WHAT IS SQL AND MYSQL?

SQL (Structured Query Language)

SQL is a standard programming language used to **manage and manipulate databases**. It is used to:

•Create, update, delete, and retrieve data from databases.

•Define and modify database structures (tables, indexes, etc.).

•Manage database security and transactions.

Examples of SQL commands:

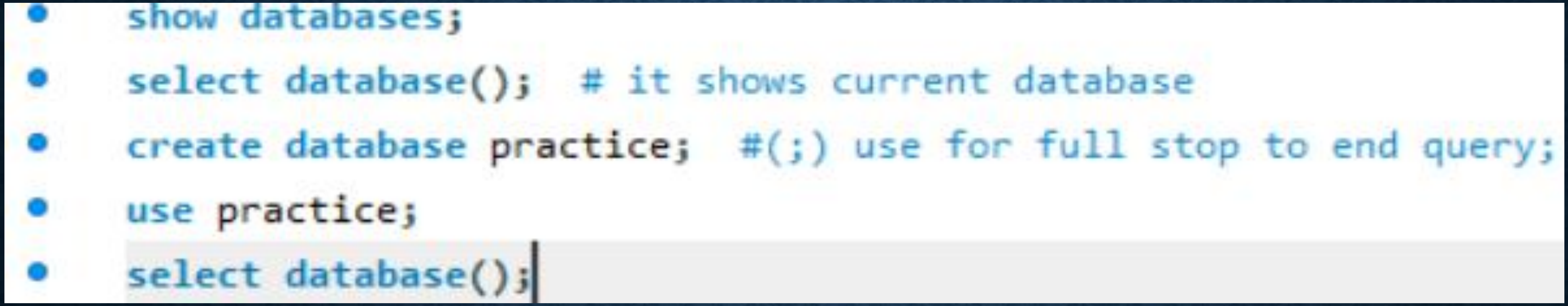
•SELECT * FROM users; → Retrieves all data from the "users" table.

•INSERT INTO users (name, email) VALUES ('John', 'john@example.com'); → Adds a new record.

MySQL

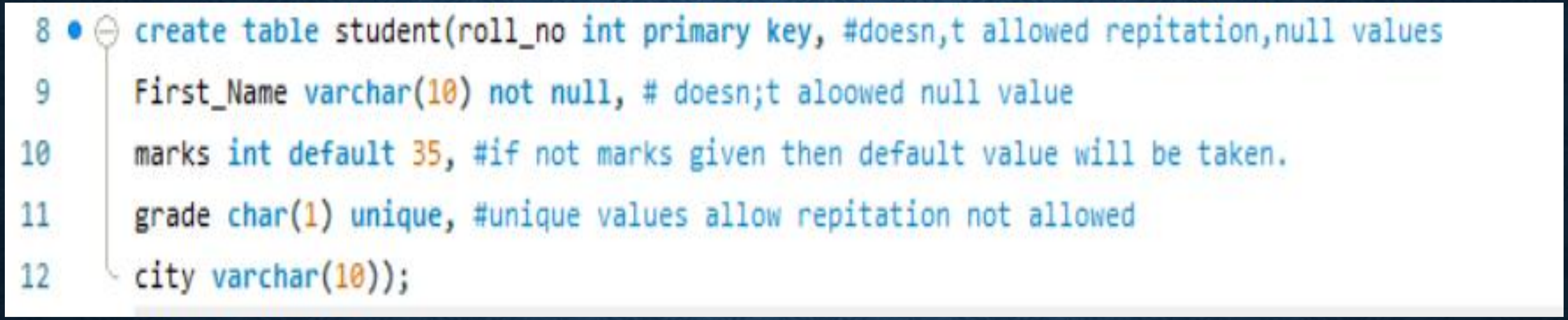
- MySQL is a **relational database management system (RDBMS)** that uses SQL to store, retrieve, and manage data. It is open-source and widely used for web applications, including **WordPress, e-commerce platforms, and enterprise applications**.
- **Key Features of MySQL:**
- **Fast & Scalable:** Can handle large amounts of data efficiently.
- **Open-Source:** Free to use with a strong community.
- **Cross-Platform:** Works on Windows, Linux, and macOS.

1.HOW TO CREATE DATABASE?



```
show databases;
select database();  # it shows current database
create database practice;  #(;) use for full stop to end query;
use practice;
select database();
```

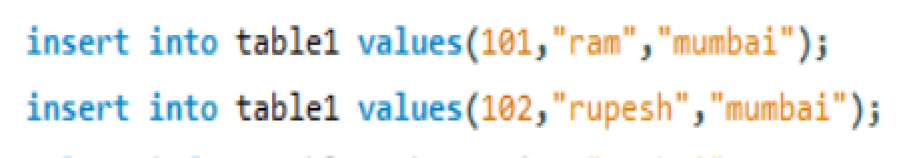
2. HOW TO CREATE A TABLE



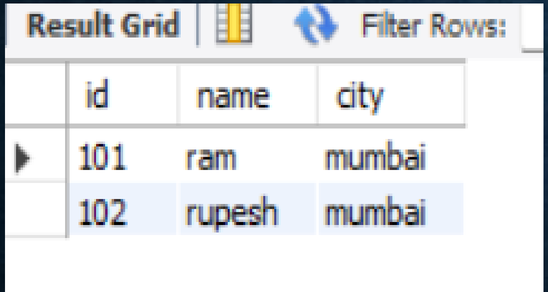
```
create table student(roll_no int primary key, #doesn,t allowed repitation,null values
First_Name varchar(10) not null, # doesn;t aloowed null value
marks int default 35, #if not marks given then default value will be taken.
grade char(1) unique, #unique values allow repitation not allowed
city varchar(10));
```

4. BASIC QUERY TO INSERT VALUES IN TABLE

Simple query to insert values

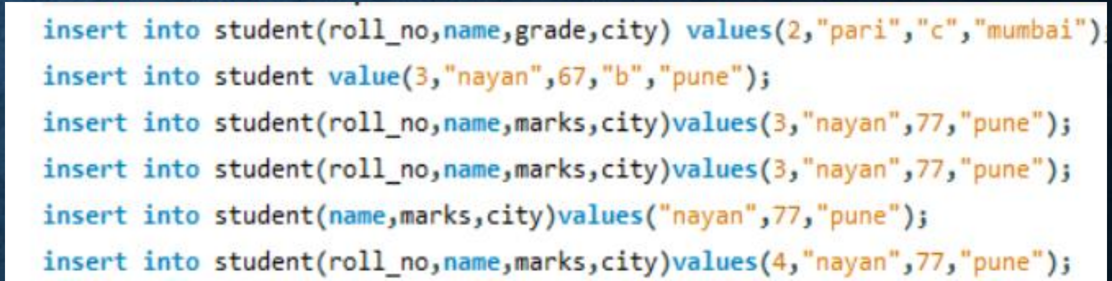


```
insert into table1 values(101,"ram","mumbai");
insert into table1 values(102,"rupesh","mumbai");
```

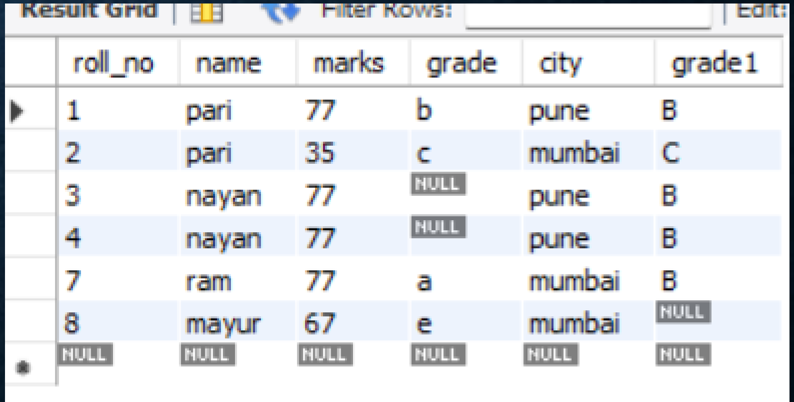


id	name	city
101	ram	mumbai
102	rupesh	mumbai

Query to insert values with null values



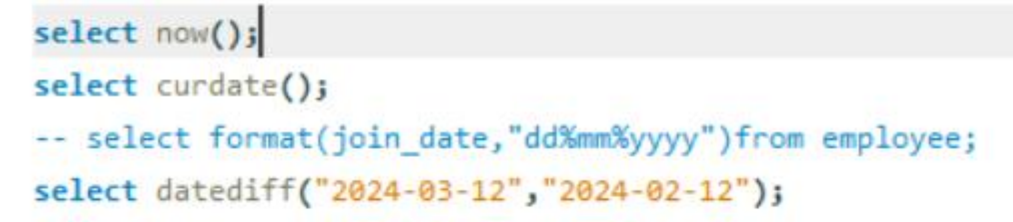
```
insert into student(roll_no,name,grade,city) values(2,"pari","c","mumbai")
insert into student value(3,"nayan",67,"b","pune");
insert into student(roll_no,name,marks,city)values(3,"nayan",77,"pune");
insert into student(roll_no,name,marks,city)values(3,"nayan",77,"pune");
insert into student(name,marks,city)values("nayan",77,"pune");
insert into student(roll_no,name,marks,city)values(4,"nayan",77,"pune");
```



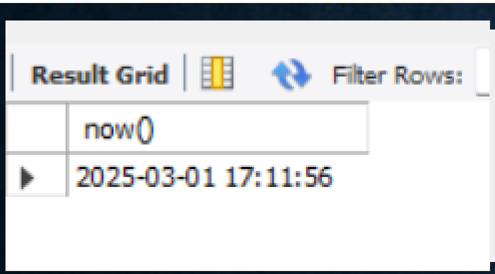
roll_no	name	marks	grade	city	grade1
1	pari	77	b	pune	B
2	pari	35	c	mumbai	C
3	nayan	77	NULL	pune	B
4	nayan	77	NULL	pune	B
7	ram	77	a	mumbai	B
8	mayur	67	e	mumbai	NULL
NULL	NULL	NULL	NULL	NULL	NULL

5.FUNCTION

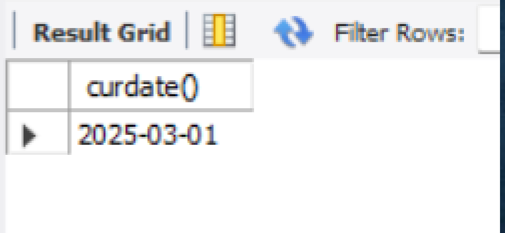
Text function



```sql
select now();
select curdate();
-- select format(join_date,"dd%mm%yyyy")from employee;
select datediff("2024-03-12","2024-02-12");
```

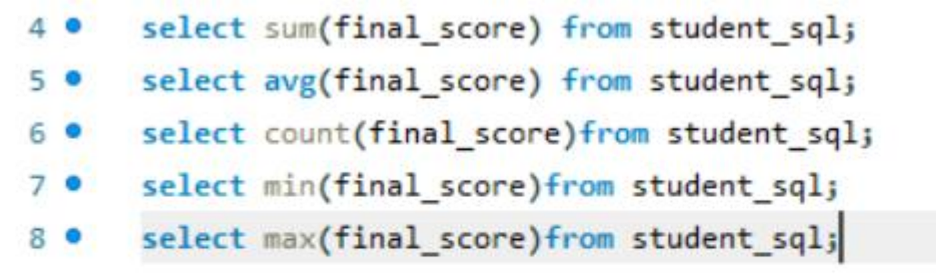


```
now()
2025-03-01 17:11:56
```

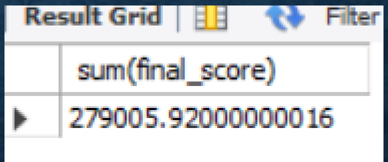


```
curdate()
2025-03-01
```

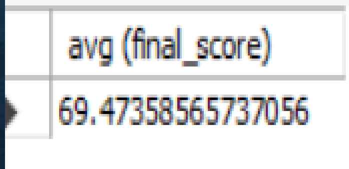
Aggregate function



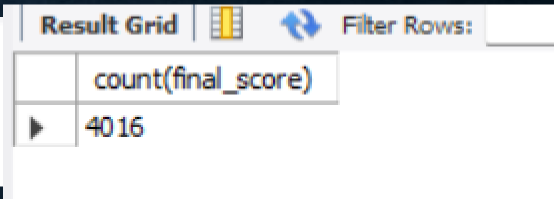
```sql
select sum(final_score) from student_sql;
select avg(final_score) from student_sql;
select count(final_score)from student_sql;
select min(final_score)from student_sql;
select max(final_score)from student_sql;
```



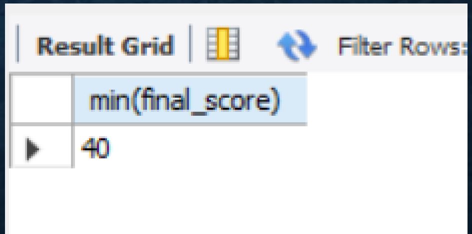
```
sum(final_score)
279005.92000000016
```



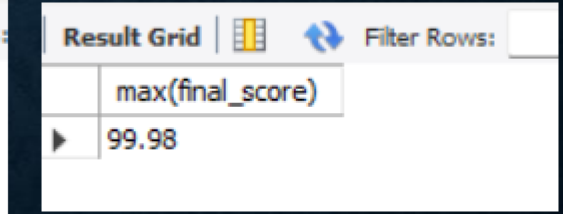
```
avg (final_score)
69.47358565737056
```



```
count(final_score)
4016
```

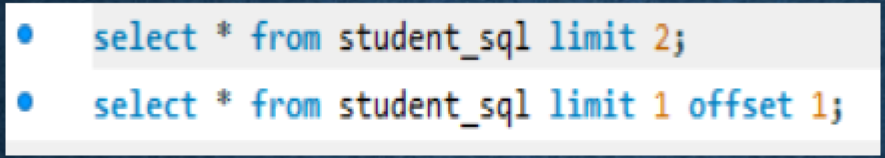


```
min(final_score)
40
```

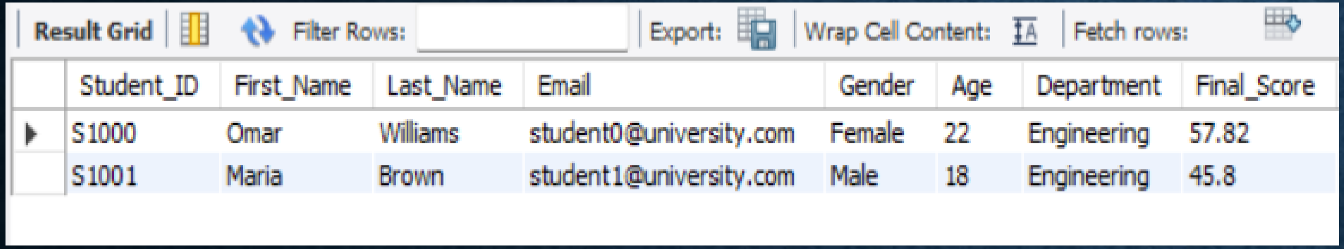


```
max(final_score)
99.98
```

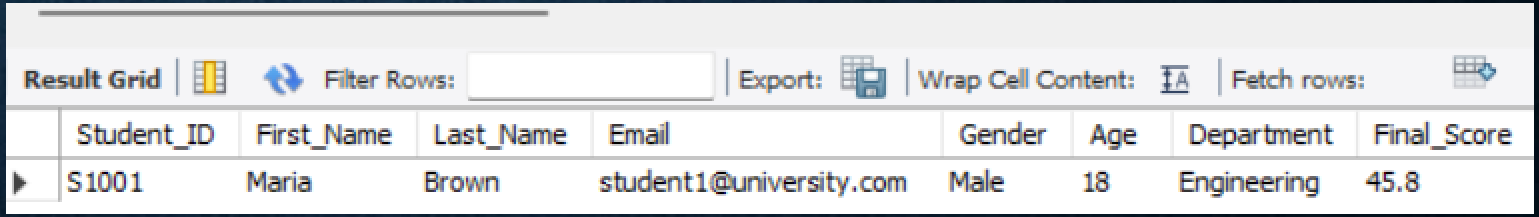
LIMIT CLAUSE WITH OFFSET



```sql
select * from student_sql limit 2;
select * from student_sql limit 1 offset 1;
```

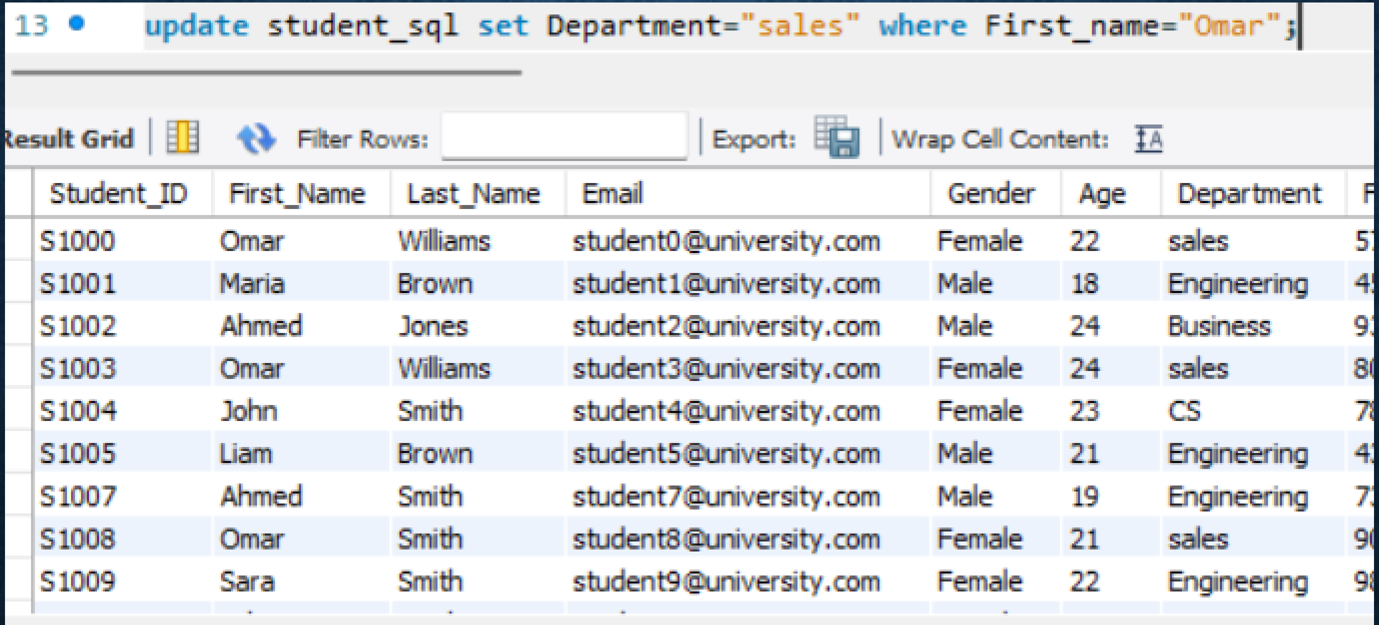


Student_ID	First_Name	Last_Name	Email	Gender	Age	Department	Final_Score
S1000	Omar	Williams	student0@university.com	Female	22	Engineering	57.82
S1001	Maria	Brown	student1@university.com	Male	18	Engineering	45.8



Student_ID	First_Name	Last_Name	Email	Gender	Age	Department	Final_Score
S1001	Maria	Brown	student1@university.com	Male	18	Engineering	45.8

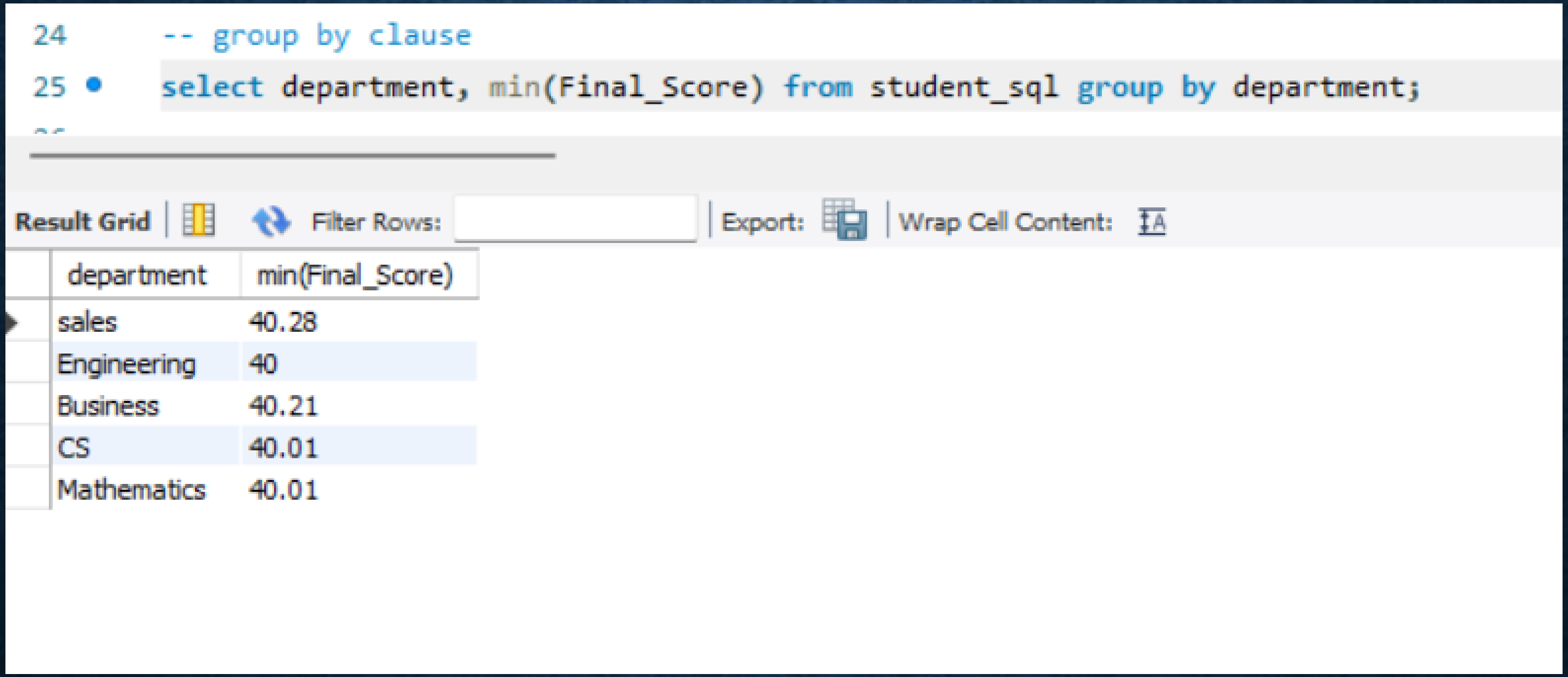
WHERE CLAUSE



```
update student_sql set Department="sales" where First_name="Omar";
```

Student_ID	First_Name	Last_Name	Email	Gender	Age	Department	F
S1000	Omar	Williams	student0@university.com	Female	22	sales	5
S1001	Maria	Brown	student1@university.com	Male	18	Engineering	4
S1002	Ahmed	Jones	student2@university.com	Male	24	Business	9
S1003	Omar	Williams	student3@university.com	Female	24	sales	8
S1004	John	Smith	student4@university.com	Female	23	CS	7
S1005	Liam	Brown	student5@university.com	Male	21	Engineering	4
S1007	Ahmed	Smith	student7@university.com	Male	19	Engineering	7
S1008	Omar	Smith	student8@university.com	Female	21	sales	9
S1009	Sara	Smith	student9@university.com	Female	22	Engineering	9

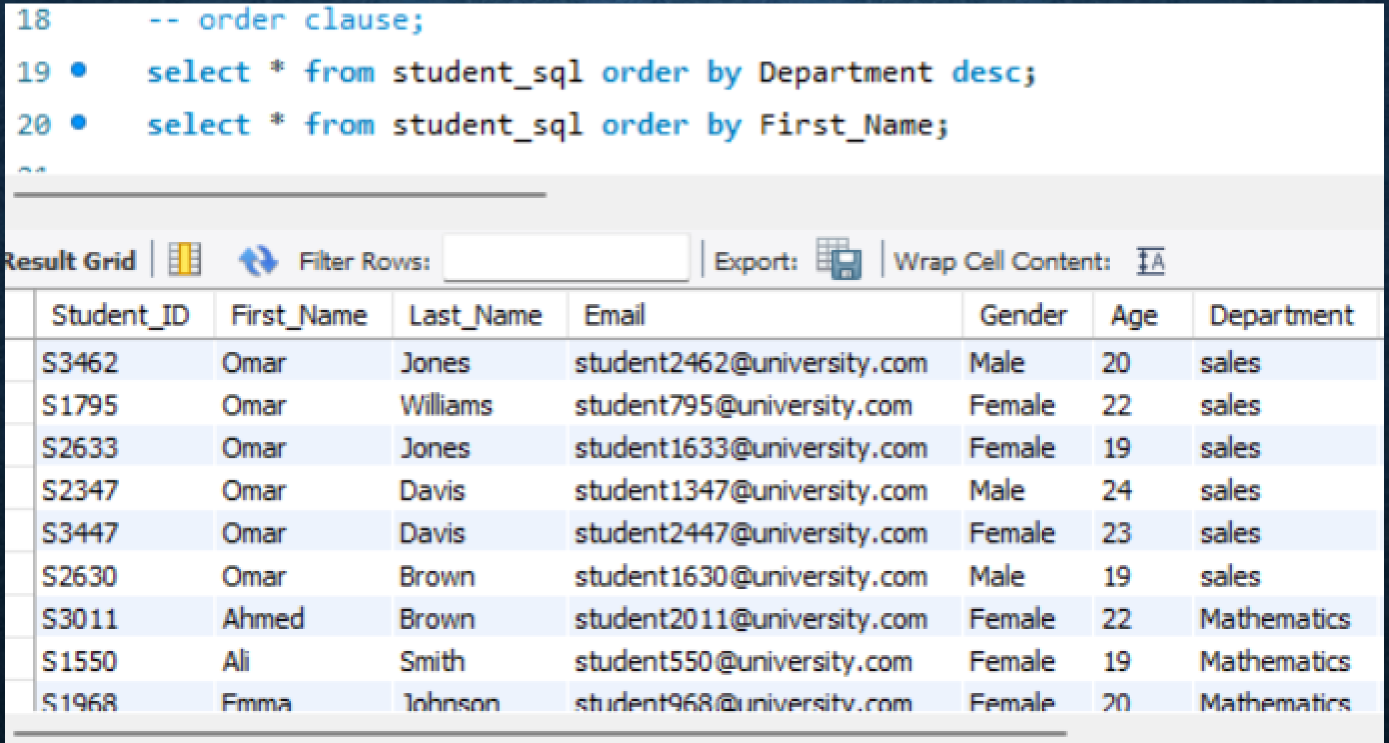
GROUP BY CLAUSE



```
-- group by clause
select department, min(Final_Score) from student_sql group by department;
```

department	min(Final_Score)
sales	40.28
Engineering	40
Business	40.21
CS	40.01
Mathematics	40.01

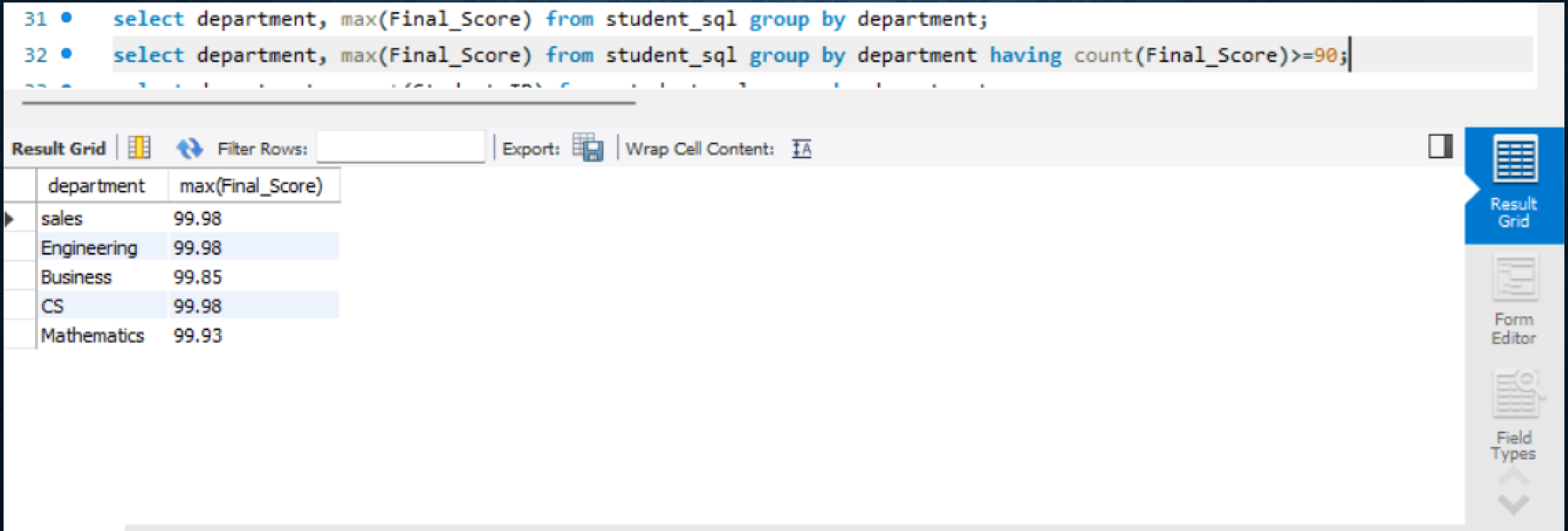
ORDER CLAUSE



```
-- order clause;
select * from student_sql order by Department desc;
select * from student_sql order by First_Name;
```

Student_ID	First_Name	Last_Name	Email	Gender	Age	Department
S3462	Omar	Jones	student2462@university.com	Male	20	sales
S1795	Omar	Williams	student795@university.com	Female	22	sales
S2633	Omar	Jones	student1633@university.com	Female	19	sales
S2347	Omar	Davis	student1347@university.com	Male	24	sales
S3447	Omar	Davis	student2447@university.com	Female	23	sales
S2630	Omar	Brown	student1630@university.com	Male	19	sales
S3011	Ahmed	Brown	student2011@university.com	Female	22	Mathematics
S1550	Ali	Smith	student550@university.com	Female	19	Mathematics
S1968	Emma	Johnson	student968@university.com	Female	20	Mathematics

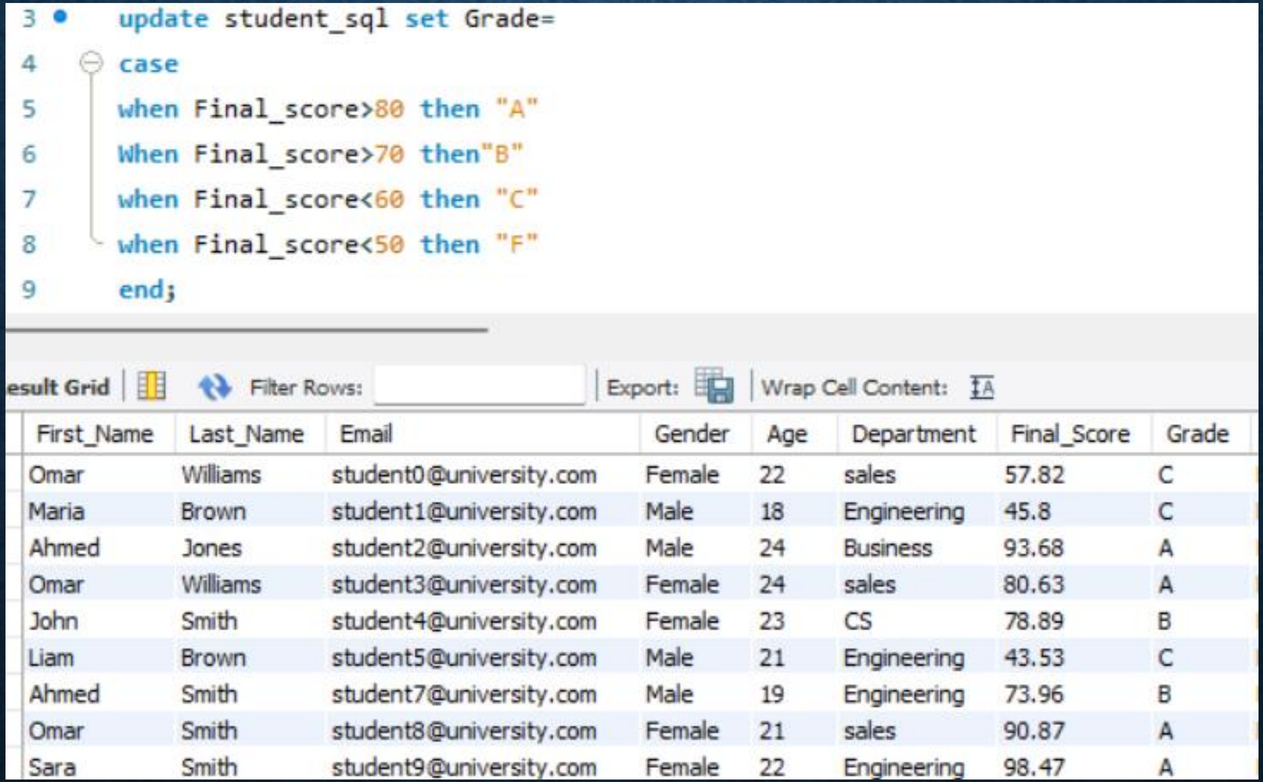
HAVING CLAUSE



```sql
select department, max(Final_Score) from student_sql group by department;
select department, max(Final_Score) from student_sql group by department having count(Final_Score)>=90;
```

department	max(Final_Score)
sales	99.98
Engineering	99.98
Business	99.85
CS	99.98
Mathematics	99.93

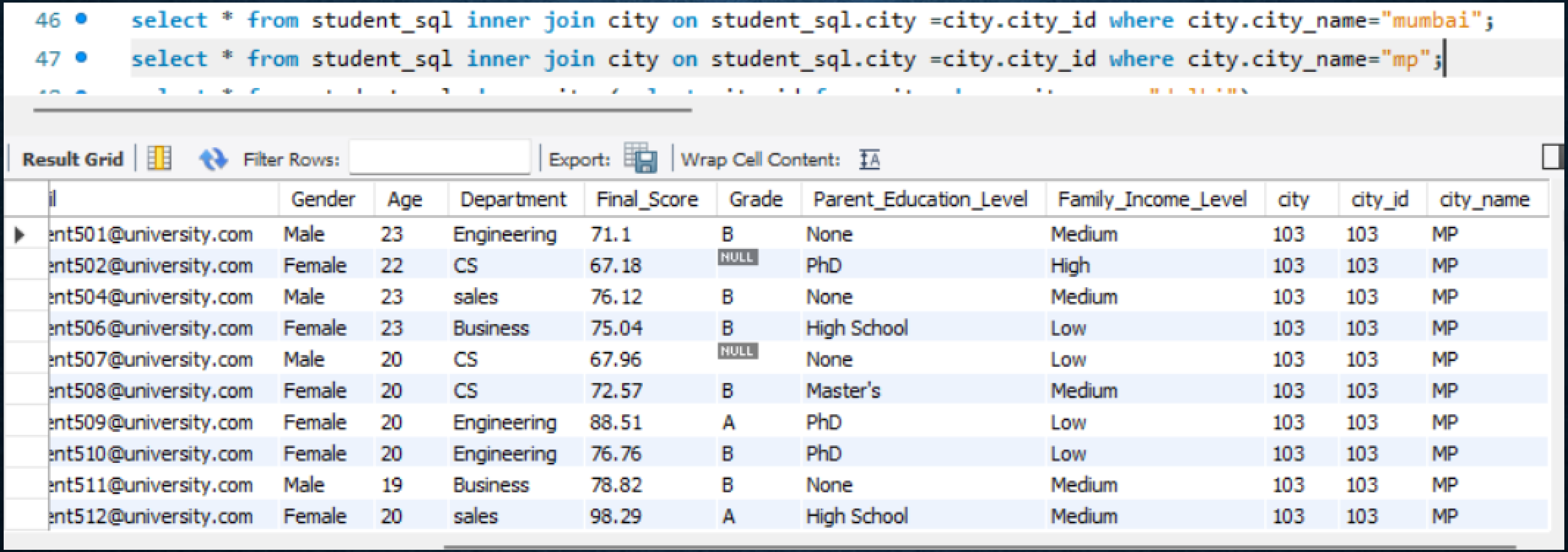
CASE STATEMENT



```
update student_sql set Grade=
case
when Final_score>80 then "A"
When Final_score>70 then"B"
when Final_score<60 then "C"
when Final_score<50 then "F"
end;
```

First_Name	Last_Name	Email	Gender	Age	Department	Final_Score	Grade
Omar	Williams	student0@university.com	Female	22	sales	57.82	C
Maria	Brown	student1@university.com	Male	18	Engineering	45.8	C
Ahmed	Jones	student2@university.com	Male	24	Business	93.68	A
Omar	Williams	student3@university.com	Female	24	sales	80.63	A
John	Smith	student4@university.com	Female	23	CS	78.89	B
Liam	Brown	student5@university.com	Male	21	Engineering	43.53	C
Ahmed	Smith	student7@university.com	Male	19	Engineering	73.96	B
Omar	Smith	student8@university.com	Female	21	sales	90.87	A
Sara	Smith	student9@university.com	Female	22	Engineering	98.47	A

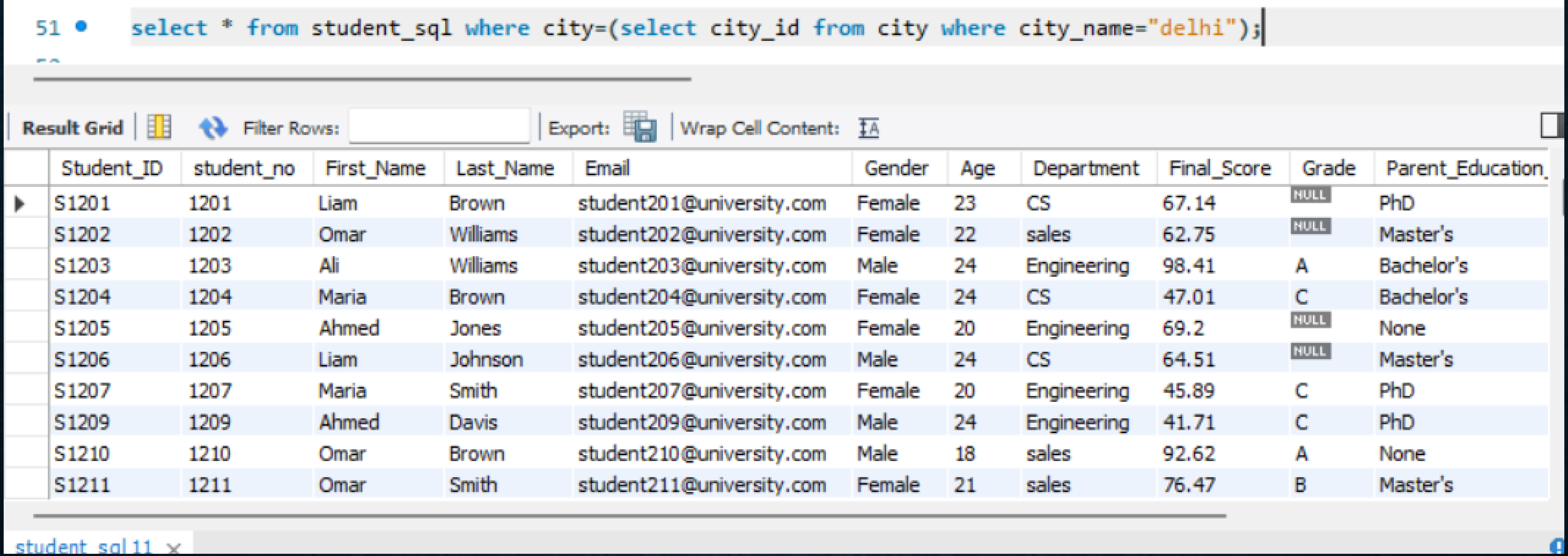
JOIN TABLE



```
select * from student_sql inner join city on student_sql.city =city.city_id where city.city_name="mumbai";
select * from student_sql inner join city on student_sql.city =city.city_id where city.city_name="mp";
```

il	Gender	Age	Department	Final_Score	Grade	Parent_Education_Level	Family_Income_Level	city	city_id	city_name
nt501@university.com	Male	23	Engineering	71.1	B	None	Medium	103	103	MP
nt502@university.com	Female	22	CS	67.18	NULL	PhD	High	103	103	MP
nt504@university.com	Male	23	sales	76.12	B	None	Medium	103	103	MP
nt506@university.com	Female	23	Business	75.04	B	High School	Low	103	103	MP
nt507@university.com	Male	20	CS	67.96	NULL	None	Low	103	103	MP
nt508@university.com	Female	20	CS	72.57	B	Master's	Medium	103	103	MP
nt509@university.com	Female	20	Engineering	88.51	A	PhD	Low	103	103	MP
nt510@university.com	Female	20	Engineering	76.76	B	PhD	Low	103	103	MP
nt511@university.com	Male	19	Business	78.82	B	None	Medium	103	103	MP
nt512@university.com	Female	20	sales	98.29	A	High School	Medium	103	103	MP

SUB QUERIE



```sql
select * from student_sql where city=(select city_id from city where city_name="delhi");
```

Student_ID	student_no	First_Name	Last_Name	Email	Gender	Age	Department	Final_Score	Grade	Parent_Education_
S1201	1201	Liam	Brown	student201@university.com	Female	23	CS	67.14	NULL	PhD
S1202	1202	Omar	Williams	student202@university.com	Female	22	sales	62.75	NULL	Master's
S1203	1203	Ali	Williams	student203@university.com	Male	24	Engineering	98.41	A	Bachelor's
S1204	1204	Maria	Brown	student204@university.com	Female	24	CS	47.01	C	Bachelor's
S1205	1205	Ahmed	Jones	student205@university.com	Female	20	Engineering	69.2	NULL	None
S1206	1206	Liam	Johnson	student206@university.com	Male	24	CS	64.51	NULL	Master's
S1207	1207	Maria	Smith	student207@university.com	Female	20	Engineering	45.89	C	PhD
S1209	1209	Ahmed	Davis	student209@university.com	Male	24	Engineering	41.71	C	PhD
S1210	1210	Omar	Brown	student210@university.com	Male	18	sales	92.62	A	None
S1211	1211	Omar	Smith	student211@university.com	Female	21	sales	76.47	B	Master's

Conclusion

The **Student table** provides a clear overview of individual student records, including details such as names, roll numbers, and marks. Based on the data:

- Most students have **consistent academic performance**, with several showing high achievement.
- There is a **range of marks**, indicating diversity in learning pace and understanding.
- The data suggests that while many students are performing well, a few may benefit from **additional academic support** or focused attention.
- This table serves as a useful tool for **tracking student progress**, identifying top performers, and highlighting those who may need intervention.