**My SQL assignment – Basic**

**-------------------------------------------------------------------------------------------------------------**

**1.Create a database named college\_db.**

**Ans:** CREATE DATABASE college\_db;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2.Create a table students with fields: id, name, age, department.**

**Ans:** USE college\_db;

CREATE TABLE students (

id INT PRIMARY KEY,

name VARCHAR(50),

age INT,

department VARCHAR(50)

);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Insert 5 records into the students table.**

**Ans:** INSERT INTO students VALUES

(1, 'Alice', 22, 'Computer Science'),

(2, 'Bob', 19, 'Mechanical'),

(3, 'John', 21, 'Electrical'),

(4, 'Aria', 24, 'Computer Science'),

(5, 'Mike', 18, 'Civil');

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Write a query to fetch all records from students.**

Ans: SELECT \* FROM students;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Fetch students whose age is greater than 20.**

Ans: SELECT \* FROM students WHERE age > 20;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Update the department of a student where name is ‘John’.**

**Ans:** UPDATE students SET department = 'Electronics' WHERE name = 'John';

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7. Delete a student whose ID is 3.**

Ans: DELETE FROM students WHERE id = 3;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**8. Select students ordered by age in descending order.**

Ans: SELECT \* FROM students ORDER BY age DESC;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**9. Fetch only distinct departments from the students table.**

**Ans:** SELECT DISTINCT department FROM students;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**10. Count the number of students in the table.**

Ans: SELECT COUNT(\*) AS total\_students FROM students;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**11. Rename the students table to student\_info.**

Ans: RENAME TABLE students TO student\_info;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**12. Add a new column email to the student\_info table.**

Ans: ALTER TABLE student\_info ADD email VARCHAR(100);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**13. Write a query to find students whose name starts with ‘A’.**

**Ans:** SELECT \* FROM student\_info WHERE name LIKE 'A%';

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**14. Display students whose age is between 18 and 25.**

**Ans:** SELECT \* FROM student\_info WHERE age BETWEEN 18 AND 25;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**15. Write a query to find the student with the highest age.**

Ans: SELECT \* FROM student\_info ORDER BY age DESC LIMIT 1;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**16. Use LIMIT to display the first 3 students.**

Ans: SELECT \* FROM student\_info LIMIT 3;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**17. Create a table courses with fields: course\_id, course\_name, credits.**

Ans: CREATE TABLE courses (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

credits INT

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**18. Insert 3 records into the courses table.**

Ans: INSERT INTO courses VALUES

(1, 'Data Structures', 4),

(2, 'Thermodynamics', 3),

(3, 'Fluid Mechanics', 4);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**19. Select all students whose department is ‘Computer Science’.**

Ans: SELECT \* FROM student\_info WHERE department = 'Computer Science';

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**20. Use IN to fetch students from specific departments.**

**Ans:** SELECT \* FROM student\_info WHERE department IN ('Computer Science', 'Mechanical');

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**21. Use BETWEEN to find students aged between 20 and 30.**

Ans: SELECT \* FROM student\_info WHERE age BETWEEN 20 AND 30;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**22. Write a query to display current system date and time.**

Ans: SELECT NOW() AS current\_datetime;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**23. Use AS to rename a column in the SELECT query.**

Ans: SELECT name AS student\_name, age FROM student\_info;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**24. Write a query to fetch all data except students of a particular department.**

Ans: SELECT \* FROM student\_info WHERE department != 'Civil';

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**25. Delete all records from the students table without dropping the table.**

Ans: DELETE FROM student\_info;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**-------------------------------------------------------------------------------------------------------------**

**My SQL assignment – Advance**

**-----------------------------------------------------------------------------------------------**

**26.Create a marks table with fields: student\_id, subject, marks.**

Ans: CREATE TABLE marks (

student\_id INT,

subject VARCHAR(50),

marks INT

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**27. Insert at least 5 records into the marks table.**

Ans: INSERT INTO marks VALUES

(1, 'Math', 85),

(2, 'Math', 90),

(1, 'Physics', 78),

(4, 'Math', 92),

(5, 'Physics', 66);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**28. Use JOIN to combine students and marks data.**

Ans: INSERT INTO marks VALUES

(1, 'Math', 85),

(2, 'Math', 90),

(1, 'Physics', 78),

(4, 'Math', 92),

(5, 'Physics', 66);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**29. Write a query to calculate average marks per student.**

Ans: SELECT student\_id, AVG(marks) AS average\_marks

FROM marks GROUP BY student\_id;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**30. Use GROUP BY to find total marks obtained by each student.**

Ans: SELECT student\_id, SUM(marks) AS total\_marks

FROM marks GROUP BY student\_id;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**31. Use HAVING to find students who scored more than 200 in total.**

Ans: SELECT student\_id, SUM(marks) AS total

FROM marks

GROUP BY student\_id

HAVING total > 200;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**32. Write a query to fetch students with the same age using GROUP BY and COUNT().**

Ans: SELECT age, COUNT(\*) AS same\_age\_count

FROM student\_info GROUP BY age

HAVING same\_age\_count > 1;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**33. Use INNER JOIN, LEFT JOIN, RIGHT JOIN and explain the difference.**

Ans:

-- INNER JOIN SELECT \* FROM student\_info s INNER JOIN marks m ON s.id = m.student\_id;

-- LEFT JOIN SELECT \* FROM student\_info s LEFT JOIN marks m ON s.id = m.student\_id;

-- RIGHT JOIN SELECT \* FROM student\_info s RIGHT JOIN marks m ON s.id = m.student\_id;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**34. Create a new table with a PRIMARY KEY and AUTO\_INCREMENT.**

Ans: CREATE TABLE departments (

dept\_id INT PRIMARY KEY AUTO\_INCREMENT,

dept\_name VARCHAR(50)

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**35. Create a table with a FOREIGN KEY referencing another table.**

Ans: CREATE TABLE staff (

staff\_id INT PRIMARY KEY,

staff\_name VARCHAR(50),

dept\_id INT,

FOREIGN KEY (dept\_id)

REFERENCES departments(dept\_id)

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**36. Write a subquery to find the maximum marks in the marks table.**

Ans: SELECT MAX(marks) AS max\_marks FROM marks;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**37. Create a view to display student names and their total marks.**

Ans: CREATE VIEW student\_total\_marks AS

SELECT s.name, SUM(m.marks) AS total\_marks

FROM student\_info s

JOIN marks m ON s.id = m.student\_id

GROUP BY s.name;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**38. Use a subquery to list students who scored more than the average mark.**

Ans: SELECT name FROM student\_info

WHERE id IN (

SELECT student\_id FROM marks

GROUP BY student\_id

HAVING AVG(marks) > (SELECT AVG(marks) FROM marks)

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**39. Create a stored procedure to insert new student data.**

Ans: CREATE PROCEDURE insert\_student(IN sid INT, IN sname VARCHAR(50), IN sage INT, IN dept VARCHAR(50))

BEGIN

INSERT INTO student\_info (id, name, age, department) VALUES (sid, sname, sage, dept);

END;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**40. Create a stored procedure to update student department.**

Ans: CREATE PROCEDURE update\_department(IN sname VARCHAR(50), IN new\_dept VARCHAR(50))

BEGIN

UPDATE student\_info SET department = new\_dept WHERE name = sname;

END;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**41. Create a user-defined function to calculate grade from marks.**

Ans: CREATE FUNCTION calculate\_grade(marks INT) RETURNS VARCHAR(2)

BEGIN

DECLARE grade VARCHAR(2);

IF marks >= 90 THEN

SET grade = 'A+';

ELSEIF marks >= 75 THEN

SET grade = 'A';

ELSEIF marks >= 60 THEN

SET grade = 'B';

ELSE

SET grade = 'C';

END IF;

RETURN grade;

END;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**42. Create a trigger that logs insert operations on students.**

Ans: CREATE TABLE logs (

log\_id INT AUTO\_INCREMENT PRIMARY KEY,

message TEXT,

log\_time DATETIME DEFAULT CURRENT\_TIMESTAMP );

CREATE TRIGGER student\_insert\_log

AFTER INSERT ON student\_info

FOR EACH ROW

BEGIN

INSERT INTO logs(message) VALUES (CONCAT('Inserted student: ', NEW.name));

END;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**43. Use a transaction to update multiple records atomically.**

Ans: START TRANSACTION;

UPDATE marks SET marks = marks + 5 WHERE subject = 'Math';

UPDATE student\_info SET age = age + 1 WHERE department = 'Mechanical';

COMMIT;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**44. Write a query to find duplicate records using GROUP BY and HAVING.**

Ans: SELECT name, COUNT(\*) FROM student\_info

GROUP BY name

HAVING COUNT(\*) > 1;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**45. Create a backup of a database using mysqldump.**

Ans: mysqldump -u root -p college\_db > backup.sql

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**46. Restore a MySQL database from a backup file.**

Ans: mysql -u root -p college\_db < backup.sql

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**47. Import data from a CSV file into a MySQL table.**

Ans: LOAD DATA INFILE '/path/to/your/file.csv'

INTO TABLE student\_info

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**48. Create an index on student name for faster search.**

Ans: CREATE INDEX idx\_name ON student\_info(name);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**49. Write a query to find the second highest mark in a subject.**

Ans: SELECT MAX(marks) AS second\_highest

FROM marks

WHERE marks < (SELECT MAX(marks) FROM marks);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**50. Drop the courses table and explain the effect.**

Ans: DROP TABLE courses;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**----------------------------------------------------------------------------------------------------------------**