Lab 1: Create a new database named school_db and a table called students with the following columns: student_id, student_name, age, class, and address.

```
Ans:-
CREATE DATABASE school_db;

CREATE TABLE students(

student_id int,
student_name varchar(25),
age int,
class varchar(25),
address text

);

student id student name age class address
```

Lab 2: Insert five records into the students table and retrieve all records using the SELECT statement.

ANS:-

```
insert into students VALUES
```

```
(1, 'jaydip', 19, 'chemistry', 'surat'),
(2, 'meet', 21, 'physics', 'ahemdabad'),
(3, 'rohan', 20, 'accounting', 'gota road'),
(4, 'kishan', 17, 'CCC', 'dholka'),
(5, 'nisha', 18, 'biology', 'ahemdabad');
```

student_id	student_name	age	class	address
1	jaydip	19	chemistry	surat
2	meet	21	physics	ahemdabad
3	rohan	20	accounting	gota road
4	kishan	17	CCC	dholka
5	nisha	18	biology	ahemdabad

Lab 1: Write SQL queries to retrieve specific columns (student_name and age) from the students table.

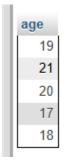
ANS:-

SELECT student name , age FROM students;



Lab 2: Write SQL queries to retrieve all students whose age is greater than 10. ANS :-

SELECT age FROM students WHERE age>10;



Lab 1: Create a table teachers with the following columns: teacher_id (Primary Key), teacher_name (NOT NULL), subject (NOT NULL), and email (UNIQUE).

ANS:-

```
CREATE TABLE teachers (
teacher_id int PRIMARY key,
teacher_name text NOT NULL,
subject varchar(40)NOT NULL,
email text UNIQUE
```

```
INSERT into teachers VALUES
      (1, 'rakeshbhai', 'python', 'rakeshr123@gmail.com'),
      (2, 'brijeshbai', 'mathematics', 'brijeshb456@gmail.com'),
      (3,'umang patel','java backend','umangp777@gmail.com'),
      (4,'rinku mam','java script','rinkur99@gmail.com'),
      (5,'khushi mam','data science','kkhushi87@gmail.com');
 teacher id teacher name
                          subject
                                      email
          1 rakeshbhai
                                      rakeshr123@gmail.com
                          python
          2 brijeshbai
                          mathematics brijeshb456@gmail.com
          3 umang patel
                          java backend umangp777@gmail.com
          4 rinku mam
                          java script rinkur99@gmail.com
          5 khushi mam
                          data science kkhushi87@gmail.com
Lab 2: Implement a FOREIGN KEY constraint to relate the teacher_id from the teachers table
with the
students table.
ANS:-
CREATE TABLE students1(
           student_id int PRIMARY KEY,
            student name varchar(25),
           age int,
           teacher_id int ,
      FOREIGN KEY (teacher id) REFERENCES teachers(teacher id)
```

student_id	student_name	age	teacher_id
1	jaydip	19	3
2	meet	21	1
3	rohan	20	5
4	kishan	17	2
5	nisha	18	4

Lab 1: Create a table courses with columns: course_id, course_name, and course_credits. Set

);

```
the course_id as the primary key.
ANS:-
CREATE TABLE course(
  course id int PRIMARY KEY,
  course name varchar(35),
  course credits int
  );
 course id course name course credits
Lab 2: Use the CREATE command to create a database university_db.
ANS:-
CREATE DATABASE university db;
Lab 1: Modify the courses table by adding a column course_duration using the ALTER
command.
ANS:-
ALTER TABLE course ADD COLUMN course_duration int;
 course id course name course credits course duration
Lab 2: Drop the course_credits column from the courses table
ANS:-
ALTER TABLE course DROP COLUMN course credits;
course id course name course duration
Lab 1: Drop the teachers table from the school db database.
```

```
DROP TABLE teachers;
```

Lab 2: Drop the students table from the school db database and verify that the table has been removed.

ANS:-

DROP TABLE students;

DROP DATABASE school db; Lab 1: Insert three records into the courses table using the INSERT command. ANS:-**INSERT INTO course VALUES** (1,'java-full stack', 9), (2,'SE-Software Engineering', 10), (3,'data analysis', 12); course_id course_name course duration 1 java-full stack 2 SE-Software Engineering 10 3 data analysis 12 Lab 2: Update the course duration of a specific course using the UPDATE command. ANS:-UPDATE course SET course duration=8 WHERE course name='java-full stack'; UPDATE course SET course duration=11 WHERE course name='SE-Software Engineering'; UPDATE course SET course duration=10 WHERE course name='data analysis'; course id course name course duration 1 java-full stack 2 SE-Software Engineering 11 3 data analysis 10

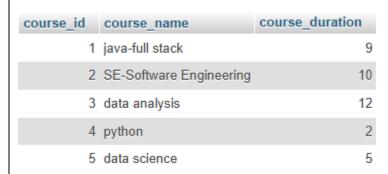
Lab 3: Delete a course with a specific course_id from the courses table using the DELETE command.

ANS:-

DELETE FROM course WHERE course_id=1; DELETE FROM course WHERE course_id=2; DELETE FROM course WHERE course_id=3;

Lab 1: Retrieve all courses from the courses table using the SELECT statement.

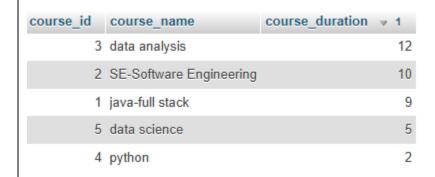
SELECT * FROM course;



Lab 2: Sort the courses based on course_duration in descending order using ORDER BY.

ANS:-

SELECT * FROM course ORDER BY course_duration DESC;



Lab 3: Limit the results of the SELECT query to show only the top two courses using LIMIT.

ANS:-SELECT * FROM COURSE LIMIT>2;

Lab 1: Create two tables: departments and employees. Perform an INNER JOIN to display employees along with their respective departments.

ANS:-

SELECT employee.emp_id , employee.emp_name , department.dep_name FROM employee INNEF JOIN department ON employee.dep_id = department.dep_id;



Lab 2: Use a LEFT JOIN to show all departments, even those without employees.

SELECT department.dep_name , employee.emp_name , department.dep_id FROM employee LEFT JOIN department ON department.dep id=employee.dep id;



Lab 1: Group employees by department and count the number of employees in each departmen using GROUP BY.

ANS:-

SELECT d_name, COUNT(emp_name) AS total_emp FROM employee GROUP BY d_name;



Lab 2: Use the AVG aggregate function to find the average salary of employees in each department.

ANS:-

SELECT AVG(emp_salary) AS aevrage_salary FROM employee;

aevrage_salary 29000.0000 Lab 1: Write a stored procedure to retrieve all employees from the employees table based or department. ANS:-Delimeter \$\$ CREATE PROCEDURE p 1 (e id int, e name varchar(60), d name varchar(60)) BEGIN INSERT INTO employee VALUES (e id , e name , d name); END; CALL p 1(21, 'rohanbhai', 'sales'); CALL p 1(22, 'kishan', 'engineering'); CALL p 1 (23 ,'vrusha','counsellar'); e_id e_name d_name 21 rohanbhai sales 22 kishan engineering 23 vrusha counsellar Lab 2: Write a stored procedure that accepts course_id as input and returns the course details. ANS:-Delimeter \$\$ CREATE PROCEDURE t 13(i int) **BEGIN**

```
Lab 1: Create a view to show all employees along with their department names.
```

CREATE VIEW v_1 AS SELECT emp_name , d_name FROM employee; SELECT * FROM v 1;

END;

ANS:-

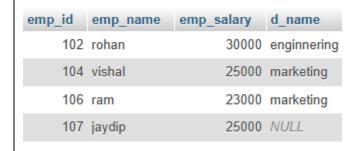
SELECT * FROM course 1 WHERE course id;



Lab 2: Modify the view to exclude employees whose salaries are below \$50,000.

CREATE VIEW v_14 AS SELECT emp_id ,emp_name , emp_salary ,d_name FROM employee WHERE emp_salary<50000;

SELECT * FROM v_14;



Lab 1: Create a trigger to automatically log changes to the employees table when a new employee is added.

ANS:-

DELIMITER \$\$

CREATE TRIGGER t 1 AFTER INSERT ON emp 1 FOR EACH ROW

BEGIN

INSERT INTO dep_1(id , name , records) VALUES(new.e_id ,new.e_name , 'record is successfully inserted.');

END

INSERT INTO emp_1 VALUES(101, 'meet'),(102, 'umang'),(103, 'mehul');

id	name	time_date	records
101	meet	2025-01-17 10:43:00	record is succesfully inserted.
102	umang	2025-01-17 10:43:00	record is succesfully inserted.
103	mehul	2025-01-17 10:43:00	record is succesfully inserted.

Lab 2: Create a trigger to update the last_modified timestamp whenever an employee record is updated.

ANS:-

DELIMITER \$\$

CREATE TRIGGER t_2 AFTER UPDATE ON emp_1 FOR EACH ROW

BEGIN

INSERT INTO dep_1(id , name , records) VALUES(new.e_id ,new.e_name , 'record is successfully updated.');

END

UPDATE emp 1 SET e name='krisha' WHERE e id=101;

id	name	time_date	records
101	meet	2025-01-17 10:43:00	record is succesfully inserted.
102	umang	2025-01-17 10:43:00	record is succesfully inserted.
103	mehul	2025-01-17 10:43:00	record is succesfully inserted.
101	krisha	2025-01-17 10:47:50	record is succesfully updated.