SQL Machine Test Paper

Batch : June-24/July-24 **Date : 26/06/2025**

Time : 3 Hrs.

instructors

Field	Type	Null	Key	Default	Extra
instructor_id	INT	NO	PRI	NULL	
instructor_name	VARCHAR(100)	YES		NULL	
salary	DECIMAL(10,2)	YES		NULL	

courses

Field	Type	Null	Key	Default	Extra
course_id	INT	NO	PRI	NULL	
course_name	VARCHAR(100)	YES		NULL	
instructor_id	INT	YES	MUL	NULL	

students

Field	Type	Null	Key	Default	Extra
student_id	INT	NO	PRI	NULL	
name	VARCHAR(100)	YES		NULL	
age	INT	YES		NULL	
course_id	INT	YES	MUL	NULL	

marks

Field	Type	Null	Key	Default	Extra
mark_id	INT	NO	PRI	NULL	
student_id	INT	YES	MUL	NULL	

subject	VARCHAR(50)	YES		NULL	
mark	INT	YES		NULL	

1. Write a query to display student names with their respective course names.
2. Retrieve the names of all instructors who teach a course taken by students older than 22.
3. Display student name, subject, and mark using appropriate JOIN(s).
4. List all students along with their instructor's name (if any). Include students even if they don’t have a course.
5. Find the average marks for each student.
6. Show courses with no students enrolled.
7. Find the names of students who scored more than the average mark in any subject.
8. Display students who are enrolled in the same course as student "Rahul".
9. Find the instructor who has the highest number of students enrolled under their courses.
10. List the names of students who have scored full marks (assume full mark is 100) in any subject.
11. Create a view student_summary showing student_id, name, course_name, and total_marks.
12. Write a stored procedure to display all subjects and marks for a given student_id.
13. Write a stored function to calculate grade based on mark (e.g., A for 90+, B for 80+, etc.).
14. Create a stored procedure that returns the number of students enrolled in a specific course.
15. Write a function to return the instructor name for a given student_id.
16. Create a view top_performers showing students who scored more than 90 in any subject.
17. Create a trigger that logs an entry into a marks_log table whenever a new mark is inserted.
18. Create a BEFORE INSERT trigger on the students table to ensure age is not less than 18.
19. Write a trigger to update a last_updated timestamp column in marks table whenever a row is updated.
20. Create a trigger to prevent deleting an instructor if they are assigned to any course.

Best Of Luck