LAB QUESTIONS

SQL

CEHTAN

Q1. Create a table named students with fields:

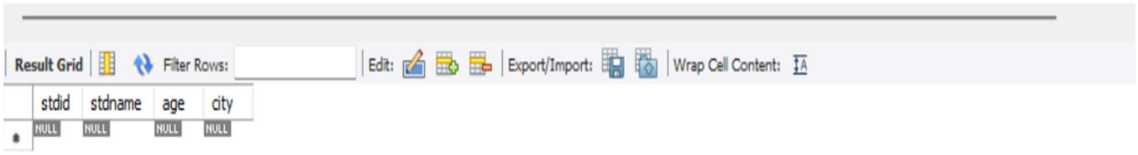
- stdid INT PRIMARY KEY
- stdname VARCHAR(50)
- age INT
- city VARCHAR(50)

Query:

```
CREATE TABLE students (
  stdid int primary key,
  stdname varchar(50),
  age int,
  city varchar(50)
  );
```

```sql
create database Collage;
use Collage;
CREATE TABLE students (
    stdid INT PRIMARY KEY,
    stdname VARCHAR(50),
    age INT,
    city VARCHAR(50)

);
select * from students;
```



Q2. Insert the following records into the students table:

stdid	stdname	age	city
1	Rohan	20	Pune
2	Meera	22	Mumbai
3	Arjun	21	Delhi
4	Kavya	23	Pune
5	Neha	22	Kolkata

Query:

INSERT INTO students (stdid, stdname, age, city) VALUES

(1, 'Rohan', 20, 'Pune'),

(2, 'Meera', 22, 'Mumbai'),

(3, 'Arjun', 21, 'Delhi'),

(4, 'Kavya', 23, 'Pune'),

(5, 'Neha', 22, 'Kolkata');

stdid	stdname	age	city
1	Rohan	20	Pune
2	Meera	22	Mumbai
3	Arjun	21	Delhi
4	Kavya	23	Pune
5	Neha	22	Kolkata

Q3. Display all student records.

Query:

select * from students

Result:

stdid	stdname	age	city
1	Rohan	20	Pune
2	Meera	22	Mumbai
3	Arjun	21	Delhi
4	Kavya	23	Pune
5	Neha	22	Kolkata

Q4. Display only the name and age of all students.

Query:

select stdname as Name, age as Age from students

Result:

Name	Age
Rohan	20
Meera	22
Arjun	21
Kavya	23
Neha	22

Q5. Display students who are from Pune.

Query:

```
select *
from students
where city == 'Pune'
```

Result:

stdid	stdname	age	city
1	Rohan	20	Pune
4	Kavya	23	Pune

Q6. Display students whose age is greater than 21.

Query:

```
select *
from students
where age>21
```

Result:

stdid	stdname	age	city
2	Meera	22	Mumbai
4	Kavya	23	Pune
5	Neha	22	Kolkata

Q7. Display students in descending order of age.

Query:

```
select *
from students
order by age desc
```

Result:

stdid	stdname	age	city
4	Kavya	23	Pune
2	Meera	22	Mumbai
5	Neha	22	Kolkata
3	Arjun	21	Delhi
1	Rohan	20	Pune

Q8. Count how many students belong to each city. (Use GROUP BY)

Query:

```
select city, count(stdid)
from students
group by city
```

Result:

city	count(stdid)
Delhi	1
Kolkata	1
Mumbai	1
Pune	2

Q9. Display students whose name starts with ‘K’. (Use LIKE)

Query:

```
SELECT *
FROM students
WHERE stdname LIKE 'K%';
```

Result:

stdid	stdname	age	city
4	Kavya	23	Pune

Q10. Delete student whose stdid = 5.

Query:

```
DELETE FROM students
WHERE stdid = 5;
```

Result:

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

Q11. Add a new column contact VARCHAR(15) to the students table.

Query:

```
ALTER TABLE students
ADD COLUMN contact VARCHAR(15);
```

Result:

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

Q12. Modify the data type of city column to VARCHAR(100).

Query:

```
ALTER TABLE students
MODIFY COLUMN city VARCHAR(100);
```

Q13. Rename the column stdname to student_name.

Query:

```
ALTER TABLE students
CHANGE COLUMN stdname student_name VARCHAR(255);
```

Q14. Drop the column contact from the table.

Query:

```
ALTER TABLE students
DROP COLUMN contact;
```

Result:

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

Q15. Add a new column gender ENUM('M','F').

Query:

```
ALTER TABLE students
ADD gender ENUM('M', 'F');
```

Q16. Display student name and marks of only those students who have matching IDs in both tables.

Query:

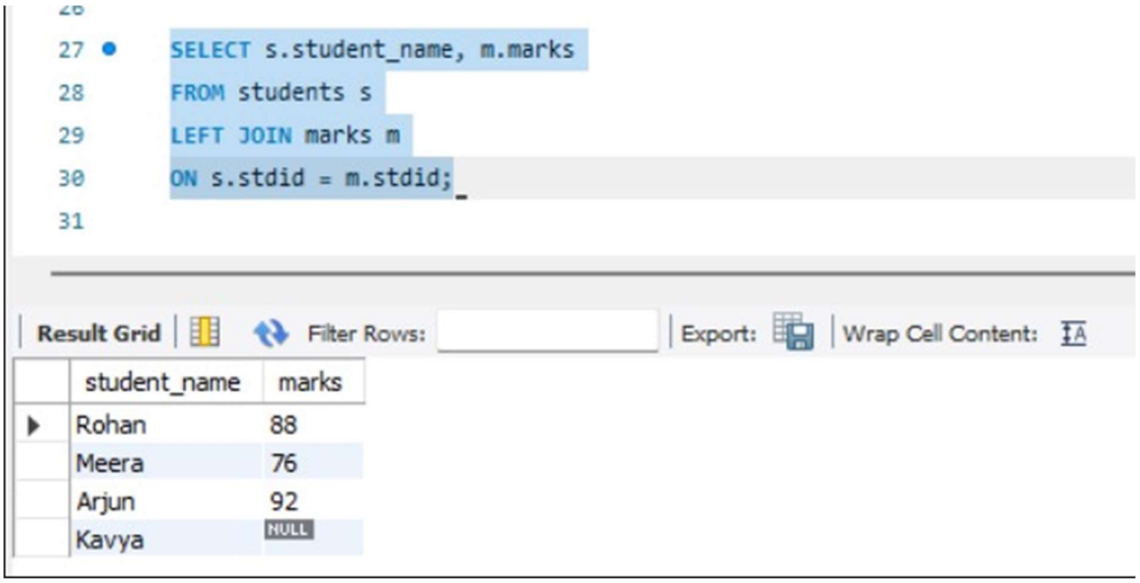
SELECT s.student_name, m.marks

FROM students s

INNER JOIN marks m

ON s.stdid = m.stdid;

Result:

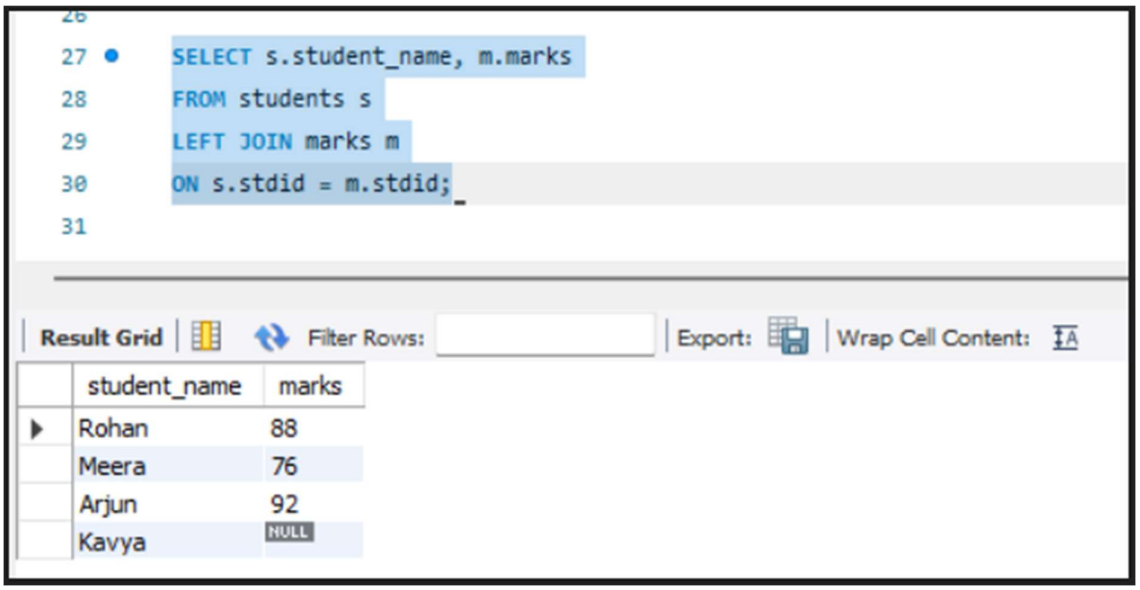


Q17. Display all students and their marks.

Query:

```
SELECT s.student_name, m.marks
FROM students s
LEFT JOIN marks m
ON s.stdid = m.stdid;
```

Result:

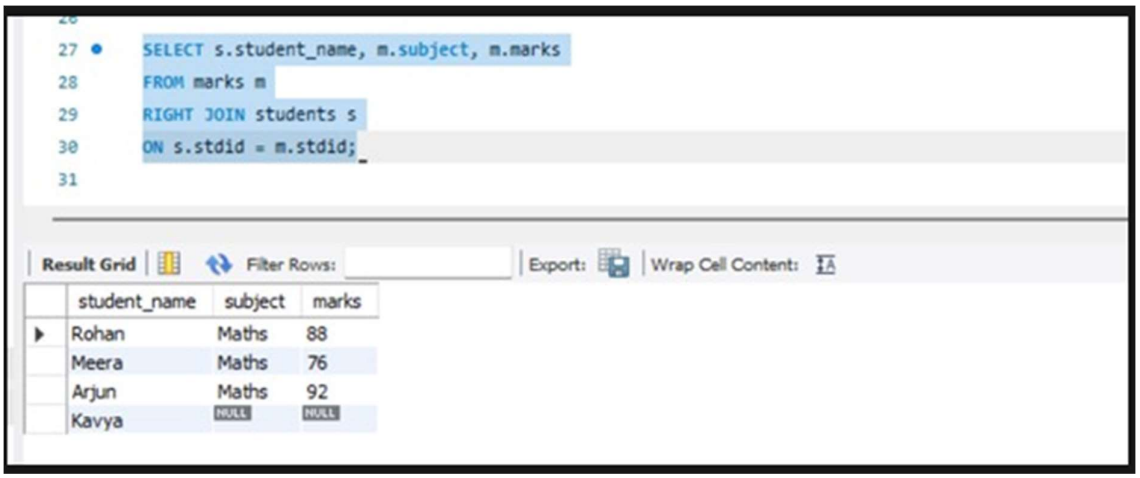


Q18. Display all marks records along with student names.

Query:

```
SELECT s.student_name, m.subject, m.marks
FROM marks m
RIGHT JOIN students s
ON s.stdid = m.stdid;
```

Result:

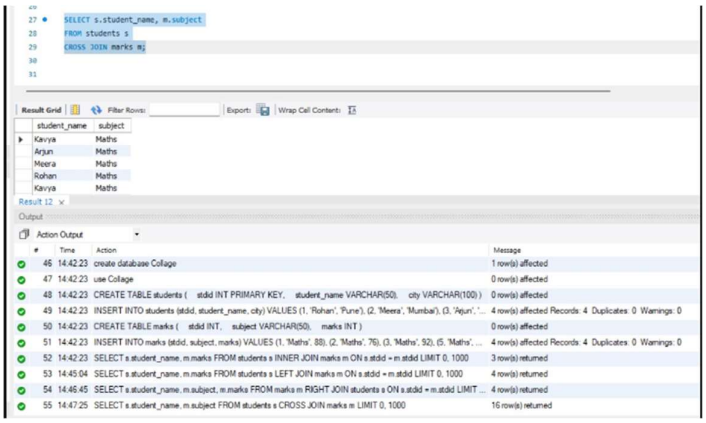


Q19. Display all possible combinations of students and subjects.

Query:

```
SELECT s.student_name, m.subject
FROM students s
CROSS JOIN marks m;
```

Result:



student_name	subject
Kavya	Maths
Arjun	Maths
Meera	Maths
Rohan	Maths
Kavya	Maths

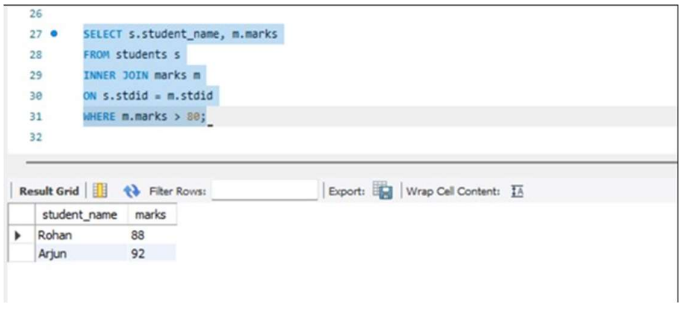
Q20. Using INNER JOIN, display students who scored more than 80.

Result:

Query:

```
SELECT s.student_name, m.marks
FROM students s
INNER JOIN marks m
ON s.stdid = m.stdid
WHERE m.marks > 80;
```

Result:



student_name	marks
Rohan	88
Arjun	92