments VALUES

(1, 101),

(2, 102),

(3, 103),

(1, 103); -- Aditi is also taking Calculus

CREATE TABLE marks (

student\_id INT,

course\_id INT,

marks\_obtained INT,

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

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

);

INSERT INTO marks VALUES

(1, 101, 88),

(1, 103, 70),

(2, 102, 45),

(3, 103, 90);

CREATE TABLE grades (

student\_id INT,

course\_id INT,

grade VARCHAR(25),

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

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

);

CREATE TABLE attendance (

id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT,

date DATE,

status ENUM('Present', 'Absent'),

FOREIGN KEY (student\_id) REFERENCES students(student\_id)

);

INSERT INTO attendance (student\_id, date, status) VALUES

(1, '2025-04-01', 'Present'),

(1, '2025-04-02', 'Absent'),

(1, '2025-04-03', 'Present'),

(1, '2025-04-04', 'Present'),

(2, '2025-04-01', 'Absent'),

(3, '2025-04-01', 'Present');