

GIRI's Tech Hub, Pune
MySQL (Machine) Test

Batch: Nov-23 to May-24 & Aug/Sep/Oct-24

Date: 23/06/2025

Time: 08:00 to 11:00 Am

Instructions:

Total:- 10 Marks

1. Solve Any 18 Queries.
2. Input should be from user.
3. Each Query have 0.50 Marks .

Student Table					
Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
age	int	YES		NULL	
department_id	int	YES		NULL	

Department Table					
Field	Type	Null	Key	Default	Extra
department_id	int	NO	PRI	NULL	
department_name	varchar(100)	YES		NULL	

Course Table					
Field	Type	Null	Key	Default	Extra
course_id	int	NO	PRI	NULL	
course_name	varchar(100)	YES		NULL	
department_id	int	YES		NULL	

Enrollment Table					
Field	Type	Null	Key	Default	Extra
enrollment_id	int	NO	PRI	NULL	
student_id	int	YES		NULL	
course_id	int	YES		NULL	
marks	int	YES		NULL	

1. List courses having an average score of more than 60.
2. List all students along with courses, including those who haven't enrolled.
3. List all students who are enrolled in the course with the highest number of enrollments.
4. Display courses not taken by any student.
5. Find Top Scorer in Each Course.
6. Find Departments with All Students Enrolled in At Least One Course.
7. Find Courses With Students From Multiple Departments.
8. Find Students Who Are Toppers in Their Department.

9. Find Department with Highest Average Score.
10. Find Students Who Never Failed (Assume <40 is fail).
11. Find List Of Students with Third(3rd) Highest Total Marks
12. Course-wise Department Performance Summary
14. Write a Procedure List Students Who Took All Courses of Their Department
15. Write a procedure that takes course_id and returns the average marks for it.
16. Create Views as Course-wise Highest and Lowest Marks.
17. Create Views as Students with Highest Marks in Their Enrolled Courses.
18. Create Views as Detailed Student Course Report (with department and marks).
19. Create a trigger that logs every insert operation on the Enrollments table into a separate table called Enrollment_Audit. This log should include student ID, course ID, marks, and a timestamp.
20. Design a trigger that stores the data of deleted students into a backup table named Deleted_Students, whenever a record is deleted from the Students table.
21. Write a trigger that recalculates and updates the total marks in the Student_Total_Marks table whenever a student's marks in any course are updated.

-----ALL THE BEST-----