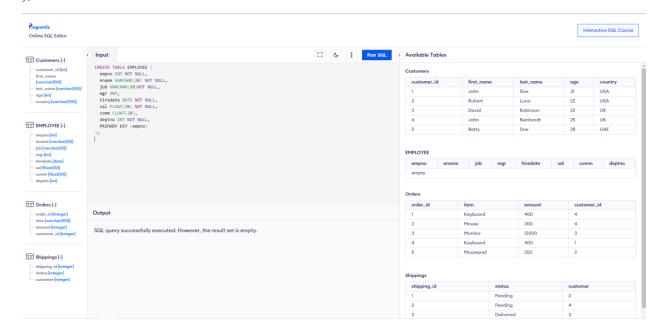
Question 1 (20 Credits)

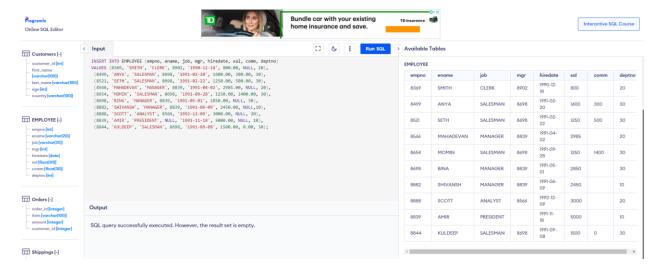
a. Create the table named EMPLOYEE. Insert the employee details below and assign a primary key.

```
CREATE TABLE EMPLOYEE (
empno INT NOT NULL,
ename VARCHAR(20) NOT NULL,
job VARCHAR(20)NOT NULL,
mgr INT,
hiredate DATE NOT NULL,
sal FLOAT(30) NOT NULL,
comm FLOAT(30),
deptno INT NOT NULL,
PRIMARY KEY (empno)
);
```



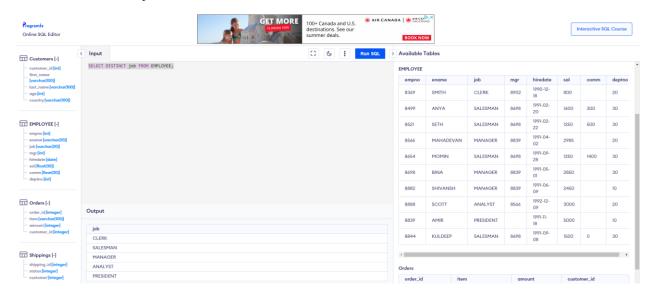
INSERT INTO EMPLOYEE (empno, ename, job, mgr, hiredate, sal, comm, deptno) VALUES (8369, 'SMITH', 'CLERK', 8902, '1990-12-18', 800.00, NULL, 20), (8499, 'ANYA', 'SALESMAN', 8698, '1991-02-20', 1600.00, 300.00, 30), (8521, 'SETH', 'SALESMAN', 8698, '1991-02-22', 1250.00, 500.00, 30), (8566, 'MAHADEVAN', 'MANAGER', 8839, '1991-04-02', 2985.00, NULL, 20),

(8654, 'MOMIN', 'SALESMAN', 8698, '1991-09-28', 1250.00, 1400.00, 30), (8698, 'BINA', 'MANAGER', 8839, '1991-05-01', 2850.00, NULL, 30), (8882, 'SHIVANSH', 'MANAGER', 8839, '1991-06-09', 2450.00, NULL, 10), (8888, 'SCOTT', 'ANALYST', 8566, '1992-12-09', 3000.00, NULL, 20), (8839, 'AMIR', 'PRESIDENT', NULL, '1991-11-18', 5000.00, NULL, 10), (8844, 'KULDEEP', 'SALESMAN', 8698, '1991-09-08', 1500.00, 0.00, 30);



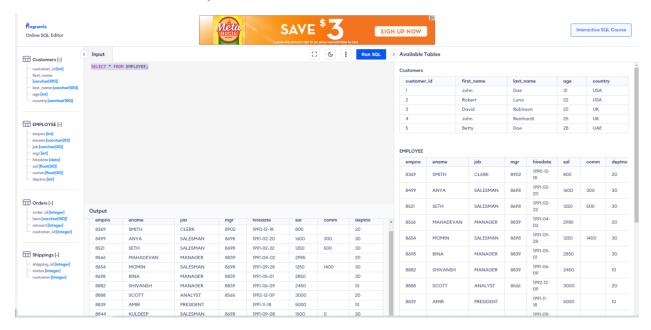
b. Return only the jobs from the table. List them only once (1).

SELECT DISTINCT job FROM EMPLOYEE;



c. Return all records from the table. (2)

SELECT * FROM EMPLOYEE;

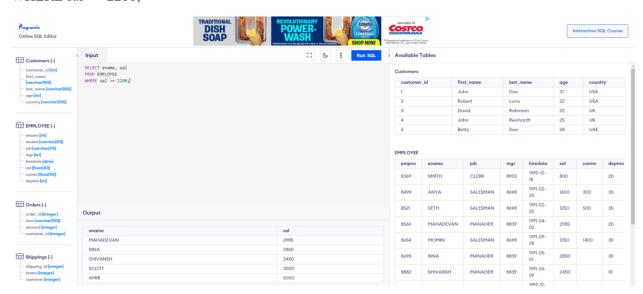


d. Return employee name and salary of employees whose salaries are greater than or equal to 2200. (2)

SELECT ename, sal

FROM EMPLOYEE

WHERE sal \geq 2200;

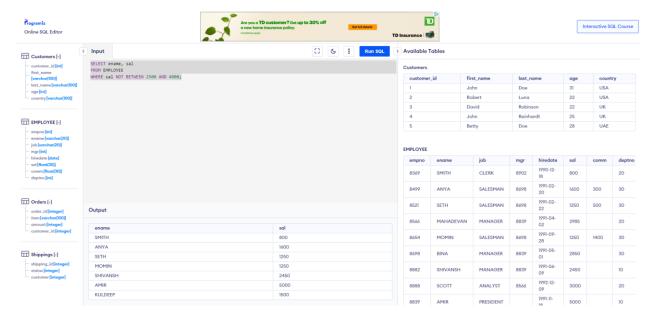


e. Return employee name and salary of those employees who do not have their salary in the range of 2500 to 4000. (2)

SELECT ename, sal

FROM EMPLOYEE

WHERE sal NOT BETWEEN 2500 AND 4000;

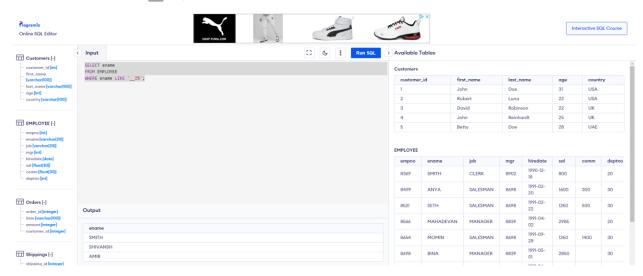


f. Return employee name whose name contains "I" as third character. (2)

SELECT ename

FROM EMPLOYEE

WHERE ename LIKE '__I%';

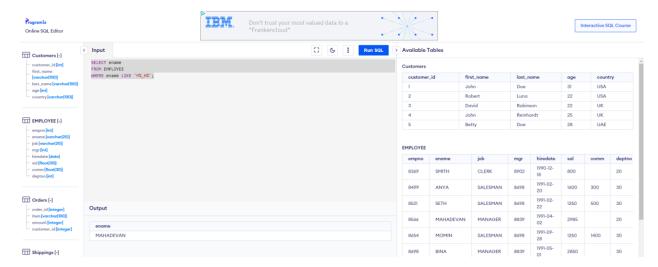


g. Return employee name whose name contains" M" as first and "H" as third character. (2)

SELECT ename

FROM EMPLOYEE

WHERE ename LIKE 'M% H%';

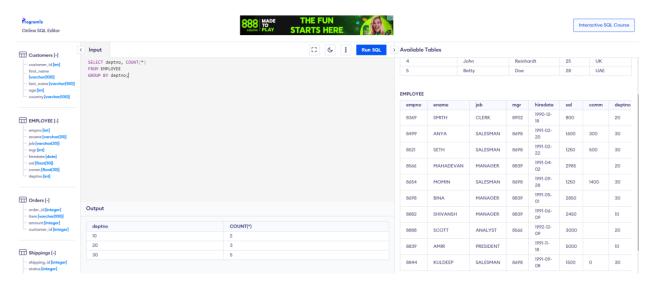


h. Count the number of employees in each department, with the department number along with the employee count. (2)

SELECT deptno, COUNT(*)

FROM EMPLOYEE

GROUP BY deptno;

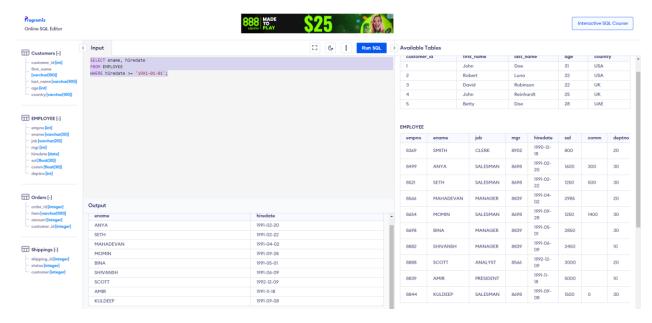


i. Return employee name and hire dates of employees hired after 1991-01-01. (2)

SELECT ename, hiredate

FROM EMPLOYEE

WHERE hiredate >= '1991-01-01';

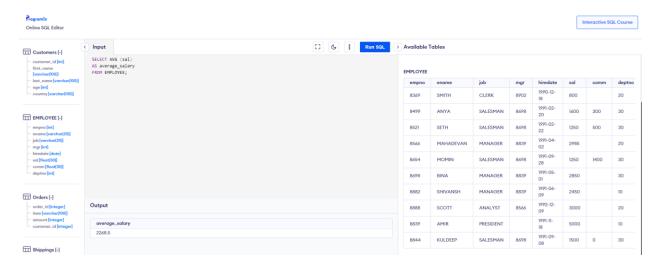


j. Calculate the average salary of all employees. (2)

SELECT AVG (sal)

AS average_salary

FROM EMPLOYEE;



Question 2 (20 Credits)

a) Create the following tables and name them student and project respectively. (2)

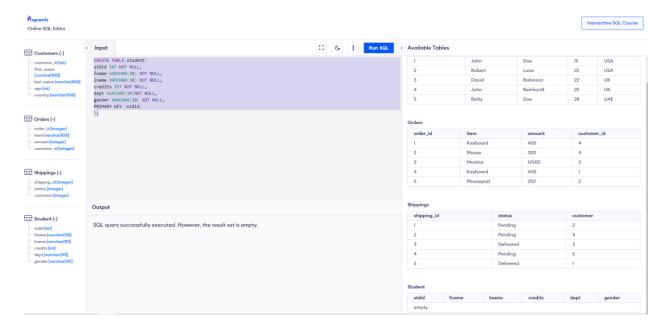
CREATE TABLE student(

stdid INT NOT NULL,

fname VARCHAR(30) NOT NULL,

Iname VARCHAR(30) NOT NULL, credits INT NOT NULL, dept VARCHAR(30)NOT NULL, gender VARCHAR(20) NOT NULL, PRIMARY KEY (stdid)

);



INSERT INTO student(stdid, fname, lname, credits, dept,gender)

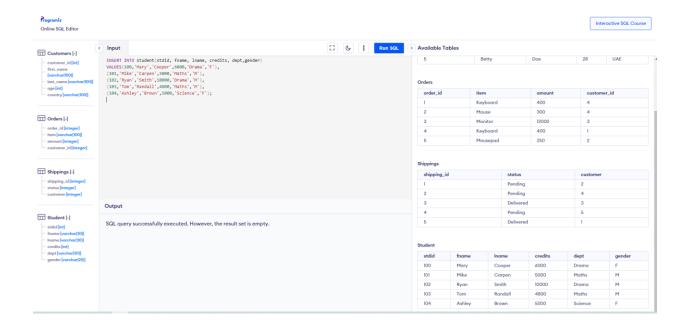
VALUES(100,'Mary','Cooper',6000,'Drama','F'),

(101,'Mike','Carpen',5000,'Maths','M'),

(102,'Ryan','Smith',10000,'Drama','M'),

(103,'Tom','Randall',4800,'Maths','M'),

(104,'Ashley','Brown',5000,'Science','F');

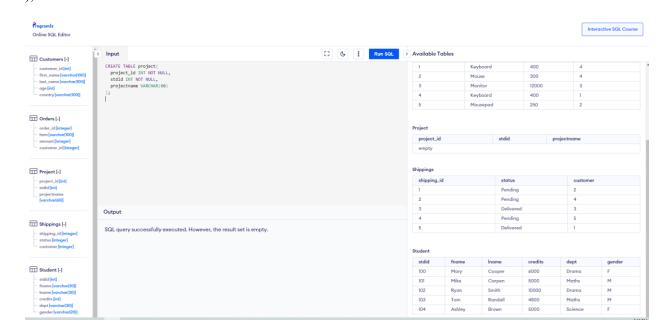


CREATE TABLE project(

project_id INT NOT NULL,

stdid INT NOT NULL,

projectname VARCHAR(60)
);



INSERT INTO project(project id, stdid, projectname)

VALUES(1,100,'Shakespeare'),

(2,100,'Greek Tragedy'),

(3,100,'Disaster'),

(4,101,'Trigonometry'),

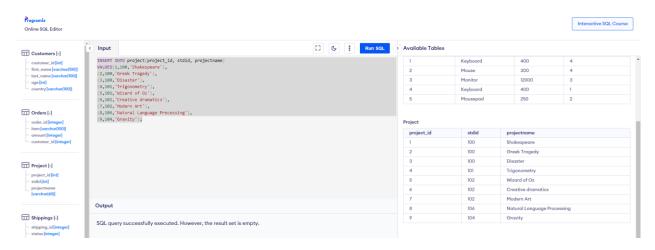
(5,102,'Wizard of Oz'),

(6,102,'Creative dramatics'),

(7,102,'Modern Art'),

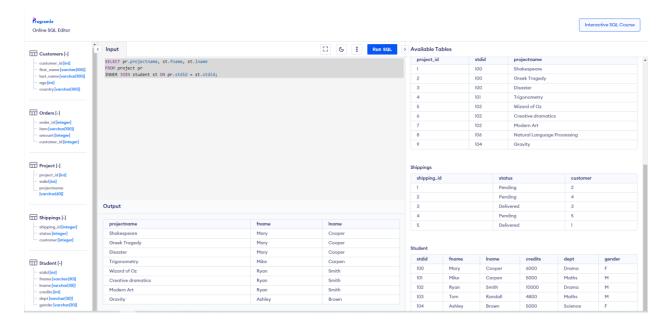
(8,106,'Natural Language Processing'),

(9,104,'Gravity');



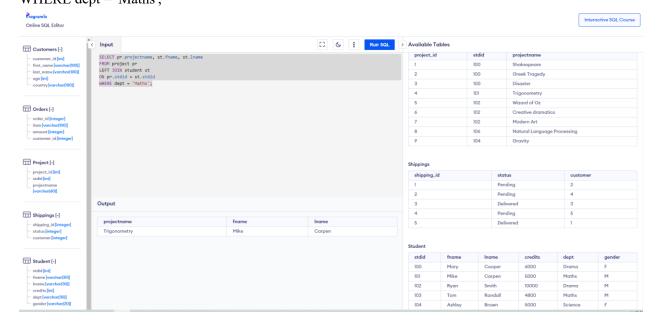
b) Return the project names along with the corresponding student details (first name, last name) for all projects. (2)

SELECT pr.projectname, st.fname, st.lname FROM project pr INNER JOIN student st ON pr.stdid = st.stdid;



c) Return the project names and the corresponding student details (first name, last name) for projects assigned to students in the "Maths" department. (2)

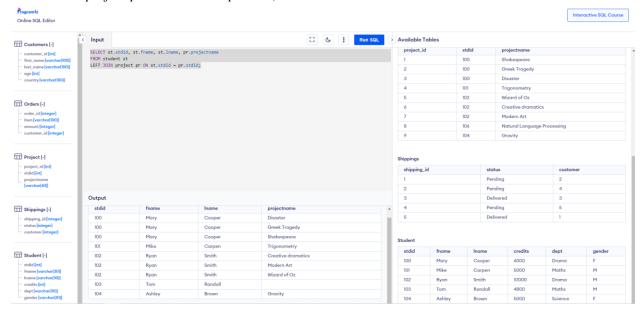
SELECT pr.projectname, st.fname, st.lname FROM project pr LEFT JOIN student st ON pr.stdid = st.stdid WHERE dept = 'Maths';



d) Return all students along with the project names (if any) they are assigned to. (2)

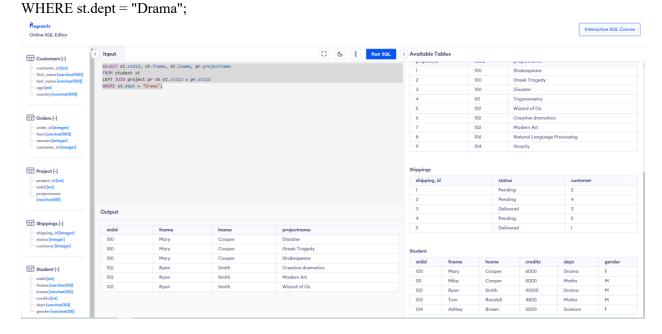
SELECT st.stdid, st.fname, st.lname, pr.projectname

FROM student st LEFT JOIN project pr ON st.stdid = pr.stdid;



e) Return all students and the project names (if any) they are assigned to, but only for the students in the "Drama" department. (2)

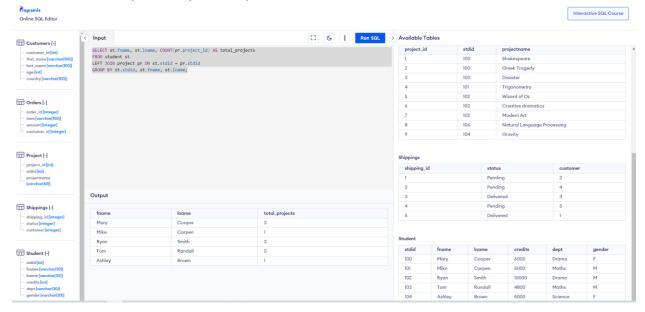
SELECT st.stdid, st.fname, st.lname, pr.projectname FROM student st LEFT JOIN project pr ON st.stdid = pr.stdid



f) Return the total number of projects each student is assigned to, along with their details (first name, last name). (2)

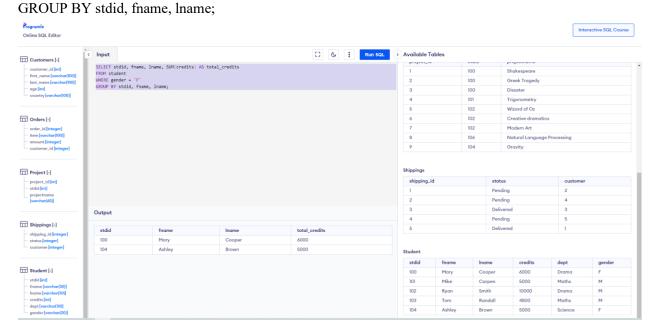
SELECT st.fname, st.lname, COUNT(pr.project id) AS total projects

FROM student st LEFT JOIN project pr ON st.stdid = pr.stdid GROUP BY st.stdid, st.fname, st.lname;



g) Return all students with the gender "F", with their total credits (2)

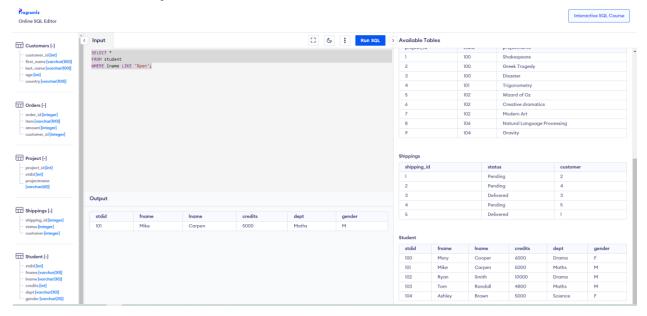
SELECT stdid, fname, lname, SUM(credits) AS total_credits FROM student
WHERE gender = 'F'



h) Return all students whose last name ends with "pen". (2)

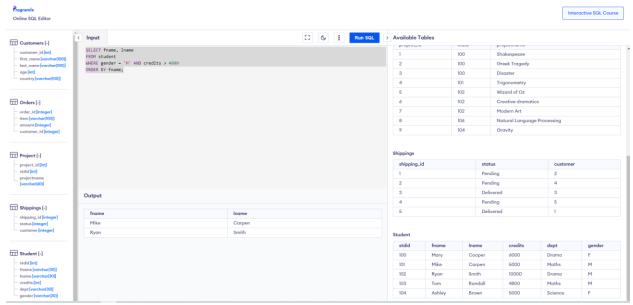
SELECT *
FROM student

WHERE lname LIKE '%pen';



i) Return the names of students with the gender "M", with more than 4999 credits, sorted alphabetically by their first names. (2)

SELECT fname, lname FROM student WHERE gender = 'M' AND credits > 4999 ORDER BY fname;



j) Return all students' details and sort them based on their credits in descending order. (2)

SELECT *
FROM student
ORDER BY credits DESC;

| Interactive SQL Course | Cou

Shippings [-]
shipping, [s [Interpret]
status [interpret]
customer [integer]
statud [interpret]
statud [interpret]
statud [interpret]
statud [interpret]
statud [interpret]
statud [interpret | interpret | interp

tdid	fname	Iname	credits	dept	gender
102	Ryan	Smith	10000	Drama	М
100	Mary	Cooper	6000	Drama	F
101	Mike	Carpen	5000	Maths	М
104	Ashley	Brown	5000	Science	F
103	Tom	Randall	4800	Maths	М

stdid	fname	Iname	credits	dept	gender
100	Mary	Cooper	6000	Drama	F
101	Mike	Carpen	5000	Maths	М
102	Ryan	Smith	10000	Drama	М
103	Tom	Randall	4800	Maths	М