MYSQL OPERATIONS – DEFINITIONS WITH EXAMPLES

1. CREATE DATABASE
Definition: Creates a new database.
Example:
CREATE DATABASE school;

2. USE DATABASE
Definition: Selects which database you want to work on.
Example:
USE school;

3. CREATE TABLE
Definition: Creates a table with columns.
Example:
CREATE TABLE students (id INT PRIMARY KEY, name VARCHAR(50), age INT);

4. DROP TABLE
Definition: Deletes a table permanently.
Example:
DROP TABLE students;

5. ALTER TABLE
Definition: Modifies table structure.
Examples:
ALTER TABLE students ADD email VARCHAR(100);
ALTER TABLE students DROP COLUMN age;
ALTER TABLE students MODIFY name VARCHAR(100);

6. INSERT DATA
Definition: Adds a new row.
Example:
INSERT INTO students (id, name, age) VALUES (1, 'Palak', 21);

7. UPDATE DATA
Definition: Modifies existing rows.
Example:
UPDATE students SET age = 22 WHERE id = 1;

8. DELETE DATA
Definition: Removes rows.
Example:
DELETE FROM students WHERE id = 1;

9. SELECT DATA
Definition: Retrieves data.
Example:
SELECT * FROM students;

10. WHERE CLAUSE
Definition: Filters data.
Example:
SELECT * FROM students WHERE age > 20;

11. ORDER BY
Definition: Sorts rows.
Example:
SELECT * FROM students ORDER BY age DESC;

12. GROUP BY
Definition: Groups data.
Example:
SELECT age, COUNT(*) FROM students GROUP BY age;

13. HAVING
Definition: Filters groups.
Example:
SELECT age, COUNT(*) FROM students GROUP BY age HAVING COUNT(*) > 1;

14. DISTINCT
Definition: Removes duplicates.
Example:
SELECT DISTINCT age FROM students;

15. LIMIT
Definition: Limits records.
Example:
SELECT * FROM students LIMIT 5;

16. JOINS
INNER JOIN:
SELECT s.name, m.marks FROM students s INNER JOIN marks m ON s.id = m.student_id;

LEFT JOIN:
SELECT s.name, m.marks FROM students s LEFT JOIN marks m ON s.id = m.student_id;

RIGHT JOIN:
SELECT s.name, m.marks FROM students s RIGHT JOIN marks m ON s.id = m.student_id;

FULL JOIN (via UNION):
SELECT * FROM A LEFT JOIN B ON A.id = B.id
UNION
SELECT * FROM A RIGHT JOIN B ON A.id = B.id;

17. AGGREGATE FUNCTIONS
COUNT(): SELECT COUNT(*) FROM students;
SUM(): SELECT SUM(salary) FROM employees;
AVG(): SELECT AVG(age) FROM students;
MIN(): SELECT MIN(age) FROM students;
MAX(): SELECT MAX(age) FROM students;

18. LIKE
Example:
SELECT * FROM students WHERE name LIKE 'P%';

19. BETWEEN
Example:

SELECT * FROM students WHERE age BETWEEN 18 AND 25;

20. IN
Example:
SELECT * FROM students WHERE age IN (18, 20, 22);

21. CREATE INDEX
Example:
CREATE INDEX idx_name ON students(name);

22. DROP INDEX
Example:
DROP INDEX idx_name ON students;

23. CREATE VIEW
Example:
CREATE VIEW student_view AS SELECT name, age FROM students;

24. DROP VIEW
Example:
DROP VIEW student_view;

25. CREATE USER
Example:
CREATE USER 'palak'@'localhost' IDENTIFIED BY 'password123';

26. GRANT PRIVILEGES
Example:
GRANT ALL PRIVILEGES ON school.* TO 'palak'@'localhost';

27. REVOKE PRIVILEGES
Example:
REVOKE SELECT ON school.students FROM 'palak'@'localhost';

28. SUBQUERY
Example:
SELECT name FROM students WHERE age > (SELECT AVG(age) FROM students);