MySQL Assignment

Q1) Create a database named college\_db.

Ans.1) CREATE DATABASE college\_db;  
  
USE college\_db;

Q2) Create a table students with fields: id, name, age, department.

Ans.2) CREATE TABLE students (id INT, name VARCHAR(30), age INT, department VARCHAR(30));

Q3) Insert 5 records into the students table.

Ans.3) INSERT INTO students VALUES  
(1, 'Alice', 19, 'Computer Science'),  
(2, 'Bob', 22, 'Electronics'),  
(3, 'John', 21, 'Mechanical'),  
(4, 'Eva', 20, 'Computer Science'),  
(5, 'Mike', 23, 'Civil');

Q4) Write a query to fetch all records from students.

Ans.4) SELECT \* FROM students;

Q5) Fetch students whose age is greater than 20.

Ans.5) SELECT \* FROM students WHERE age > 20;

Q6) Update the department of a student where name is ‘John’.

Ans.6) UPDATE students SET department = 'Electrical' WHERE name = 'John';

Q7) Delete a student whose ID is 3.

Ans.7) DELETE FROM students WHERE id = 3;

Q8) Select students ordered by age in descending order.

Ans.8) SELECT \* FROM students ORDER BY age DESC;

Q9) Fetch only distinct departments from the students table.

Ans.9) SELECT DISTINCT department FROM students;

Q10)

Ans.10) SELECT COUNT(\*) FROM students;

Q11)

Ans.11) RENAME TABLE students TO student\_info;

Q12)

Ans.12) ALTER TABLE student\_info ADD email VARCHAR(50);

Q13)

Ans.13) SELECT \* FROM student\_info WHERE name LIKE 'A%';

Q14)

Ans.14) SELECT \* FROM student\_info WHERE age BETWEEN 18 AND 25;

Q15)

Ans.15) SELECT \* FROM student\_info ORDER BY age DESC LIMIT 1;

Q16)

Ans.16) SELECT \* FROM student\_info LIMIT 3;

Q17)

Ans.17) CREATE TABLE courses (  
 course\_id INT,  
 course\_name VARCHAR(50),  
 credits INT  
);

Q18)

Ans.18) INSERT INTO courses VALUES  
(101, 'DBMS', 4),  
(102, 'Operating Systems', 3),  
(103, 'Networks', 3);

Q19)

Ans.19) SELECT \* FROM student\_info WHERE department = 'Computer Science';

Q20)

Ans.20) SELECT \* FROM student\_info WHERE department IN ('Computer Science', 'Electronics');

Q21)

Ans.21) SELECT \* FROM student\_info WHERE age BETWEEN 20 AND 30;

Q22)

Ans.22) SELECT NOW();

Q23)

Ans.23) SELECT name AS student\_name FROM student\_info;

Q24)

Ans.24) SELECT \* FROM student\_info WHERE department <> 'Mechanical';

Q25)

Ans.25) DELETE FROM student\_info;

Q26)

Ans.26) CREATE TABLE marks (  
 student\_id INT,  
 subject VARCHAR(30),  
 marks INT  
);

Q27)

Ans.27) INSERT INTO marks VALUES  
(1, 'DBMS', 85),  
(1, 'OS', 78),  
(2, 'DBMS', 90),  
(2, 'OS', 88),  
(4, 'DBMS', 75);

Q28)

Ans.28) SELECT s.id, s.name, m.subject, m.marks  
FROM student\_info s  
JOIN marks m ON s.id = m.student\_id;

Q29)

Ans.29) SELECT student\_id, AVG(marks) AS avg\_marks  
FROM marks  
GROUP BY student\_id;

Q30)

Ans.30) SELECT student\_id, SUM(marks) AS total\_marks  
FROM marks  
GROUP BY student\_id;

Q31)

Ans.31) SELECT student\_id, SUM(marks) AS total  
FROM marks  
GROUP BY student\_id  
HAVING total > 200;

Q32)

Ans.32) SELECT age, COUNT(\*) AS total\_students  
FROM student\_info  
GROUP BY age  
HAVING total\_students > 1;

Q33)

Ans.33) -- 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;

Q34)

Ans.34) CREATE TABLE faculty (  
 faculty\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(30)  
);

Q35)

Ans.35) CREATE TABLE enrollments (  
 student\_id INT,  
 course\_id INT,  
 FOREIGN KEY (student\_id) REFERENCES student\_info(id),  
 FOREIGN KEY (course\_id) REFERENCES courses(course\_id)  
);

Q36)

Ans.36) SELECT \* FROM marks  
WHERE marks = (SELECT MAX(marks) FROM marks);

Q37)

Ans.37) CREATE VIEW student\_totals 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;

Q38)

Ans.38) SELECT \* FROM marks  
WHERE marks > (SELECT AVG(marks) FROM marks);

Q39)

Ans.39) DELIMITER //  
CREATE PROCEDURE insert\_student(IN sid INT, IN sname VARCHAR(30), IN sage INT, IN dept VARCHAR(30))  
BEGIN  
 INSERT INTO student\_info (id, name, age, department) VALUES (sid, sname, sage, dept);  
END;//  
DELIMITER ;

Q40)

Ans.40) DELIMITER //  
CREATE PROCEDURE update\_dept(IN sid INT, IN new\_dept VARCHAR(30))  
BEGIN  
 UPDATE student\_info SET department = new\_dept WHERE id = sid;  
END;//  
DELIMITER ;

Q41)

Ans.41) DELIMITER //  
CREATE FUNCTION get\_grade(marks INT) RETURNS VARCHAR(2)  
BEGIN  
 RETURN CASE  
 WHEN marks >= 90 THEN 'A'  
 WHEN marks >= 75 THEN 'B'  
 WHEN marks >= 60 THEN 'C'  
 ELSE 'D'  
 END;  
END;//  
DELIMITER ;

Q42)

Ans.42) CREATE TABLE student\_log (  
 log\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 student\_name VARCHAR(30),  
 inserted\_on DATETIME  
);  
  
DELIMITER //  
CREATE TRIGGER log\_student\_insert  
AFTER INSERT ON student\_info  
FOR EACH ROW  
BEGIN  
 INSERT INTO student\_log (student\_name, inserted\_on) VALUES (NEW.name, NOW());  
END;//  
DELIMITER ;

Q43)

Ans.43) START TRANSACTION;  
  
UPDATE student\_info SET age = age + 1 WHERE department = 'Computer Science';  
UPDATE student\_info SET age = age + 1 WHERE department = 'Electronics';  
  
COMMIT;

Q44)

Ans.44) SELECT name, COUNT(\*) AS count  
FROM student\_info  
GROUP BY name  
HAVING count > 1;

Q45)

Ans.45) mysqldump -u root -p college\_db > backup\_college\_db.sql

Q46)

Ans.46) mysql -u root -p college\_db < backup\_college\_db.sql

Q47)

Ans.47) LOAD DATA INFILE '/path/to/file.csv'  
INTO TABLE student\_info  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;

Q48)

Ans.48) CREATE INDEX idx\_name ON student\_info(name);

Q49)

Ans.49) SELECT MAX(marks) FROM marks  
WHERE marks < (SELECT MAX(marks) FROM marks);

Q50)

Ans.50) DROP TABLE courses;  
-- Effect: Permanently deletes the table and its data.