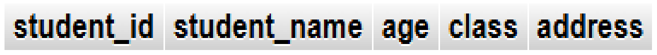
MODULE 4 :- LAB EXCERSISE

• 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 student_db;

CREATE TABLE student(
                    student_id int,
                    student_name text,
                    age int,
                    class text,
                    address text
                    );
```

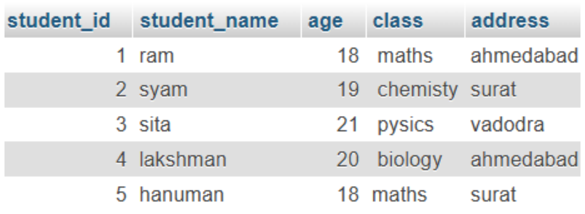


student_id	student_name	age	class	address

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

Ans:

```
INSERT INTO student VALUES(1,'ram',18,' maths','ahmedabad'),(2,'syam',19,'
chemisty','surat'),(3,'sita',21,' pysics','vadodra'),(4,'lakshman',20,'
biology','ahmedabad'),(5,'hanuman',18,'maths','surat');
```

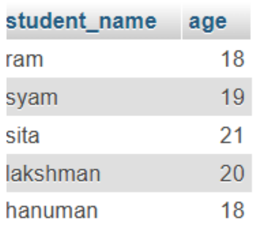


student_id	student_name	age	class	address
1	ram	18	maths	ahmedabad
2	syam	19	chemisty	surat
3	sita	21	pysics	vadodra
4	lakshman	20	biology	ahmedabad
5	hanuman	18	maths	surat

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

Ans:

```
SELECT student_name,age FROM student;
```

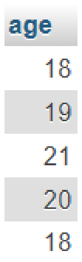


student_name	age
ram	18
syam	19
sita	21
lakshman	20
hanuman	18

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

Ans:

```
SELECT age FROM student WHERE age>10;
```

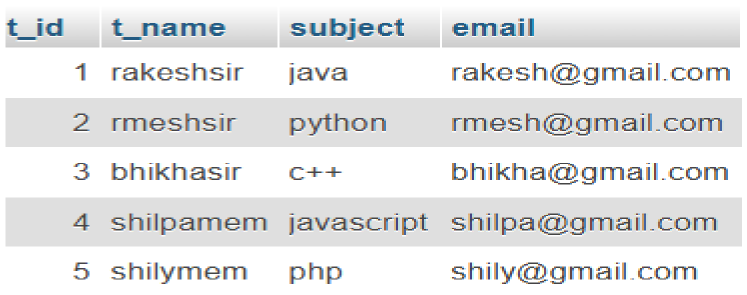


age
18
19
21
20
18

• 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 teacher(
   t_id int PRIMARY KEY,
   t_name text NOT null,
   subject text NOT null,
   email text UNIQUE
   );

INSERT INTO teacher
VALUES(1,'rakeshsir','java','rakesh@gmail.com'),(2,'rmeshsir','python','rmesh@gmail.com'),(3,'bhikhas
ir','c++','bhikha@gmail.com'),(4,'shilpamem','javascript','shilpa@gmail.com'),(5,'shilymem','php','shily
@gmail.com');
```

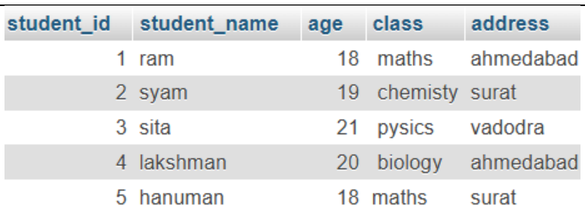


t_id	t_name	subject	email
1	rakeshsir	java	rakesh@gmail.com
2	rmeshsir	python	rmesh@gmail.com
3	bhikhasir	c++	bhikha@gmail.com
4	shilpamem	javascript	shilpa@gmail.com
5	shilymem	php	shily@gmail.com

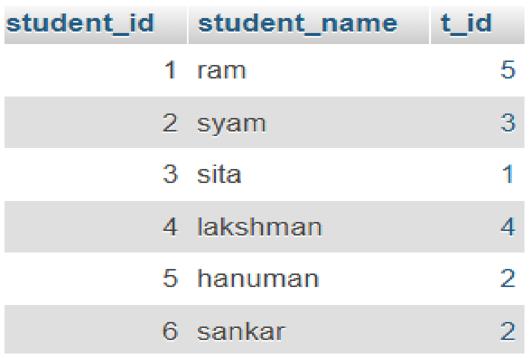
• Lab 2: Implement a FOREIGN KEY constraint to relate the teacher_id from the teachers table with the students table.

Ans:

```
CREATE TABLE student1(
     student_id int PRIMARY KEY,
     student_name text,
     t_id int,
          FOREIGN KEY (t_id) REFERENCES teacher(t_id));
```



student_id	student_name	age	class	address
1	ram	18	maths	ahmedabad
2	syam	19	chemisty	surat
3	sita	21	pysics	vadodra
4	lakshman	20	biology	ahmedabad
5	hanuman	18	maths	surat

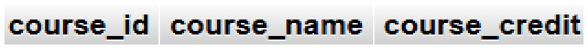


student_id	student_name	t_id
1	ram	5
2	syam	3
3	sita	1
4	lakshman	4
5	hanuman	2
6	sankar	2

• 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 text,
    course_credit INT);
```



course_id	course_name	course_credit

• Lab 2: Use the CREATE command to create a database university_db.

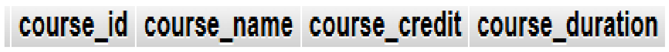
Ans:

```
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 course_duration int;
```

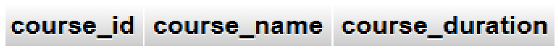


course_id	course_name	course_credit	course_duration

• Lab 2: Drop the course_credits column from the courses table.

Ans:

```
ALTER TABLE course DROP COLUMN course_credit;
```



course_id	course_name	course_duration

• Lab 1: Drop the teachers table from the school_db database

Ans:

DROP TABLE teacher;

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

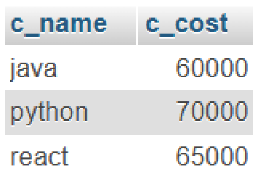
Ans:

DROP TABLE student1;
DROP DATABASE student_db;

• Lab 1: Insert three records into the courses table using the INSERT command.

Ans:

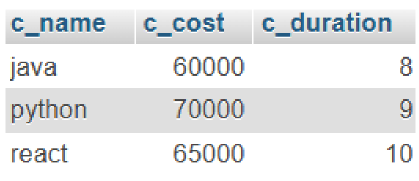
INSERT INTO course VALUES('java',60000),('python',70000),('react',65000);



c_name	c_cost
java	60000
python	70000
react	65000

• Lab 2: Update the course duration of a specific course using the UPDATE command.

Ans:
UPDATE course SET c_duration=8 WHERE c_name='java';
UPDATE course SET c_duration=9 WHERE c_name='python';
UPDATE course SET c_duration=10 WHERE c_name='react';



c_name	c_cost	c_duration
java	60000	8
python	70000	9
react	65000	10

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

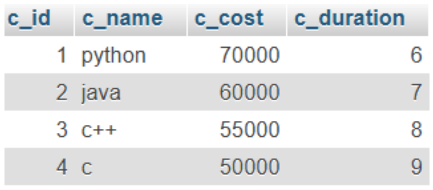
Ans:

DELETE FROM course WHERE c_name='java';
DELETE FROM course WHERE c_name='python';
DELETE FROM course WHERE c_name='react';

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

Ans:

SELECT * FROM course1;

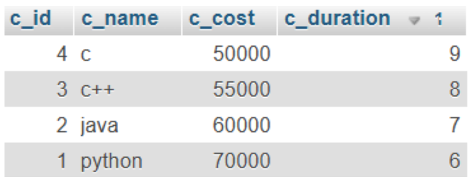


c_id	c_name	c_cost	c_duration
1	python	70000	6
2	java	60000	7
3	c++	55000	8
4	c	50000	9

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

Ans:

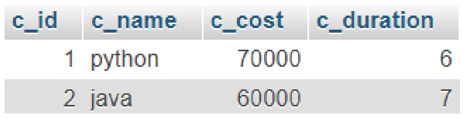
SELECT * FROM course1 ORDER BY c_duration DESC;



c_id	c_name	c_cost	c_duration
4	c	50000	9
3	c++	55000	8
2	java	60000	7
1	python	70000	6

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

Ans:
SELECT * FROM course1 c_cost LIMIT 2;

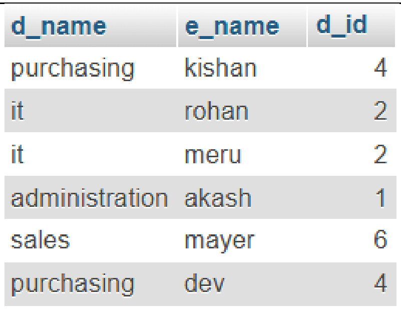


c_id	c_name	c_cost	c_duration
1	python	70000	6
2	java	60000	7

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

Ans:

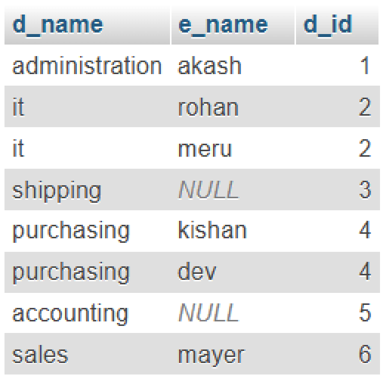
SELECT departmenr.d_name,emp.e_name,departmenr.d_id FROM departmenr INNER JOIN emp ON departmenr.d_id=emp.d_id;



d_name	e_name	d_id
purchasing	kishan	4
it	rohan	2
it	meru	2
administration	akash	1
sales	mayer	6
purchasing	dev	4

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

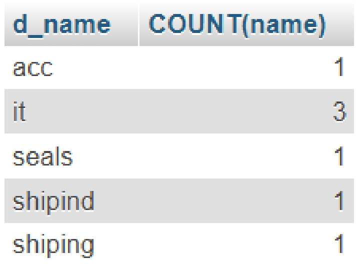
Ans:
SELECT departmenr.d_name,emp.e_name,departmenr.d_id FROM departmenr LEFT JOIN emp ON de partmenr.d_id=emp.d_id;



d_name	e_name	d_id
administration	akash	1
it	rohan	2
it	meru	2
shipping	*NULL*	3
purchasing	kishan	4
purchasing	dev	4
accounting	*NULL*	5
sales	mayer	6

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

Ans:
SELECT d_name,COUNT(name) FROM emp1 GROUP BY d_name;

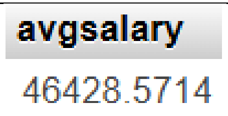


d_name	COUNT(name)
acc	1
it	3
seals	1
shipind	1
shiping	1

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

Ans:
SELECT AVG(salary) AS avgsalary FROM emp1;

MODULE 4 :- LAB EXCERSISE

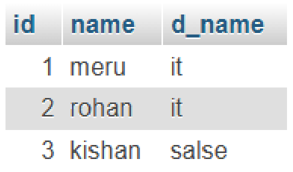


avgsalary
46428.5714

• Lab 1: Write a stored procedure to retrieve all employees from the employees table based on department.

Ans:

```
CREATE PROCEDURE ins1(id int,name text,d_name text)
BEGIN
INSERT INTO emp VALUES(id,name,d_name);
End

CALL ins1(1,'meru',’it');
CALL ins1(2,'rohan','it');
CALL ins1(3,'kishan','salse');
```



id	name	d_name
1	meru	it
2	rohan	it
3	kishan	salse

• Lab 2: Write a stored procedure that accepts course_id as input and returns the course details.

Ans:

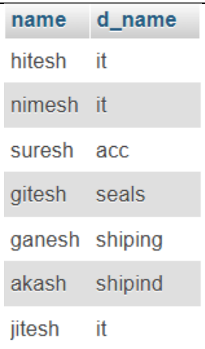
```
Delimeter $$
Create procedure t_13( in I int   , out course_d text)
Begin
Select coursedetail   into course_d from course_1 where i=course_id;
End

Call t_13(1 ,@coursedetail);
Select @coursedetail;
```

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

Ans:

```
CREATE VIEW v1 AS SELECT name,d_name FROM emp1;
SELECT * FROM v1;
```

MODULE 4 :- LAB EXCERSISE

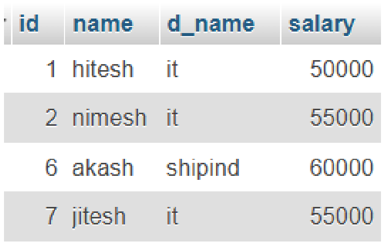


name	d_name
hitesh	it
nimesh	it
suresh	acc
gitesh	seals
ganesh	shiping
akash	shipind
jitesh	it

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

Ans:

CREATE VIEW v12 AS SELECT * FROM emp1 WHERE salary>=50000;
SELECT * FROM v12;

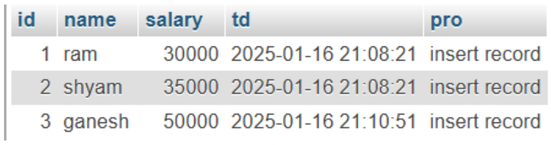


id	name	d_name	salary
1	hitesh	it	50000
2	nimesh	it	55000
6	akash	shipind	60000
7	jitesh	it	55000

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

Ans:
CREATE TRIGGER t1 AFTER insert ON em FOR EACH ROW BEGIN INSERT INTO emdata(id,name,salary,pro) VALUES(new.id,new.name,new.salary,'insert record'); END

INSERT INTO em VALUES(1,'ram',30000),(2,'shyam',35000);
INSERT INTO em VALUES(3,'ganesh',50000);

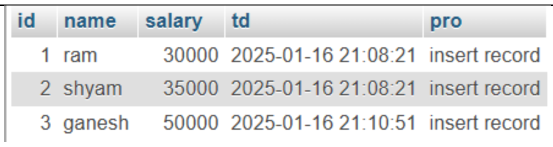


id	name	salary	td	pro
1	ram	30000	2025-01-16 21:08:21	insert record
2	shyam	35000	2025-01-16 21:08:21	insert record
3	ganesh	50000	2025-01-16 21:10:51	insert record

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

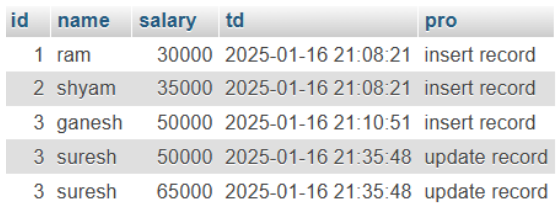
Ans:

MODULE 4 :- LAB EXCERSISE



id	name	salary	td	pro
1	ram	30000	2025-01-16 21:08:21	insert record
2	shyam	35000	2025-01-16 21:08:21	insert record
3	ganesh	50000	2025-01-16 21:10:51	insert record

```
CREATE TRIGGER t2 AFTER UPDATE ON em FOR EACH ROW BEGIN INSERT INTO emdata(id,name,salary
,pro) VALUES (new.id,new.name,new.salary,'update record'); END;

UPDATE em SET name='suresh' WHERE id=3;
UPDATE em SET salary=65000 WHERE id=3;
```



id	name	salary	td	pro
1	ram	30000	2025-01-16 21:08:21	insert record
2	shyam	35000	2025-01-16 21:08:21	insert record
3	ganesh	50000	2025-01-16 21:10:51	insert record
3	suresh	50000	2025-01-16 21:35:48	update record
3	suresh	65000	2025-01-16 21:35:48	update record