

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)  
);
```

Programiz
Online SQL Editor

Interactive SQL Course

Input

```
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)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
empty							

Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3

```
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);

Programiz Online SQL Editor

Bundle car with your existing home insurance and save. TD Insurance

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shipings [-]

Input

```
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);
```

Run SQL

Available Tables

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

SQL query successfully executed. However, the result set is empty.

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

SELECT DISTINCT job FROM EMPLOYEE;

Programiz Online SQL Editor

GET MORE FLIGHTS. 100+ Canada and U.S. destinations. See our summer deals. AIR CANADA. BOOK NOW

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shipings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Input

```
SELECT DISTINCT job FROM EMPLOYEE;
```

Run SQL

Available Tables

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

job
CLERK
SALESMAN
MANAGER
ANALYST
PRESIDENT

Orders

order_id	item	amount	customer_id
----------	------	--------	-------------

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

SELECT * FROM EMPLOYEE;

Programiz Online SQL Editor

Meta MUCIL SAVE \$3 SIGN UP NOW

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Input

```
SELECT * FROM EMPLOYEE;
```

Run SQL

Available Tables

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

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 >= 2200;

Programiz Online SQL Editor

TRADITIONAL DISH SOAP REVOLUTIONARY POWER-WASH AVAILABLE AT: COSTCO SHOP NOW

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Input

```
SELECT ename, sal  
FROM EMPLOYEE  
WHERE sal >= 2200;
```

Run SQL

Available Tables

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

ename	sal
MAHADEVAN	2985
BINA	2850
SHIVANSH	2450
SCOTT	3000
AMIR	5000

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;

Programiz Online SQL Editor

Are you a TD customer? Get up to 30% off a new home insurance policy. [See full details](#)

Interactive SQL Course

```
SELECT ename, sal
FROM EMPLOYEE
WHERE sal NOT BETWEEN 2500 AND 4000;
```

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10

Output

ename	sal
SMITH	800
ANYA	1600
SETH	1250
MOMIN	1250
SHIVANSH	2450
AMIR	5000
KULDEEP	1500

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

SELECT ename

FROM EMPLOYEE

WHERE ename LIKE '__I%';

Programiz Online SQL Editor

SHOP PUMA.COM

Interactive SQL Course

```
SELECT ename
FROM EMPLOYEE
WHERE ename LIKE '__I%';
```

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30

Output

ename
SMITH
SHIVANSH
AMIR

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%';

Programiz Online SQL Editor

IBM Don't trust your most valued data to a "Frankencloud"

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]

Input

```
SELECT ename
FROM EMPLOYEE
WHERE ename LIKE 'M%';
```

Run SQL

Available Tables

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30

Output

ename
MAHADEVAN

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;

Programiz Online SQL Editor

888 MADE TO PLAY THE FUN STARTS HERE

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]

Input

```
SELECT deptno, COUNT(*)
FROM EMPLOYEE
GROUP BY deptno;
```

Run SQL

Available Tables

Customers

customer_id	first_name	last_name	age	country
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

deptno	COUNT(*)
10	2
20	3
30	5

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';

Programiz Online SQL Editor

888 MADE TO PLAY \$25

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Input

```
SELECT ename, hiredate
FROM EMPLOYEE
WHERE hiredate >= '1991-01-01';
```

Run SQL

Available Tables

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

ename	hiredate
ANYA	1991-02-20
SETH	1991-02-22
MAHADEVAN	1991-04-02
MOMIN	1991-09-28
BINA	1991-05-01
SHIVANSH	1991-06-09
SCOTT	1992-12-09
AMIR	1991-11-18
KULDEEP	1991-09-08

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

SELECT AVG (sal)

AS average_salary

FROM EMPLOYEE;

Programiz Online SQL Editor

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

EMPLOYEE [-]

- empno [int]
- ename [varchar(20)]
- job [varchar(20)]
- mgr [int]
- hiredate [date]
- sal [float(30)]
- comm [float(30)]
- deptno [int]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Input

```
SELECT AVG (sal)
AS average_salary
FROM EMPLOYEE;
```

Run SQL

Available Tables

EMPLOYEE

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	SMITH	CLERK	8902	1990-12-18	800		20
8499	ANYA	SALESMAN	8698	1991-02-20	1600	300	30
8521	SETH	SALESMAN	8698	1991-02-22	1250	500	30
8566	MAHADEVAN	MANAGER	8839	1991-04-02	2985		20
8654	MOMIN	SALESMAN	8698	1991-09-28	1250	1400	30
8698	BINA	MANAGER	8839	1991-05-01	2850		30
8882	SHIVANSH	MANAGER	8839	1991-06-09	2450		10
8888	SCOTT	ANALYST	8566	1992-12-09	3000		20
8839	AMIR	PRESIDENT		1991-11-18	5000		10
8844	KULDEEP	SALESMAN	8698	1991-09-08	1500	0	30

Output

average_salary
2268.5

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,

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

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Student [-]

- stdid [int]
- fname [varchar(30)]
- lname [varchar(30)]
- credits [int]
- dept [varchar(30)]
- gender [varchar(20)]

Input

```
CREATE TABLE student(
stdid INT NOT NULL,
fname VARCHAR(30) NOT NULL,
lname VARCHAR(30) NOT NULL,
credits INT NOT NULL,
dept VARCHAR(30) NOT NULL,
gender VARCHAR(20) NOT NULL,
PRIMARY KEY (stdid)
);
```

Run SQL

Available Tables

1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
empty					

Output

SQL query successfully executed. However, the result set is empty.

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');

Programiz

Online SQL Editor

Interactive SQL Course

Customers [-]

customer_id [int]

first_name [varchar(100)]

last_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order_id [integer]

item [varchar(100)]

amount [integer]

customer_id [integer]

Shippings [-]

shipping_id [integer]

status [integer]

customer [integer]

Student [-]

stdid [int]

fname [varchar(20)]

lname [varchar(20)]

credits [int]

dept [varchar(30)]

gender [varchar(20)]

Input

```

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');

```

Run SQL

Available Tables

5	Betty	Doe	28	UAE
---	-------	-----	----	-----

Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

SQL query successfully executed. However, the result set is empty.

```

CREATE TABLE project(
    project_id INT NOT NULL,
    stdid INT NOT NULL,
    projectname VARCHAR(60)
);

```

Programiz

Online SQL Editor

Interactive SQL Course

Customers [-]

customer_id [int]

first_name [varchar(100)]

last_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order_id [integer]

item [varchar(100)]

amount [integer]

customer_id [integer]

Project [-]

project_id [int]

stdid [int]

projectname [varchar(60)]

Shippings [-]

shipping_id [integer]

status [integer]

customer [integer]

Student [-]

stdid [int]

fname [varchar(20)]

lname [varchar(20)]

credits [int]

dept [varchar(30)]

gender [varchar(20)]

Input

```

CREATE TABLE project(
    project_id INT NOT NULL,
    stdid INT NOT NULL,
    projectname VARCHAR(60)
);

```

Run SQL

Available Tables

1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Project

project_id	stdid	projectname
empty		

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

SQL query successfully executed. However, the result set is empty.


```

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');

```

Programiz Online SQL Editor Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Project [-]

- project_id [int]
- stdid [int]
- projectname [varchar(60)]

Shipments [-]

- shipping_id [integer]
- status [integer]

Input

```

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');

```

Run SQL

Available Tables

1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Project

project_id	stdid	projectname
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

Output

SQL query successfully executed. However, the result set is empty.

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;

```

Programiz Online SQL Editor

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Project [-]

- project_id [int]
- stdid [int]
- projectname [varchar(60)]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Student [-]

- stdid [int]
- fname [varchar(30)]
- lname [varchar(30)]
- credits [int]
- dept [varchar(30)]
- gender [varchar(20)]

Input

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

Run SQL

Available Tables

project_id	stdid	projectname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

projectname	fname	lname
Shakespeare	Mary	Cooper
Greek Tragedy	Mary	Cooper
Disaster	Mary	Cooper
Trigonometry	Mike	Carpen
Wizard of Oz	Ryan	Smith
Creative dramatics	Ryan	Smith
Modern Art	Ryan	Smith
Gravity	Ashley	Brown

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';
```

Programiz Online SQL Editor

Interactive SQL Course

Customers [-]

- customer_id [int]
- first_name [varchar(100)]
- last_name [varchar(100)]
- age [int]
- country [varchar(100)]

Orders [-]

- order_id [integer]
- item [varchar(100)]
- amount [integer]
- customer_id [integer]

Project [-]

- project_id [int]
- stdid [int]
- projectname [varchar(60)]

Shippings [-]

- shipping_id [integer]
- status [integer]
- customer [integer]

Student [-]

- stdid [int]
- fname [varchar(30)]
- lname [varchar(30)]
- credits [int]
- dept [varchar(30)]
- gender [varchar(20)]

Input

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

Run SQL

Available Tables

project_id	stdid	projectname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

projectname	fname	lname
Trigonometry	Mike	Carpen

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;

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(40)]

Shippings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

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

Run SQL

Available Tables

project_id	stdid	projectname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
100	Mary	Cooper	6000	Drama	F
101	Mike	Carpenter	5000	Maths	M
102	Ryan	Smith	10000	Drama	M
103	Tom	Randall	4800	Maths	M
104	Ashley	Brown	5000	Science	F

Output

stdid	fname	lname	projectname
100	Mary	Cooper	Disaster
100	Mary	Cooper	Greek Tragedy
100	Mary	Cooper	Shakespeare
101	Mike	Carpenter	Trigonometry
102	Ryan	Smith	Creative dramatics
102	Ryan	Smith	Modern Art
102	Ryan	Smith	Wizard of Oz
103	Tom	Randall	
104	Ashley	Brown	Gravity

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
WHERE st.dept = "Drama";

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(40)]

Shippings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

```
SELECT st.stdid, st.fname, st.lname, pr.projectname
FROM student st
LEFT JOIN project pr ON st.stdid = pr.stdid
WHERE st.dept = "Drama";
```

Run SQL

Available Tables

project_id	stdid	projectname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
100	Mary	Cooper	6000	Drama	F
101	Mike	Carpenter	5000	Maths	M
102	Ryan	Smith	10000	Drama	M
103	Tom	Randall	4800	Maths	M
104	Ashley	Brown	5000	Science	F

Output

stdid	fname	lname	projectname
100	Mary	Cooper	Disaster
100	Mary	Cooper	Greek Tragedy
100	Mary	Cooper	Shakespeare
102	Ryan	Smith	Creative dramatics
102	Ryan	Smith	Modern Art
102	Ryan	Smith	Wizard of Oz

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;

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(60)]

Shipings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

```
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;
```

Run SQL

Available Tables

project_id	stdid	projectname
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

Shipings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

fname	lname	total_projects
Mary	Cooper	3
Mike	Carpen	1
Ryan	Smith	3
Tom	Randall	0
Ashley	Brown	1

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'
GROUP BY stdid, fname, lname;

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(60)]

Shipings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

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

Run SQL

Available Tables

project_id	stdid	projectname
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

Shipings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

stdid	fname	lname	total_credits
100	Mary	Cooper	6000
104	Ashley	Brown	5000

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

SELECT *
FROM student

WHERE lname LIKE '%pen';

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(40)]

Shippings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

```
SELECT *  
FROM student  
WHERE lname LIKE '%pen';
```

Run SQL

Available Tables

id	fname	lname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

stdid	fname	lname	credits	dept	gender
101	Mike	Carpen	5000	Maths	M

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;
```

Programiz
Online SQL Editor

Interactive SQL Course

Customers [-]
customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

Orders [-]
order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

Project [-]
project_id [int]
stdid [int]
projectname [varchar(40)]

Shippings [-]
shipping_id [integer]
status [integer]
customer [integer]

Student [-]
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]

Input

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

Run SQL

Available Tables

id	fname	lname
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

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

fname	lname
Mike	Carpen
Ryan	Smith

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

```
SELECT *  
FROM student  
ORDER BY credits DESC;
```

Customers [-]

```
customer_id [int]
first_name [varchar(30)]
last_name [varchar(30)]
age [int]
country [varchar(100)]
```

Orders [-]

```
order_id [integer]
item [varchar(300)]
amount [integer]
customer_id [integer]
```

Project [-]

```
project_id [int]
stdid [int]
projectname [varchar(60)]
```

Shippings [-]

```
shipping_id [integer]
status [integer]
customer [integer]
```

Student [-]

```
stdid [int]
fname [varchar(30)]
lname [varchar(30)]
credits [int]
dept [varchar(30)]
gender [varchar(20)]
```

Input

```
SELECT *
FROM student
ORDER BY credits DESC;
```

Run SQL

Available Tables

projectname	stdid	projectname
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	104	Natural Language Processing
9	104	Gravity

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Student

stdid	fname	lname	credits	dept	gender
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

Output

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