Prashant Bharti **Roll No. 202251102**

Indian Institute of Information Technology Vadodara

Database Management Systems Laboratory Assignment

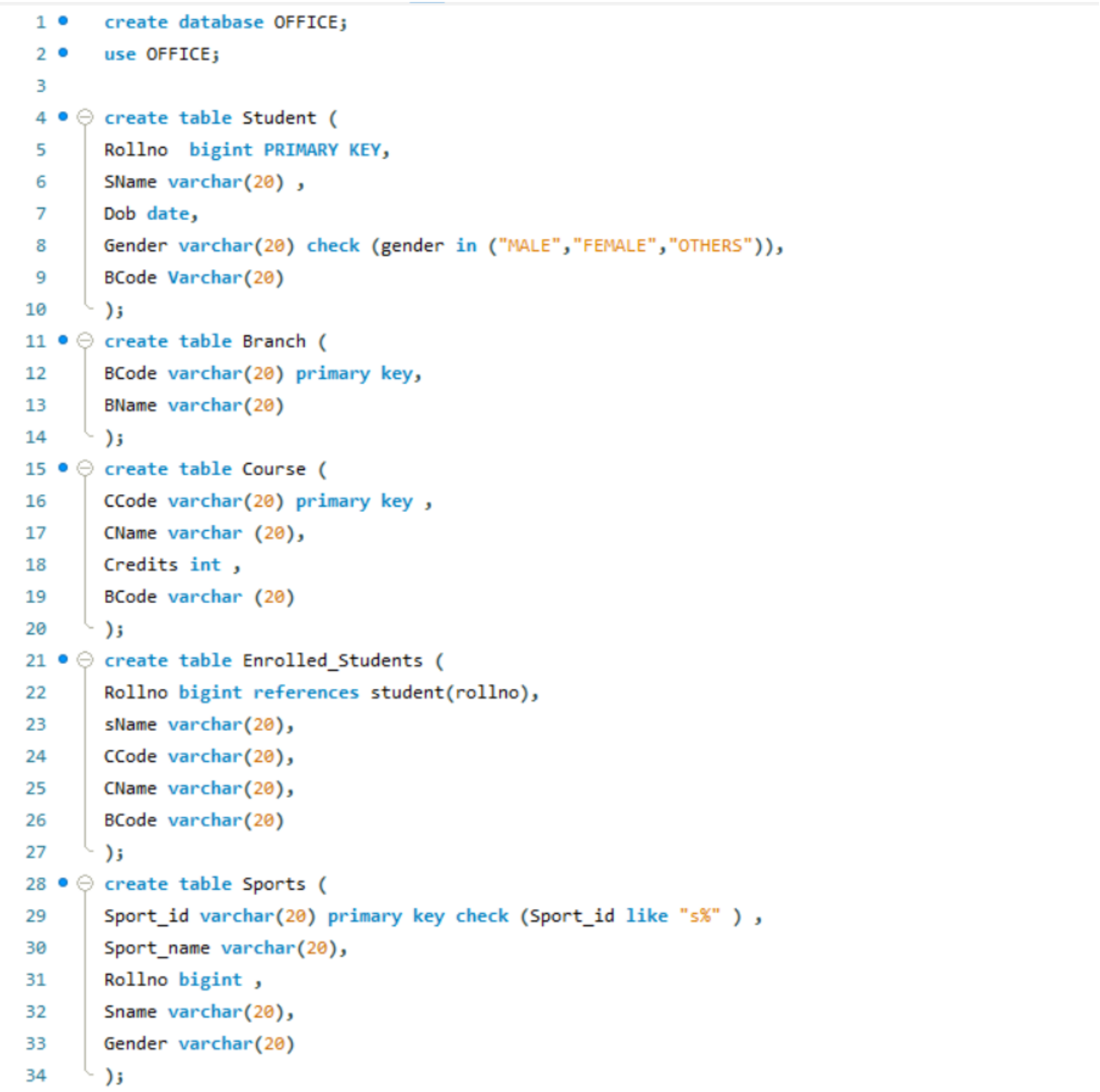
Week 1

-by **PRASHANT BHARTI**

(202251102)

QUESTION 1

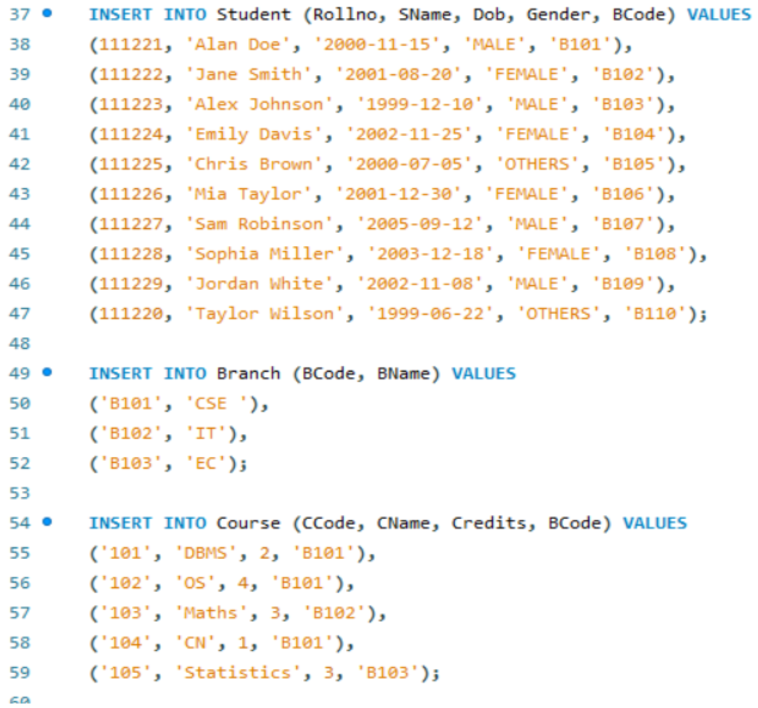
Implement the following queries:



```sql
create database OFFICE;
use OFFICE;

create table Student (
Rollno  bigint PRIMARY KEY,
SName varchar(20) ,
Dob date,
Gender varchar(20) check (gender in ("MALE","FEMALE","OTHERS")),
BCode Varchar(20)
);
create table Branch (
BCode varchar(20) primary key,
BName varchar(20)
);
create table Course (
CCode varchar(20) primary key ,
CName varchar (20),
Credits int ,
BCode varchar (20)
);
create table Enrolled_Students (
Rollno bigint references student(rollno),
sName varchar(20),
CCode varchar(20),
CName varchar(20),
BCode varchar(20)
);
create table Sports (
Sport_id varchar(20) primary key check (Sport_id like "s%" ) ,
Sport_name varchar(20),
Rollno bigint ,
Sname varchar(20),
Gender varchar(20)
);
```

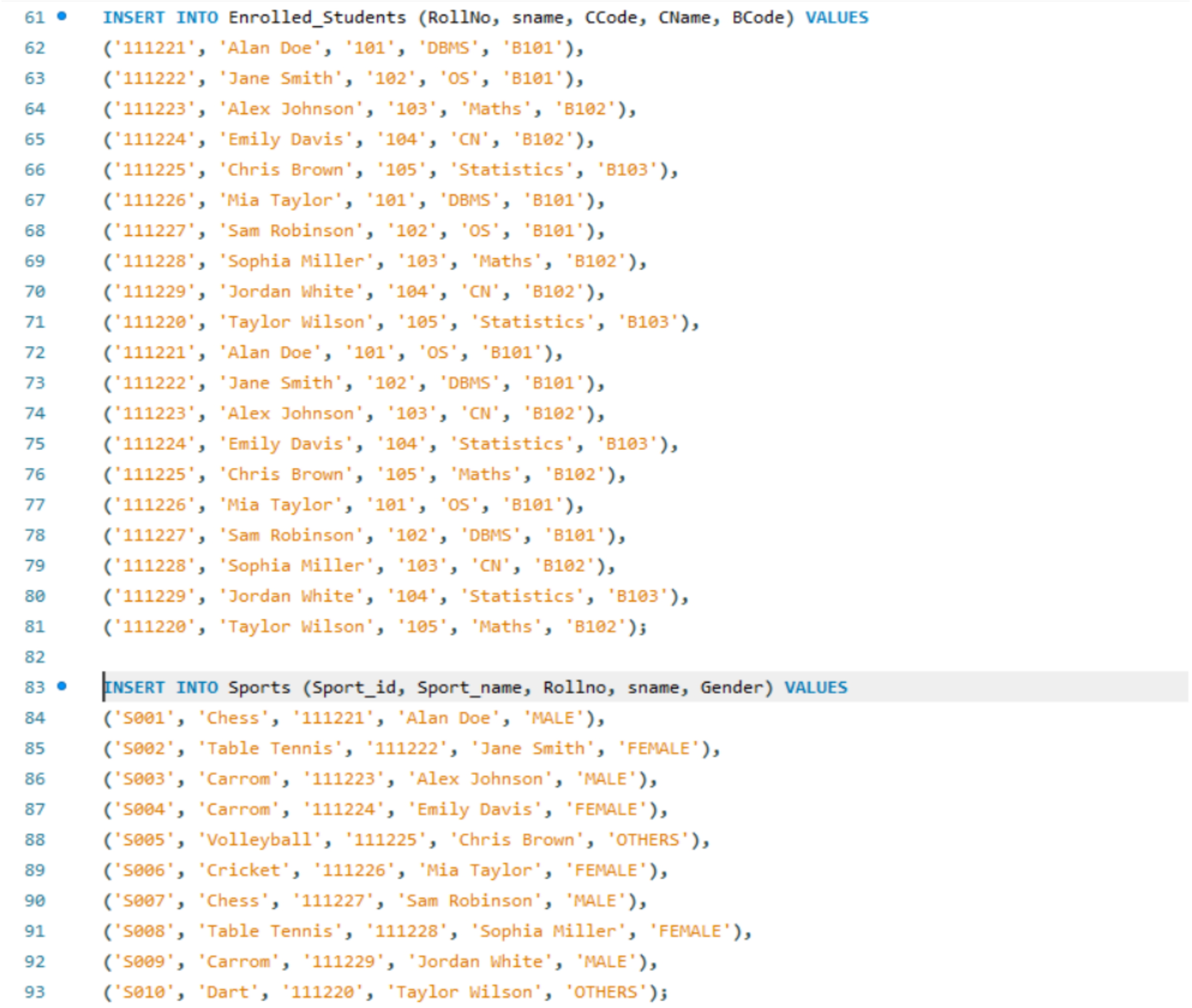
Inserted Data…



```sql
INSERT INTO Student (Rollno, SName, Dob, Gender, BCode) VALUES
(111221, 'Alan Doe', '2000-11-15', 'MALE', 'B101'),
(111222, 'Jane Smith', '2001-08-20', 'FEMALE', 'B102'),
(111223, 'Alex Johnson', '1999-12-10', 'MALE', 'B103'),
(111224, 'Emily Davis', '2002-11-25', 'FEMALE', 'B104'),
(111225, 'Chris Brown', '2000-07-05', 'OTHERS', 'B105'),
(111226, 'Mia Taylor', '2001-12-30', 'FEMALE', 'B106'),
(111227, 'Sam Robinson', '2005-09-12', 'MALE', 'B107'),
(111228, 'Sophia Miller', '2003-12-18', 'FEMALE', 'B108'),
(111229, 'Jordan White', '2002-11-08', 'MALE', 'B109'),
(111220, 'Taylor Wilson', '1999-06-22', 'OTHERS', 'B110');

INSERT INTO Branch (BCode, BName) VALUES
('B101', 'CSE '),
('B102', 'IT'),
('B103', 'EC');

INSERT INTO Course (CCode, CName, Credits, BCode) VALUES
('101', 'DBMS', 2, 'B101'),
('102', 'OS', 4, 'B101'),
('103', 'Maths', 3, 'B102'),
('104', 'CN', 1, 'B101'),
('105', 'Statistics', 3, 'B103');
```

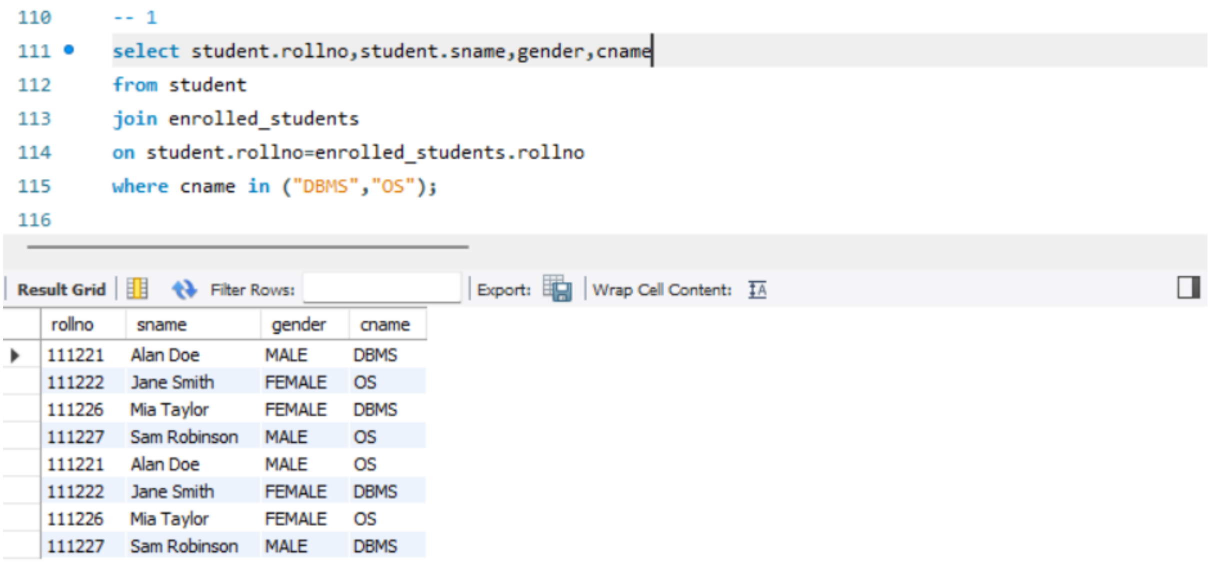


```sql
INSERT INTO Enrolled_Students (RollNo, sname, CCode, CName, BCode) VALUES
('111221', 'Alan Doe', '101', 'DBMS', 'B101'),
('111222', 'Jane Smith', '102', 'OS', 'B101'),
('111223', 'Alex Johnson', '103', 'Maths', 'B102'),
('111224', 'Emily Davis', '104', 'CN', 'B102'),
('111225', 'Chris Brown', '105', 'Statistics', 'B103'),
('111226', 'Mia Taylor', '101', 'DBMS', 'B101'),
('111227', 'Sam Robinson', '102', 'OS', 'B101'),
('111228', 'Sophia Miller', '103', 'Maths', 'B102'),
('111229', 'Jordan White', '104', 'CN', 'B102'),
('111220', 'Taylor Wilson', '105', 'Statistics', 'B103'),
('111221', 'Alan Doe', '101', 'OS', 'B101'),
('111222', 'Jane Smith', '102', 'DBMS', 'B101'),
('111223', 'Alex Johnson', '103', 'CN', 'B102'),
('111224', 'Emily Davis', '104', 'Statistics', 'B103'),
('111225', 'Chris Brown', '105', 'Maths', 'B102'),
('111226', 'Mia Taylor', '101', 'OS', 'B101'),
('111227', 'Sam Robinson', '102', 'DBMS', 'B101'),
('111228', 'Sophia Miller', '103', 'CN', 'B102'),
('111229', 'Jordan White', '104', 'Statistics', 'B103'),
('111220', 'Taylor Wilson', '105', 'Maths', 'B102');

INSERT INTO Sports (Sport_id, Sport_name, Rollno, sname, Gender) VALUES
('S001', 'Chess', '111221', 'Alan Doe', 'MALE'),
('S002', 'Table Tennis', '111222', 'Jane Smith', 'FEMALE'),
('S003', 'Carrom', '111223', 'Alex Johnson', 'MALE'),
('S004', 'Carrom', '111224', 'Emily Davis', 'FEMALE'),
('S005', 'Volleyball', '111225', 'Chris Brown', 'OTHERS'),
('S006', 'Cricket', '111226', 'Mia Taylor', 'FEMALE'),
('S007', 'Chess', '111227', 'Sam Robinson', 'MALE'),
('S008', 'Table Tennis', '111228', 'Sophia Miller', 'FEMALE'),
('S009', 'Carrom', '111229', 'Jordan White', 'MALE'),
('S010', 'Dart', '111220', 'Taylor Wilson', 'OTHERS');
```

Queries….

1. Write and execute an SQL query to list all the student’s Rollno, Name, and Gender

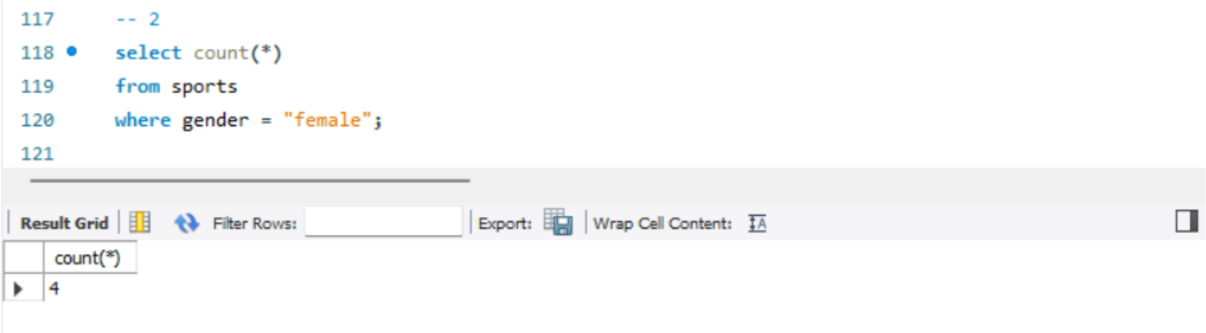


```
-- 1
select student.rollno,student.sname,gender,cname
from student
join enrolled_students
on student.rollno=enrolled_students.rollno
where cname in ("DBMS","OS");
```

rollno	sname	gender	cname
111221	Alan Doe	MALE	DBMS
111222	Jane Smith	FEMALE	OS
111226	Mia Taylor	FEMALE	DBMS
111227	Sam Robinson	MALE	OS
111221	Alan Doe	MALE	OS
111222	Jane Smith	FEMALE	DBMS
111226	Mia Taylor	FEMALE	OS
111227	Sam Robinson	MALE	DBMS

who are enrolled in course DBMS and OS.

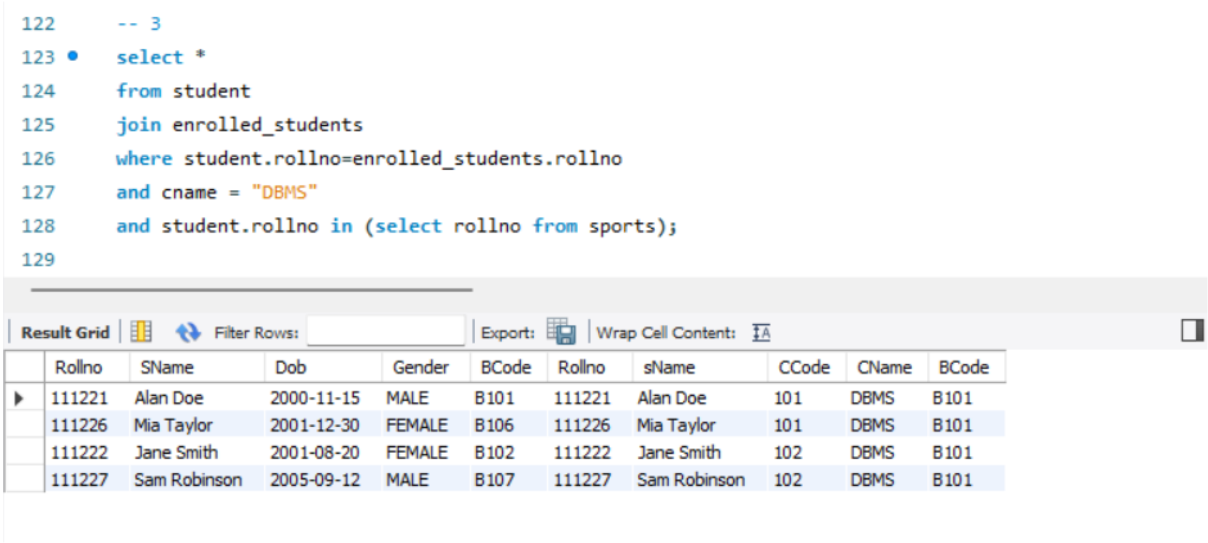
2. Write and execute an SQL query to list the number of female students in sports.



```
-- 2
select count(*)
from sports
where gender = "female";
```

count(*)
4

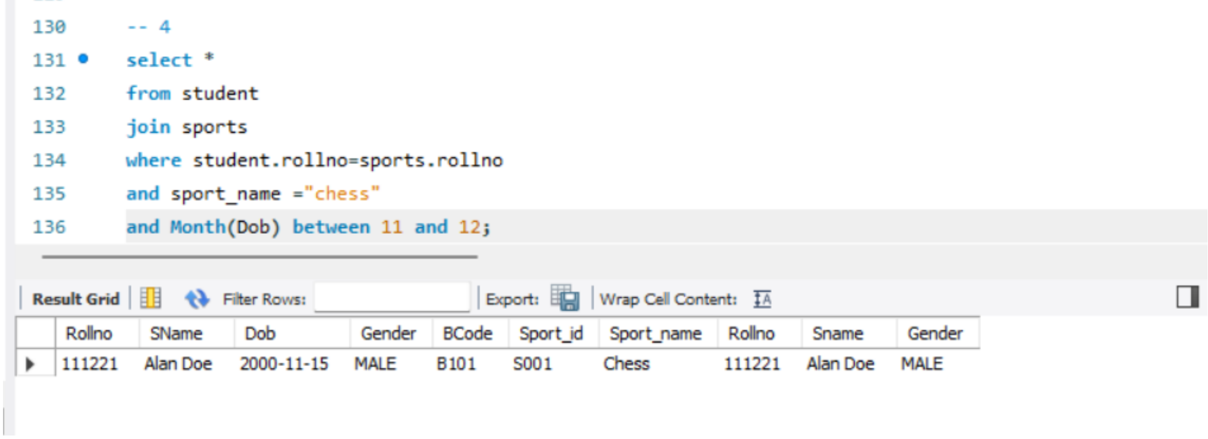
3. Write and execute an SQL query to list all the details of the students who are enrolled in the DBMS course and have also participated in some sports.



```sql
-- 3
select *
from student
join enrolled_students
where student.rollno=enrolled_students.rollno
and cname = "DBMS"
and student.rollno in (select rollno from sports);
```

Rollno	SName	Dob	Gender	BCode	Rollno	sName	CCode	CName	BCode
111221	Alan Doe	2000-11-15	MALE	B101	111221	Alan Doe	101	DBMS	B101
111226	Mia Taylor	2001-12-30	FEMALE	B106	111226	Mia Taylor	101	DBMS	B101
111222	Jane Smith	2001-08-20	FEMALE	B102	111222	Jane Smith	102	DBMS	B101
111227	Sam Robinson	2005-09-12	MALE	B107	111227	Sam Robinson	102	DBMS	B101

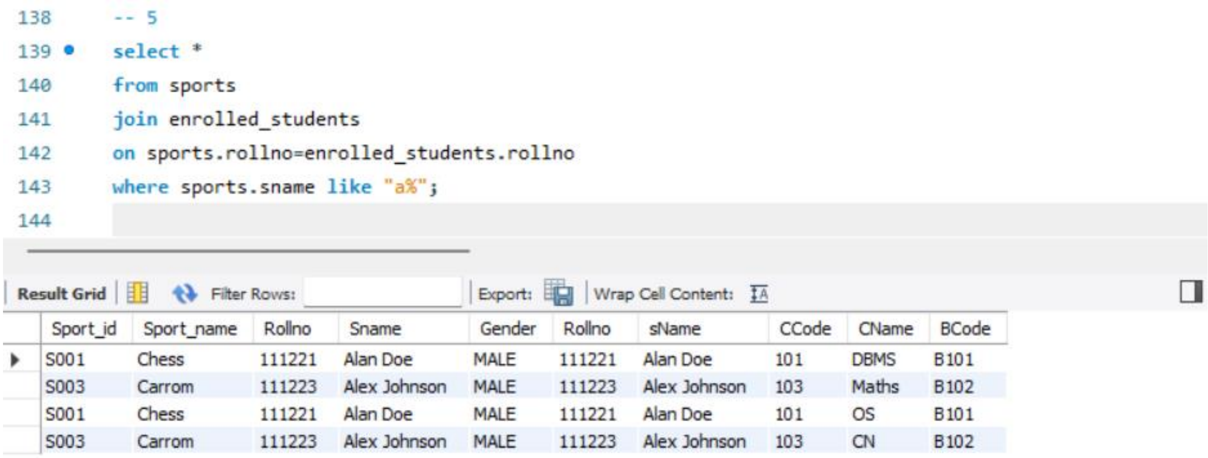
4. Write and execute a query to list details of the students who are playing chess and are born between month November and December.



```sql
-- 4
select *
from student
join sports
where student.rollno=sports.rollno
and sport_name ="chess"
and Month(Dob) between 11 and 12;
```

Rollno	SName	Dob	Gender	BCode	Sport_id	Sport_name	Rollno	Sname	Gender
111221	Alan Doe	2000-11-15	MALE	B101	S001	Chess	111221	Alan Doe	MALE

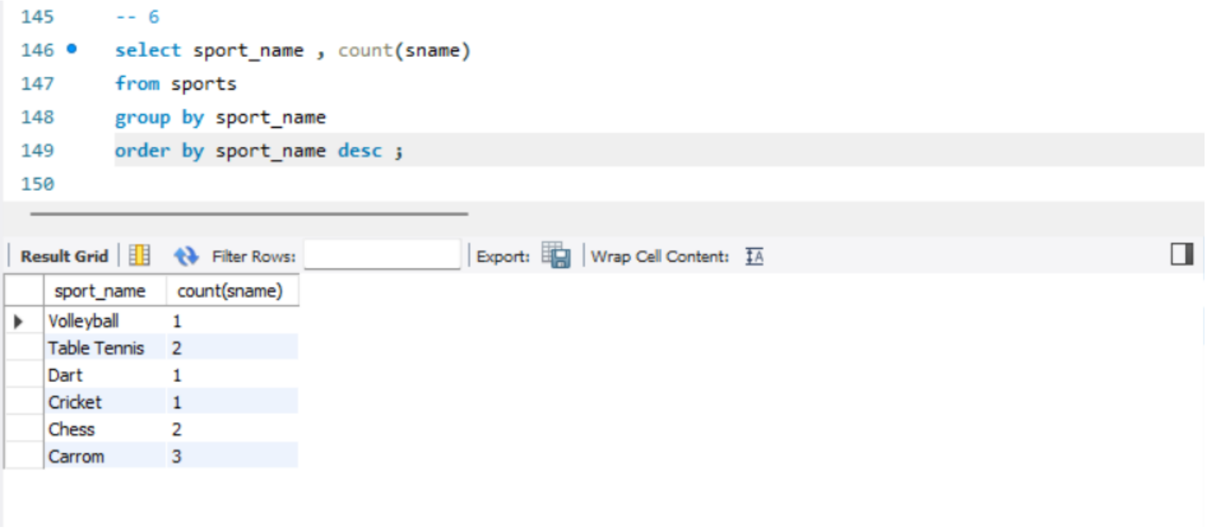
5. Write and execute a query that gives Courses details, Sports participation details by a student whose name starts with ‘A’.



```sql
-- 5
select *
from sports
join enrolled_students
on sports.rollno=enrolled_students.rollno
where sports.sname like "a%";
```

Sport_id	Sport_name	Rollno	Sname	Gender	Rollno	sName	CCode	CName	BCode
S001	Chess	111221	Alan Doe	MALE	111221	Alan Doe	101	DBMS	B101
S003	Carrom	111223	Alex Johnson	MALE	111223	Alex Johnson	103	Maths	B102
S001	Chess	111221	Alan Doe	MALE	111221	Alan Doe	101	OS	B101
S003	Carrom	111223	Alex Johnson	MALE	111223	Alex Johnson	103	CN	B102

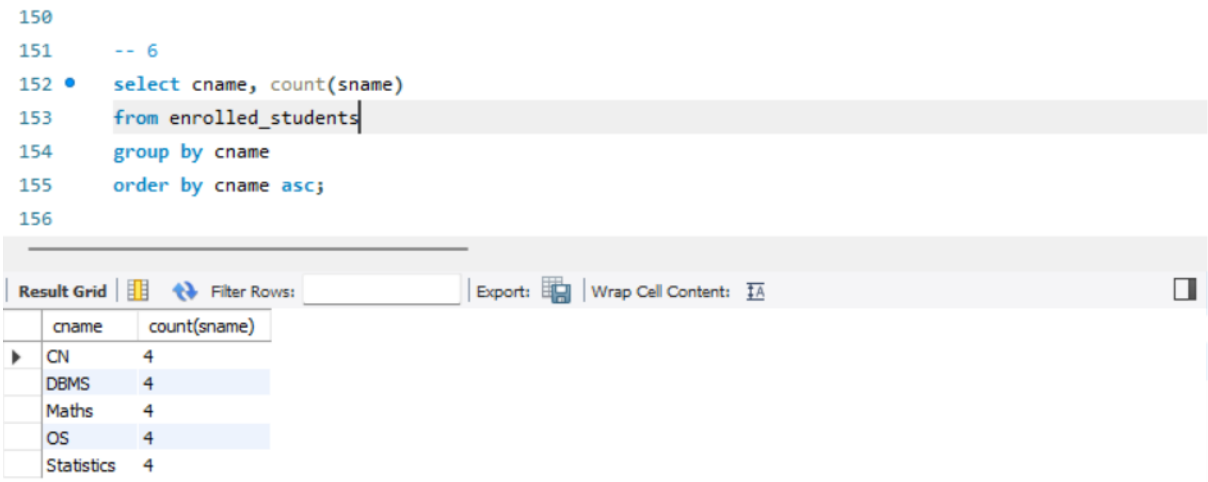
6. Give the count of the students in each sport in descending order with the name of the student, and the name of the sport.



```
-- 6
select sport_name , count(sname)
from sports
group by sport_name
order by sport_name desc ;
```

sport_name	count(sname)
Volleyball	1
Table Tennis	2
Dart	1
Cricket	1
Chess	2
Carrom	3

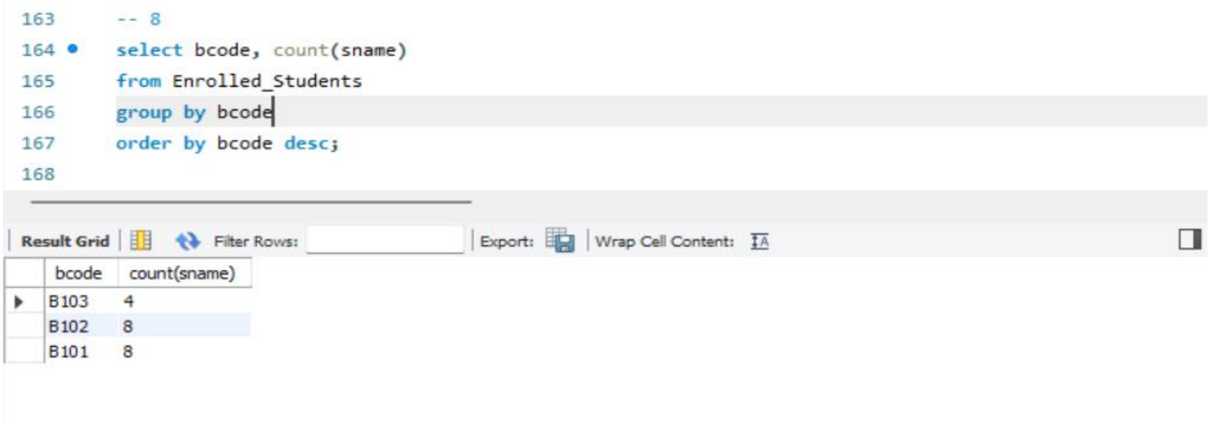
7. Give the count of students enrolled in each course and show it in ascending order.



```
-- 6
select cname, count(sname)
from enrolled_students
group by cname
order by cname asc;
```

cname	count(sname)
CN	4
DBMS	4
Maths	4
OS	4
Statistics	4

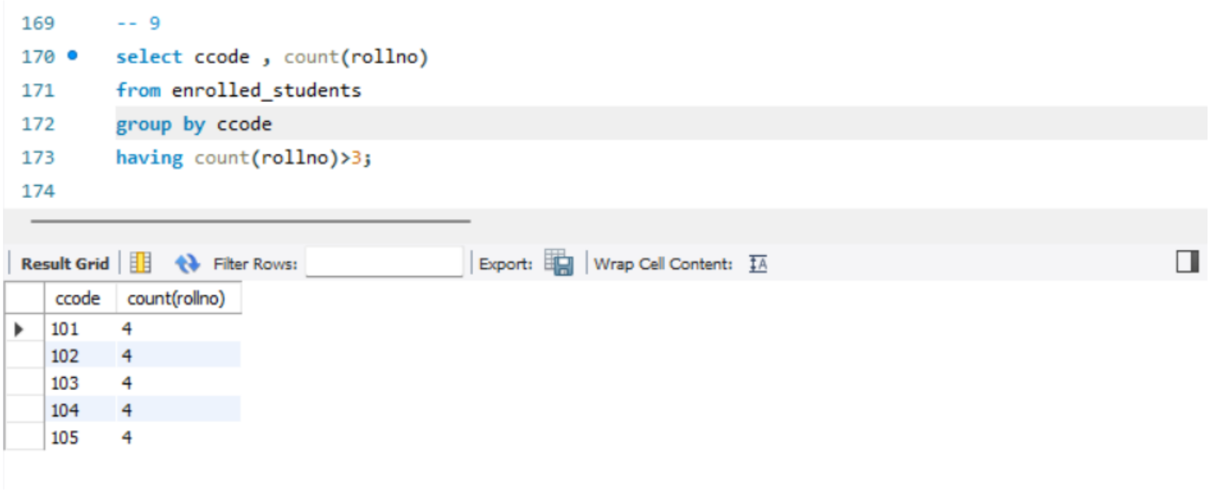
8. Give the count of students in each Branch and show it in descending order.



```
-- 8
select bcode, count(sname)
from Enrolled_Students
group by bcode
order by bcode desc;
```

bcode	count(sname)
B103	4
B102	8
B101	8

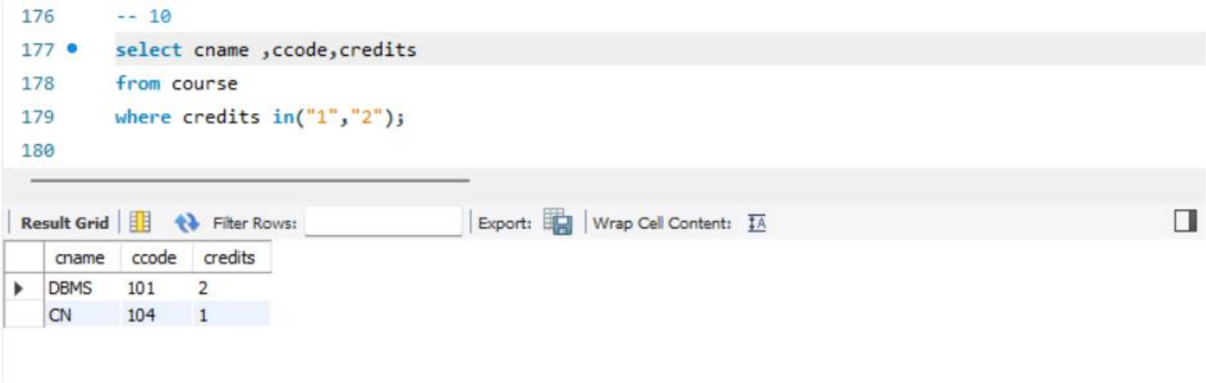
9. Write and execute an SQL query that gives the name and count of courses in which more than 3 students are enrolled.



```
-- 9
select ccode , count(rollno)
from enrolled_students
group by ccode
having count(rollno)>3;
```

ccode	count(rollno)
101	4
102	4
103	4
104	4
105	4

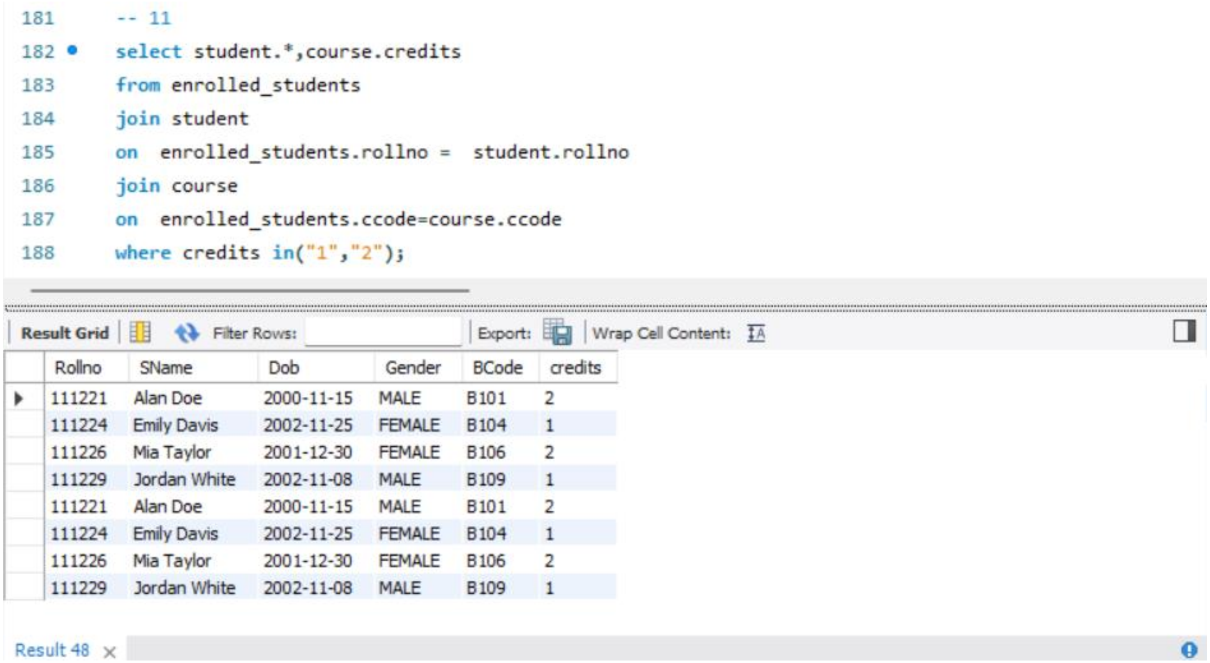
10. Write and execute an SQL query that gives the name and course code of courses whose credits are either 2 or 1.



```
-- 10
select cname ,ccode,credits
from course
where credits in("1","2");
```

cname	ccode	credits
DBMS	101	2
CN	104	1

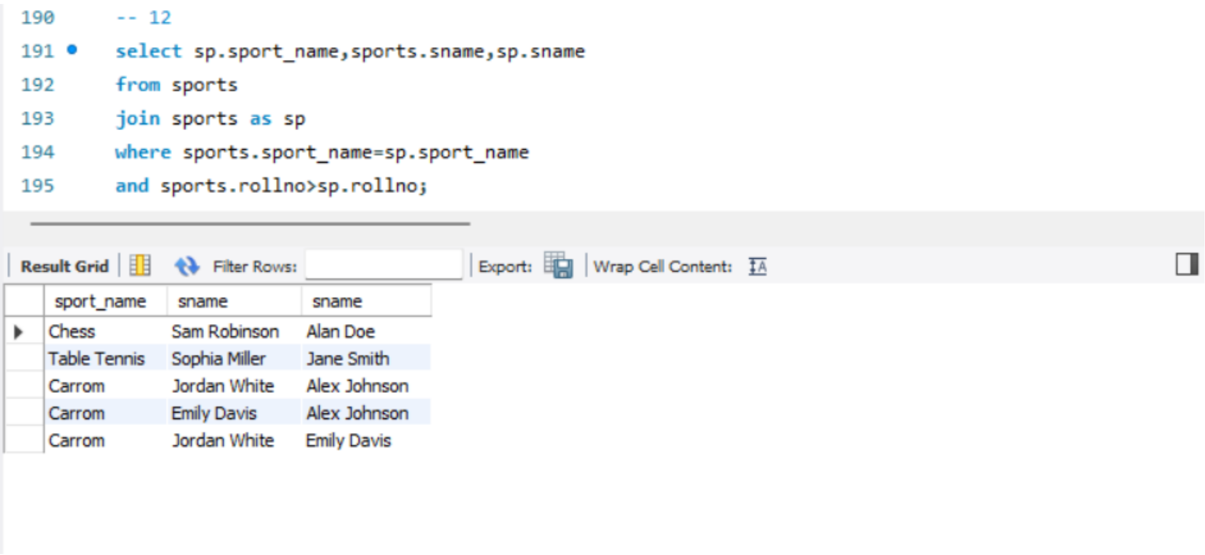
11. Write and execute an SQL query that gives details of students who are enrolled in courses whose credits are either 2 or 1.



```sql
-- 11
select student.*,course.credits
from enrolled_students
join student
on  enrolled_students.rollno =  student.rollno
join course
on  enrolled_students.ccode=course.ccode
where credits in("1","2");
```

Rollno	SName	Dob	Gender	BCode	credits
111221	Alan Doe	2000-11-15	MALE	B101	2
111224	Emily Davis	2002-11-25	FEMALE	B104	1
111226	Mia Taylor	2001-12-30	FEMALE	B106	2
111229	Jordan White	2002-11-08	MALE	B109	1
111221	Alan Doe	2000-11-15	MALE	B101	2
111224	Emily Davis	2002-11-25	FEMALE	B104	1
111226	Mia Taylor	2001-12-30	FEMALE	B106	2
111229	Jordan White	2002-11-08	MALE	B109	1

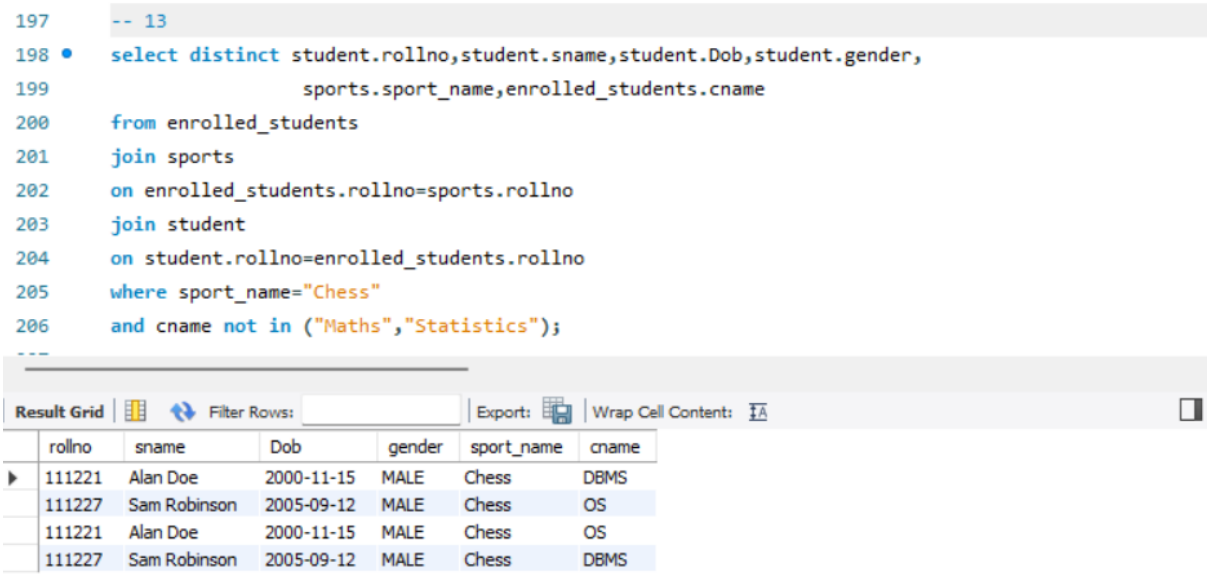
12. Write and execute an SQL query to list name of the students who are playing same Sports.



```sql
-- 12
select sp.sport_name,sports.sname,sp.sname
from sports
join sports as sp
where sports.sport_name=sp.sport_name
and sports.rollno>sp.rollno;
```

sport_name	sname	sname
Chess	Sam Robinson	Alan Doe
Table Tennis	Sophia Miller	Jane Smith
Carrom	Jordan White	Alex Johnson
Carrom	Emily Davis	Alex Johnson
Carrom	Jordan White	Emily Davis

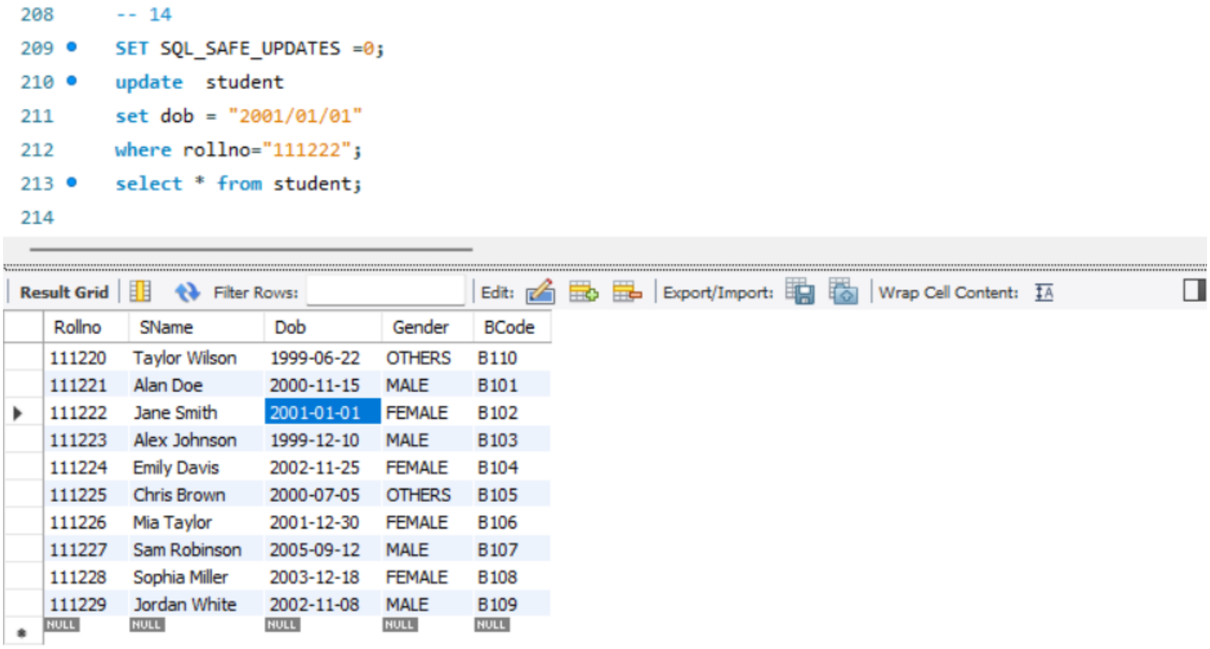
13. Write and execute an SQL query that shows details of students who are playing chess and are not enrolled for Maths and Statistics.



```
-- 13
select distinct student.rollno,student.sname,student.Dob,student.gender,
                 sports.sport_name,enrolled_students.cname
from enrolled_students
join sports
on enrolled_students.rollno=sports.rollno
join student
on student.rollno=enrolled_students.rollno
where sport_name="Chess"
and cname not in ("Maths","Statistics");
```

rollno	sname	Dob	gender	sport_name	cname
111221	Alan Doe	2000-11-15	MALE	Chess	DBMS
111227	Sam Robinson	2005-09-12	MALE	Chess	OS
111221	Alan Doe	2000-11-15	MALE	Chess	OS
111227	Sam Robinson	2005-09-12	MALE	Chess	DBMS

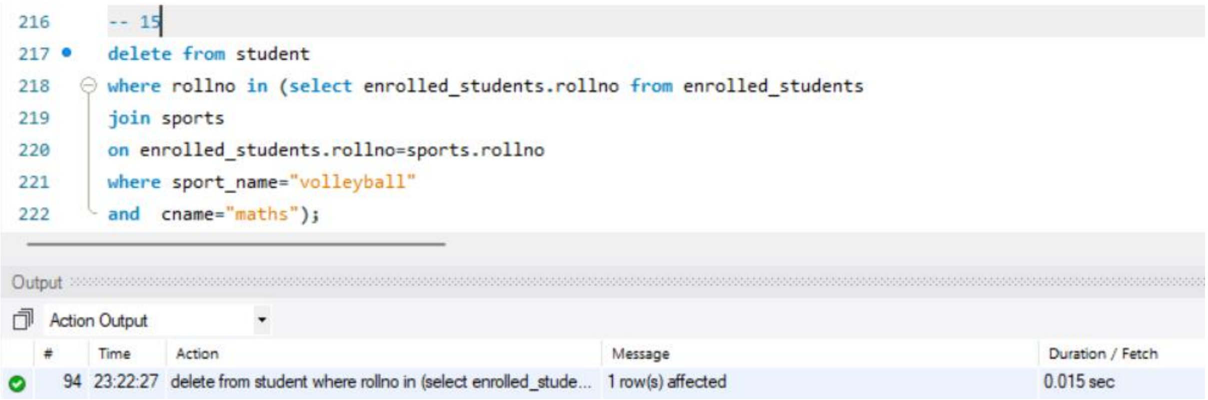
14. Write and execute an SQL query to update a student detail whose rollno is 111222 and show that rollno 111222 exists in the student table.



```
-- 14
SET SQL_SAFE_UPDATES =0;
update  student
set dob = "2001/01/01"
where rollno="111222";
select * from student;
```

Rollno	SName	Dob	Gender	BCode
111220	Taylor Wilson	1999-06-22	OTHERS	B110
111221	Alan Doe	2000-11-15	MALE	B101
111222	Jane Smith	2001-01-01	FEMALE	B102
111223	Alex Johnson	1999-12-10	MALE	B103
111224	Emily Davis	2002-11-25	FEMALE	B104
111225	Chris Brown	2000-07-05	OTHERS	B105
111226	Mia Taylor	2001-12-30	FEMALE	B106
111227	Sam Robinson	2005-09-12	MALE	B107
111228	Sophia Miller	2003-12-18	FEMALE	B108
111229	Jordan White	2002-11-08	MALE	B109
NULL	NULL	NULL	NULL	NULL

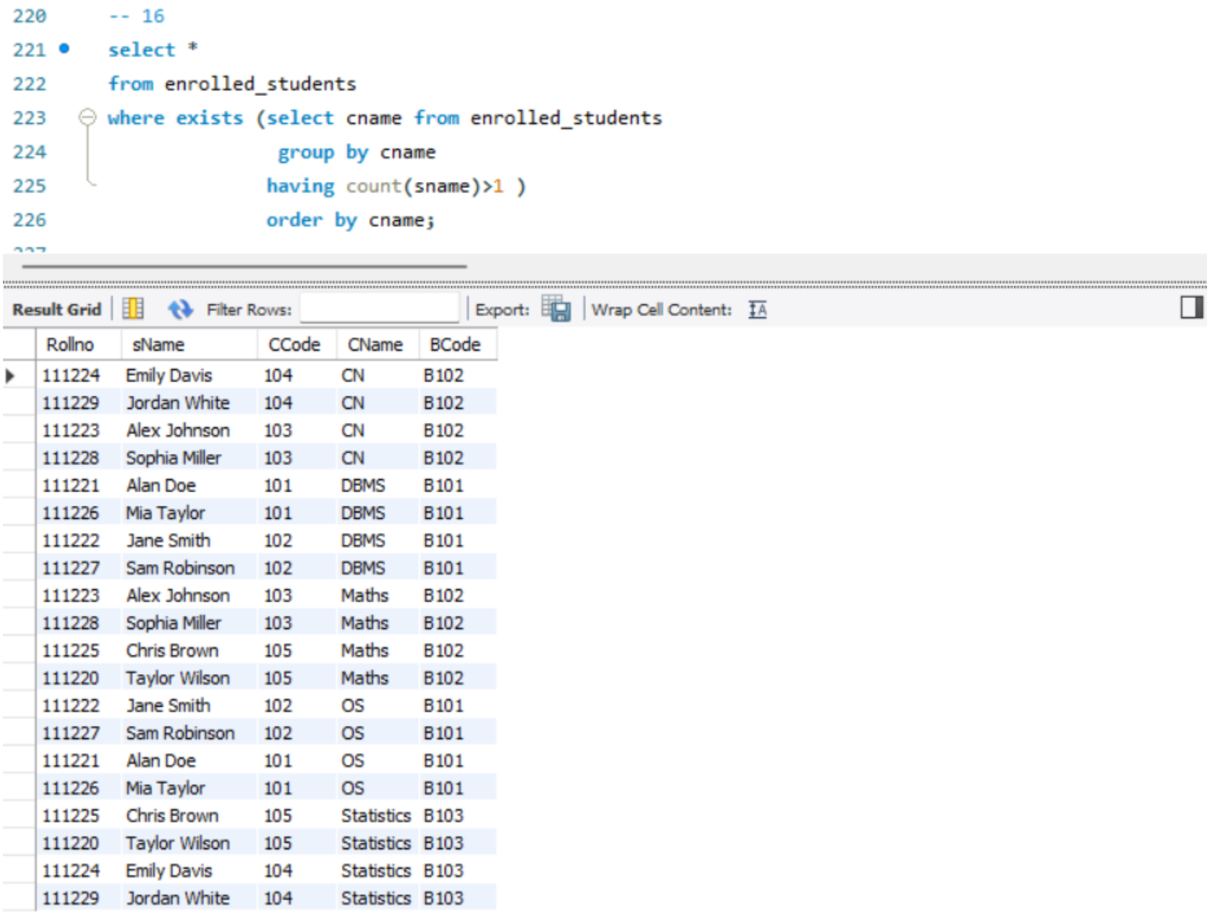
15. Write and execute an SQL query to Delete the details of students who have enrolled in course Maths and are playing volleyball from enrolled Students table.



```
-- 15
delete from student
where rollno in (select enrolled_students.rollno from enrolled_students
join sports
on enrolled_students.rollno=sports.rollno
where sport_name="volleyball"
and  cname="maths");
```

#	Time	Action	Message	Duration / Fetch
94	23:22:27	delete from student where rollno in (select enrolled_stude...	1 row(s) affected	0.015 sec

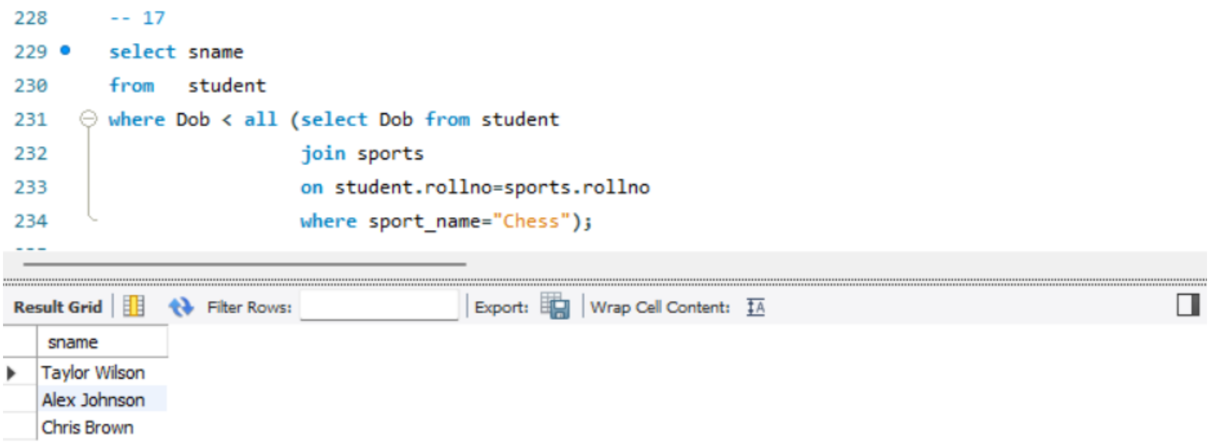
16. Write a query to display duplicates in CName column in Enrolled Students table using Exists operator.



```
-- 16
select *
from enrolled_students
where exists (select cname from enrolled_students
                 group by cname
                having count(sname)>1 )
                order by cname;
```

Rollno	sName	CCode	CName	BCode
111224	Emily Davis	104	CN	B102
111229	Jordan White	104	CN	B102
111223	Alex Johnson	103	CN	B102
111228	Sophia Miller	103	CN	B102
111221	Alan Doe	101	DBMS	B101
111226	Mia Taylor	101	DBMS	B101
111222	Jane Smith	102	DBMS	B101
111227	Sam Robinson	102	DBMS	B101
111223	Alex Johnson	103	Maths	B102
111228	Sophia Miller	103	Maths	B102
111225	Chris Brown	105	Maths	B102
111220	Taylor Wilson	105	Maths	B102
111222	Jane Smith	102	OS	B101
111227	Sam Robinson	102	OS	B101
111221	Alan Doe	101	OS	B101
111226	Mia Taylor	101	OS	B101
111225	Chris Brown	105	Statistics	B103
111220	Taylor Wilson	105	Statistics	B103
111224	Emily Davis	104	Statistics	B103
111229	Jordan White	104	Statistics	B103

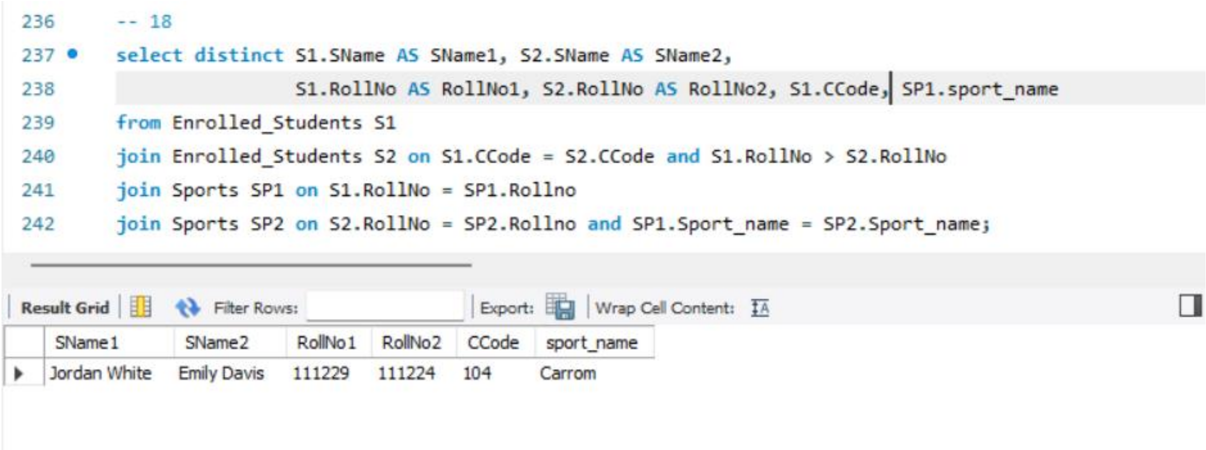
17. Write a query to get the name of students whose age is greater than all the students in playing 'Chess'.



```
-- 17
select sname
from   student
where Dob < all (select Dob from student
                 join sports
                 on student.rollno=sports.rollno
                 where sport_name="Chess");
```

sname
Taylor Wilson
Alex Johnson
Chris Brown

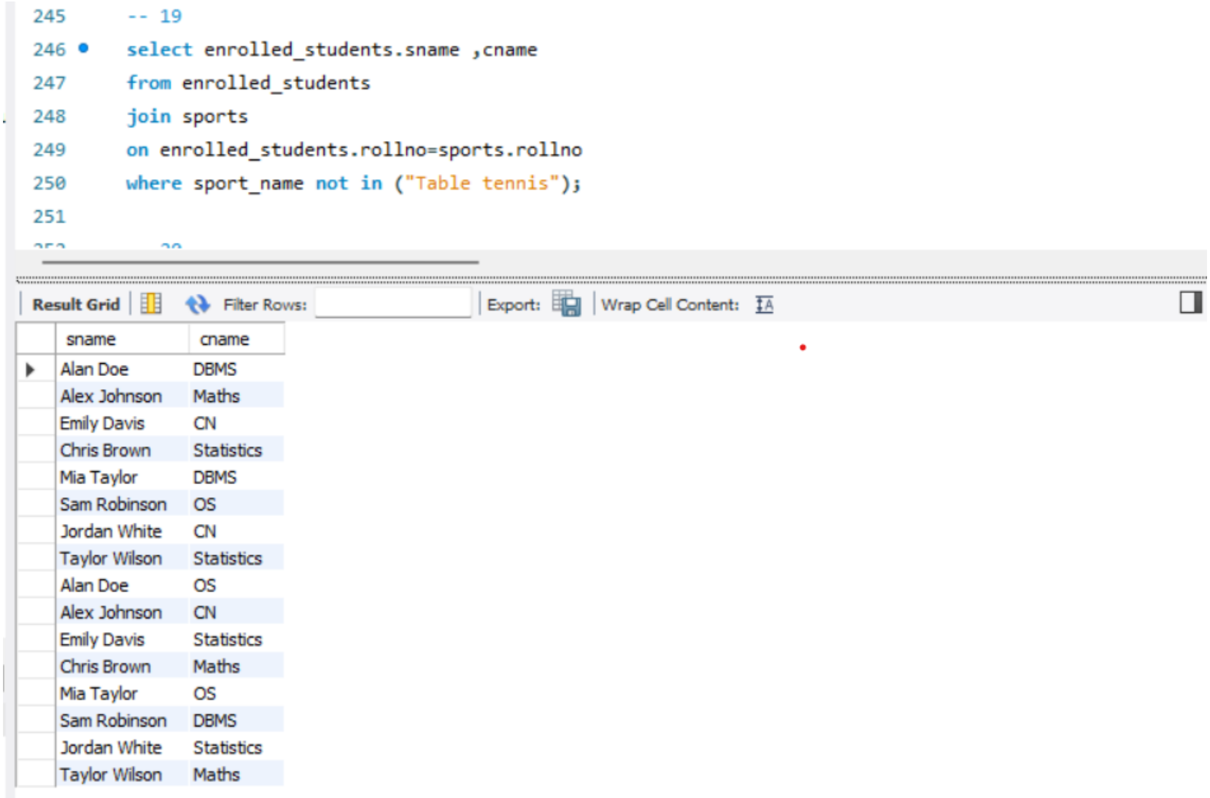
18. Write a query to get the names of students who are playing the same sport and are enrolled in the same courses.



```
-- 18
select distinct S1.SName AS SName1, S2.SName AS SName2,
                S1.RollNo AS RollNo1, S2.RollNo AS RollNo2, S1.CCode, SP1.sport_name
from Enrolled_Students S1
join Enrolled_Students S2 on S1.CCode = S2.CCode and S1.RollNo > S2.RollNo
join Sports SP1 on S1.RollNo = SP1.Rollno
join Sports SP2 on S2.RollNo = SP2.Rollno and SP1.Sport_name = SP2.Sport_name;
```

SName1	SName2	RollNo1	RollNo2	CCode	sport_name
Jordan White	Emily Davis	111229	111224	104	Carrom

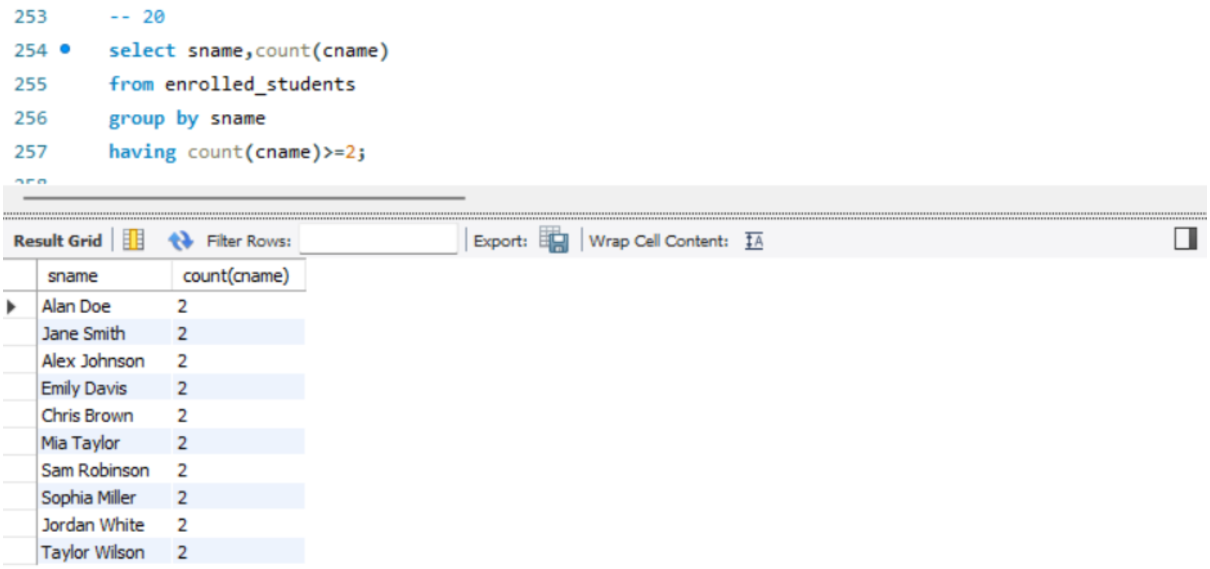
19. Write a query to get the course name where no student is playing 'Table Tennis'.



```sql
-- 19
select enrolled_students.sname ,cname
from enrolled_students
join sports
on enrolled_students.rollno=sports.rollno
where sport_name not in ("Table tennis");
```

sname	cname
Alan Doe	DBMS
Alex Johnson	Maths
Emily Davis	CN
Chris Brown	Statistics
Mia Taylor	DBMS
Sam Robinson	OS
Jordan White	CN
Taylor Wilson	Statistics
Alan Doe	OS
Alex Johnson	CN
Emily Davis	Statistics
Chris Brown	Maths
Mia Taylor	OS
Sam Robinson	DBMS
Jordan White	Statistics
Taylor Wilson	Maths

20. Write a query to get the names of students who are enrolled in more than 2 courses.



```sql
-- 20
select sname,count(cname)
from enrolled_students
group by sname
having count(cname)>=2;
```

sname	count(cname)
Alan Doe	2
Jane Smith	2
Alex Johnson	2
Emily Davis	2
Chris Brown	2
Mia Taylor	2
Sam Robinson	2
Sophia Miller	2
Jordan White	2
Taylor Wilson	2

Question 2

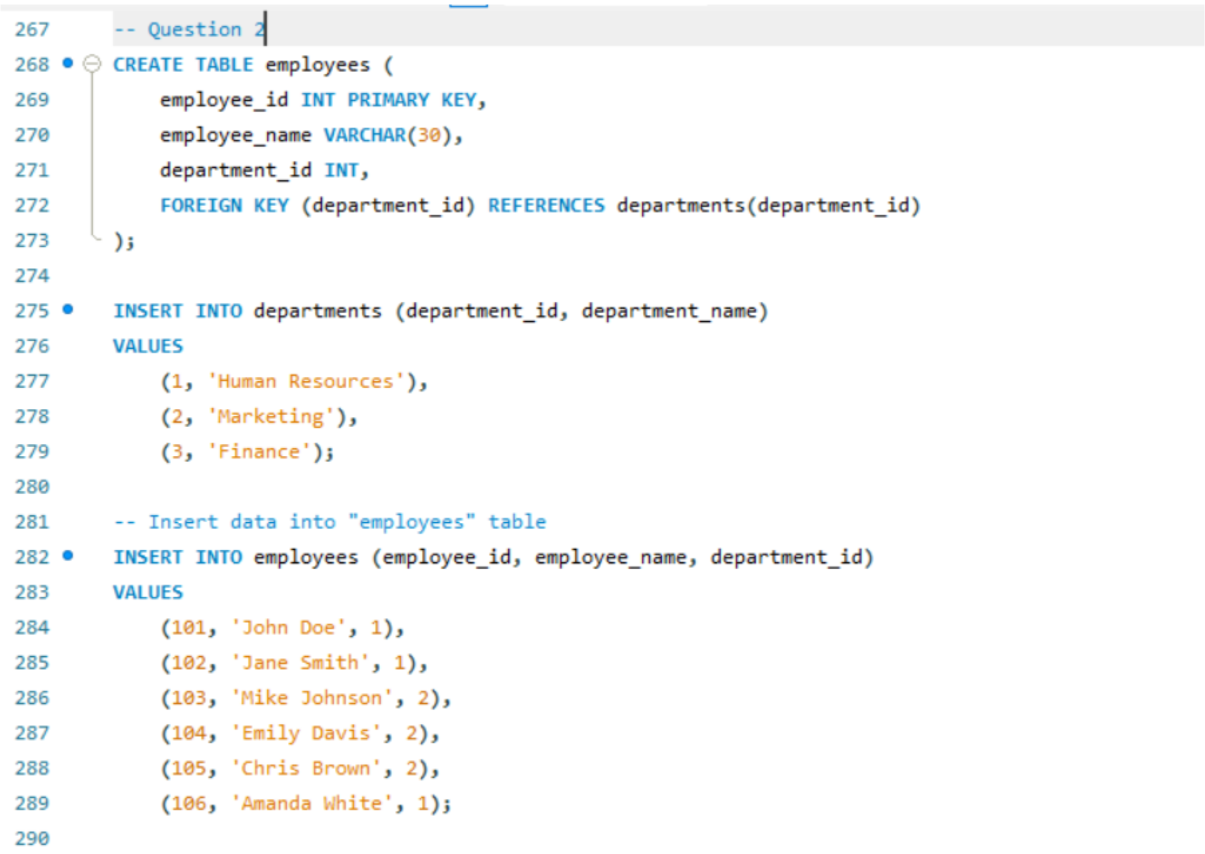
Consider two tables: employees and departments. The employees table contains information about employees including their ID, name, and

department ID. The departments table contains information about departments including their ID and name.

Write a query to find all departments that have at least one employee.

Use the EXISTS keyword to accomplish this task.

Creation and insertion ……

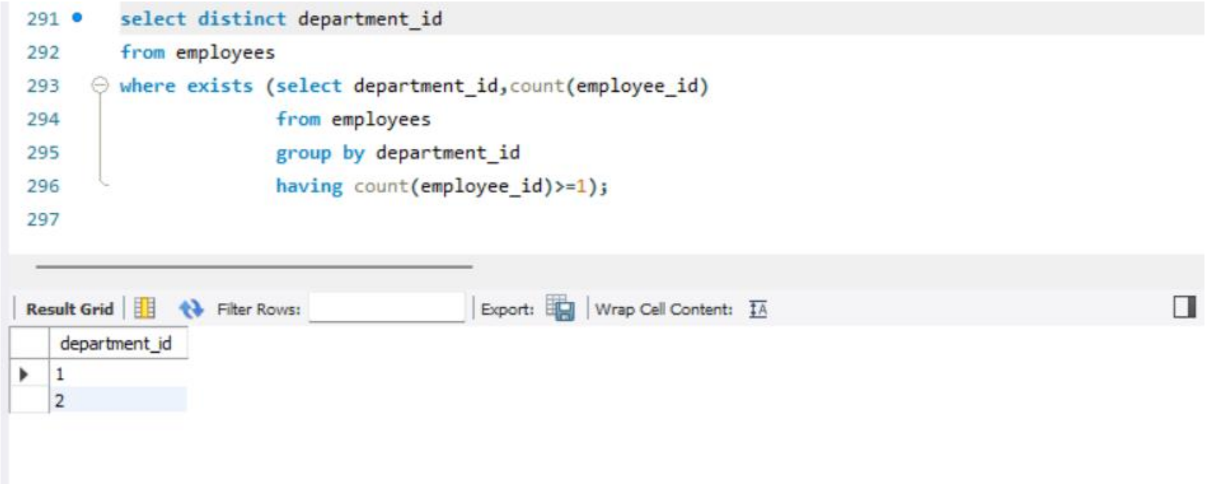


```sql
-- Question 2
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(30),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);

INSERT INTO departments (department_id, department_name)
VALUES
    (1, 'Human Resources'),
    (2, 'Marketing'),
    (3, 'Finance');

-- Insert data into "employees" table
INSERT INTO employees (employee_id, employee_name, department_id)
VALUES
    (101, 'John Doe', 1),
    (102, 'Jane Smith', 1),
    (103, 'Mike Johnson', 2),
    (104, 'Emily Davis', 2),
    (105, 'Chris Brown', 2),
    (106, 'Amanda White', 1);
```

Query…



```sql
select distinct department_id
from employees
where exists (select department_id,count(employee_id)
              from employees
              group by department_id
              having count(employee_id)>=1);
```

department_id
1
2

- - - [END] - - -