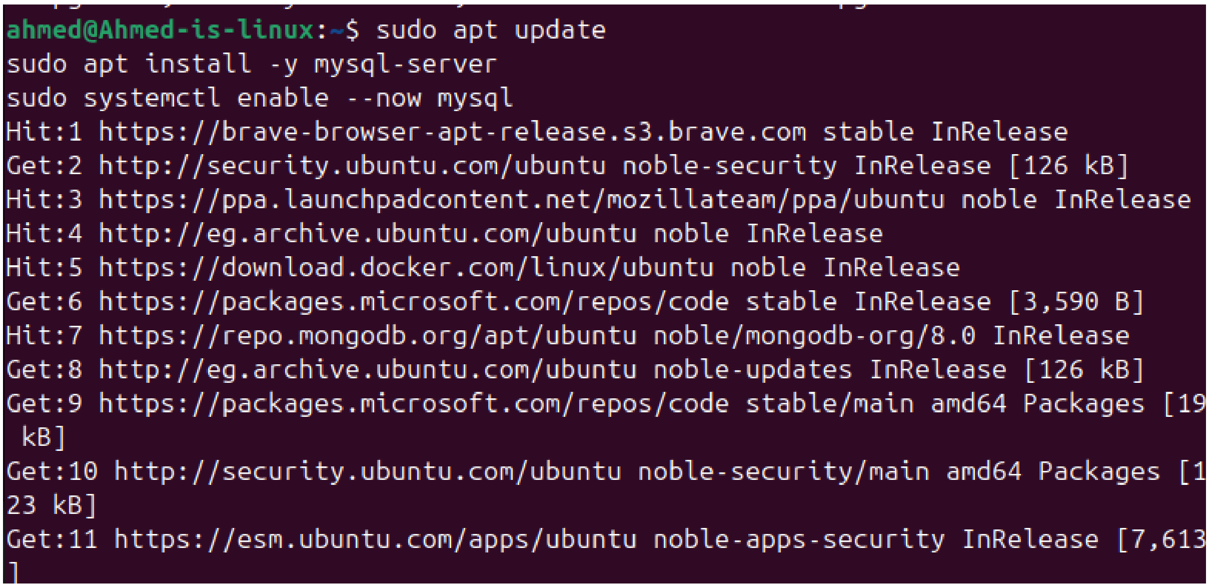
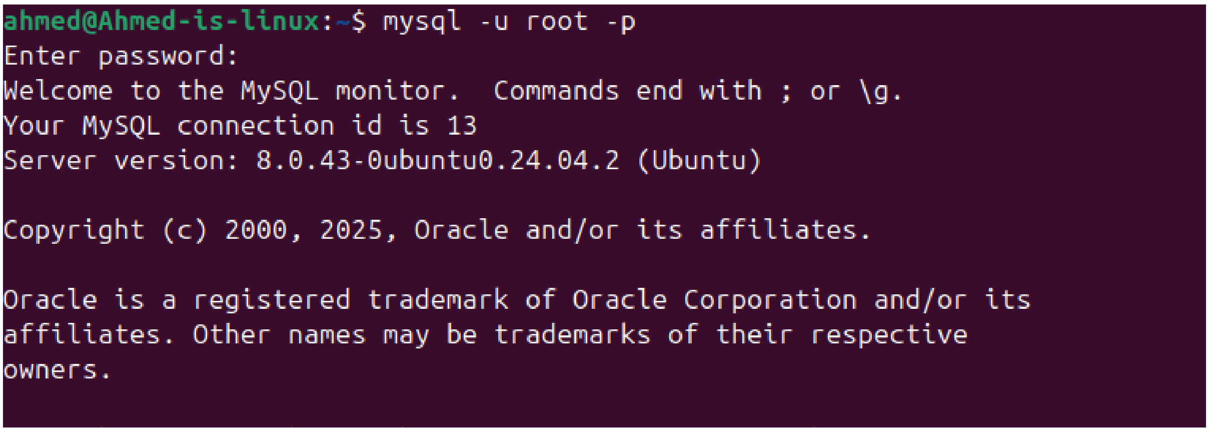


```
ahmed@Ahmed-is-linux:~$ sudo apt update
sudo apt install -y mysql-server
sudo systemctl enable --now mysql
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:3 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu noble InRelease
Hit:4 http://eg.archive.ubuntu.com/ubuntu noble InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://packages.microsoft.com/repos/code stable InRelease [3,590 B]
Hit:7 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 InRelease
Get:8 http://eg.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:9 https://packages.microsoft.com/repos/code stable/main amd64 Packages [19
 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1
23 kB]
Get:11 https://esm.ubuntu.com/apps/ubuntu noble-apps-security InRelease [7,613
]
```



```
ahmed@Ahmed-is-linux:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Student : الاسم، الإيميل، العنوان

Phone:اكتر من رقم للطالب

Subject: اسم المادة، الوصف، اعلي درجه

Exam :التاريخ وامتحان  ايه

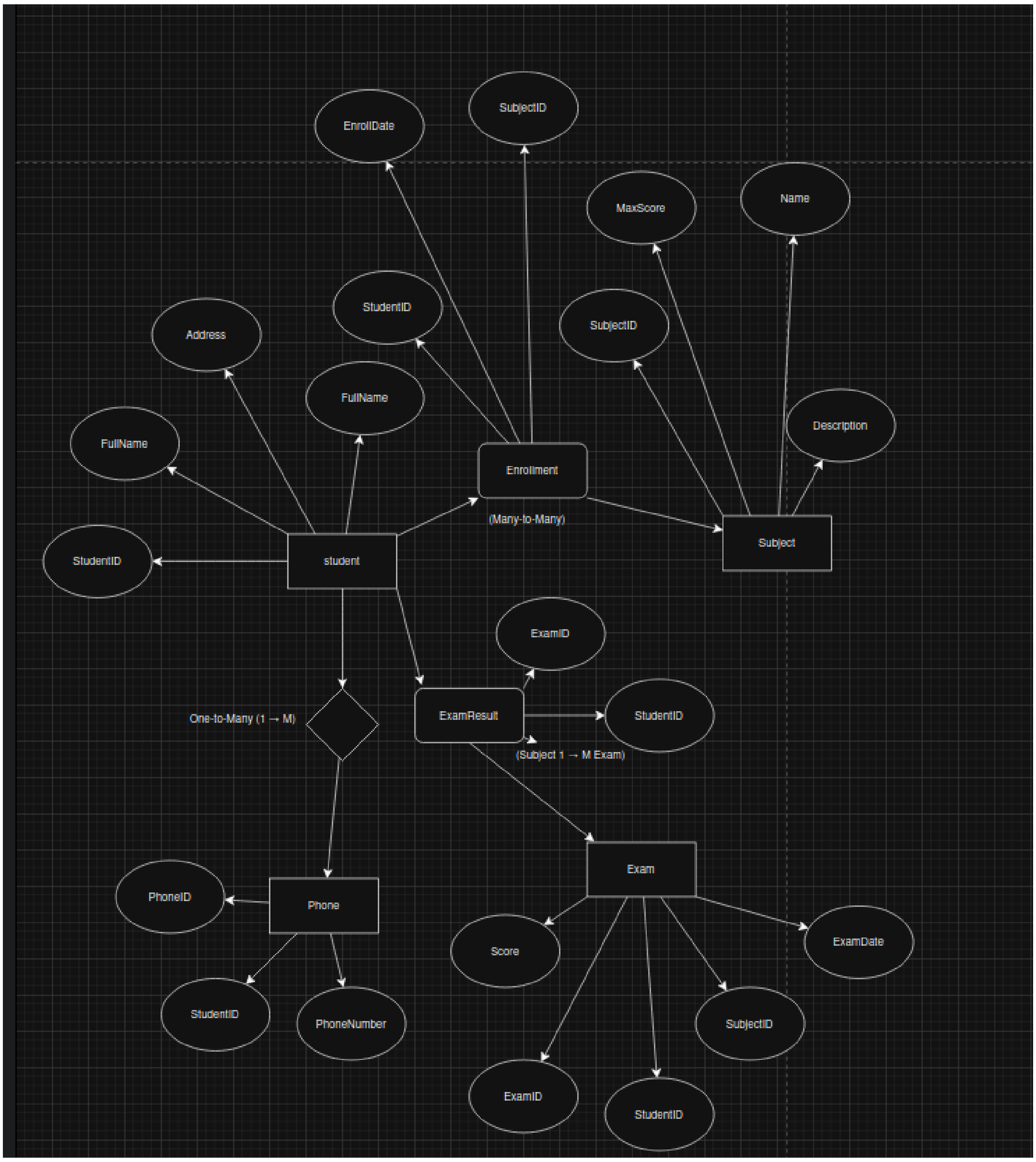
Enrollment : بين الطالب والماده

ExamResult : الدرجه في ا

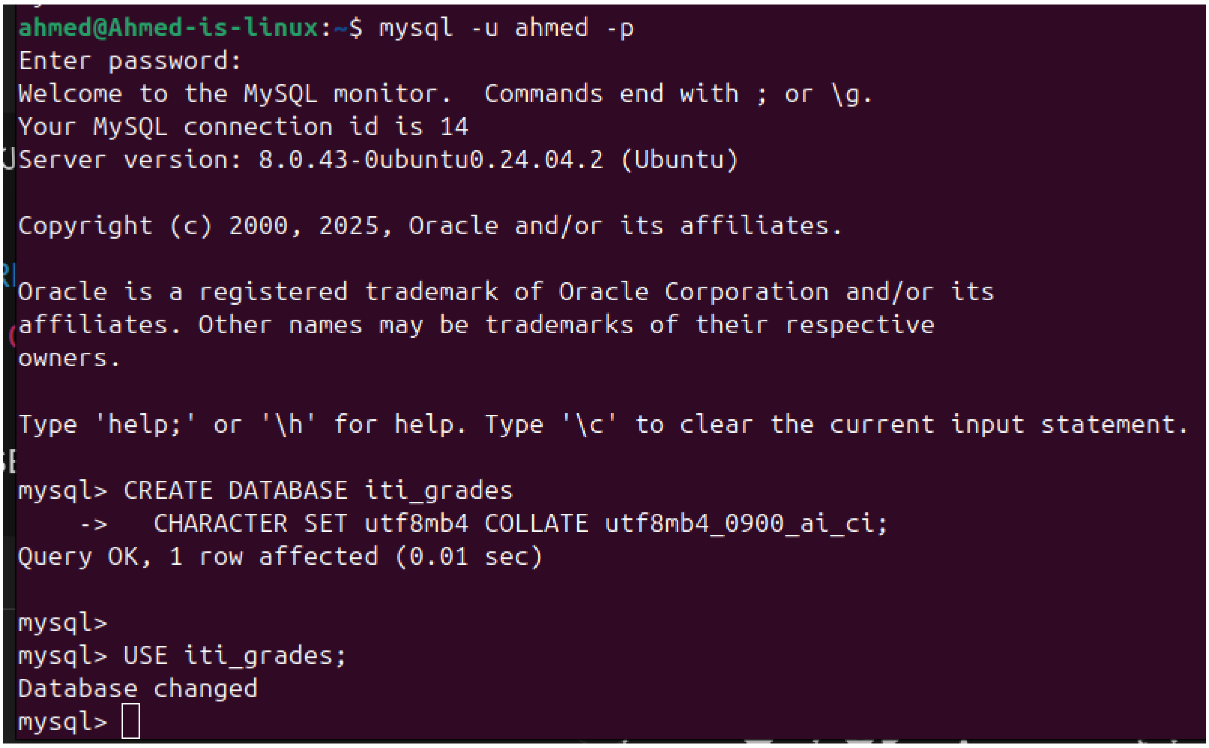
Student 1—M Phone

- Student M—N Subject عبر Enrollment
- Subject 1—M Exam
- Student M—N Exam عبر ExamResult

F



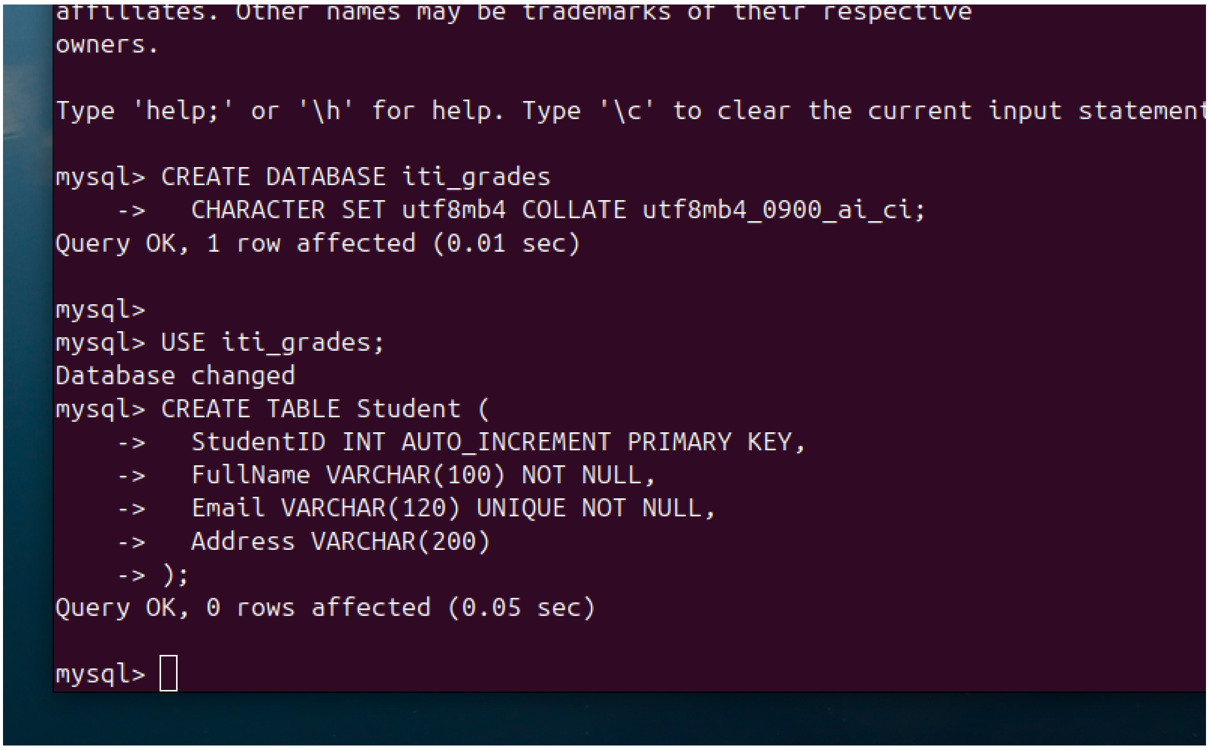
```
CREATE DATABASE iti_grades CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;

USE iti_grades;
```



```
ahmed@Ahmed-is-linux:~$ mysql -u ahmed -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE iti_grades
    ->   CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> USE iti_grades;
Database changed
mysql> 
```

```
CREATE TABLE Student (
 StudentID INT AUTO_INCREMENT PRIMARY KEY,
 FullName VARCHAR(100) NOT NULL,
 Email VARCHAR(120) UNIQUE NOT NULL,
 Address VARCHAR(200)
```

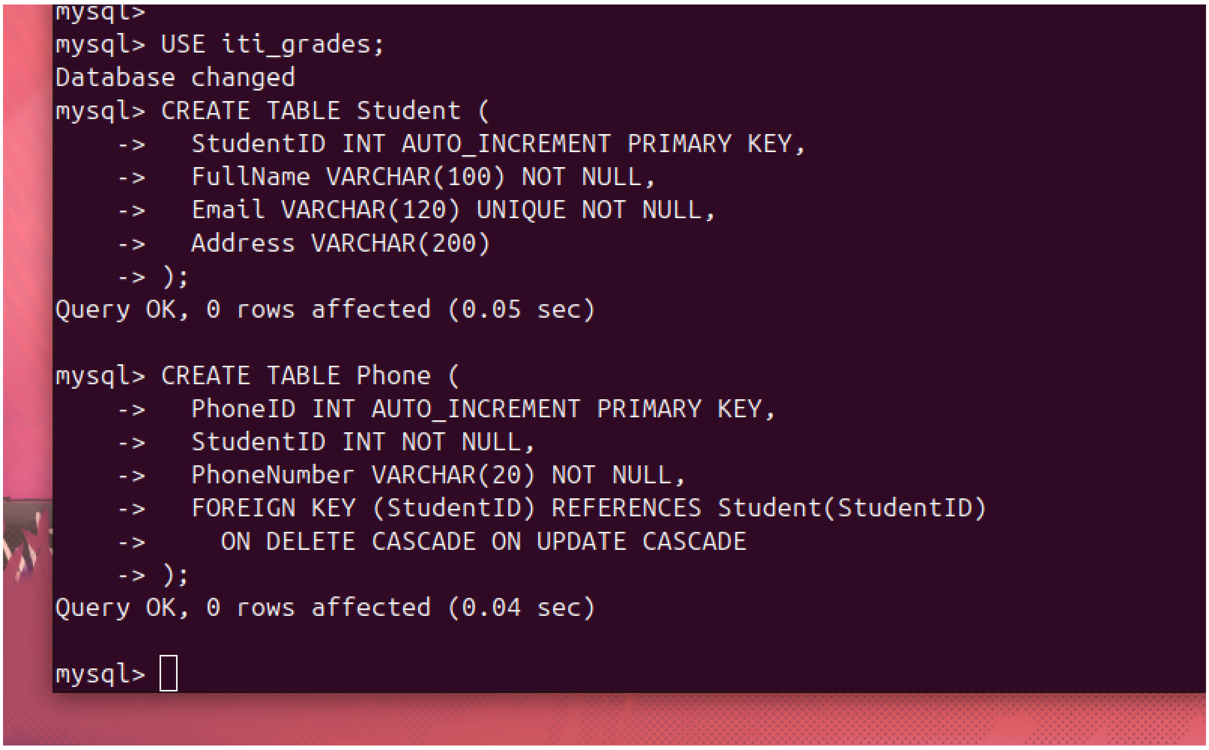
);



```
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement

mysql> CREATE DATABASE iti_grades
    ->   CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> USE iti_grades;
Database changed
mysql> CREATE TABLE Student (
    ->   StudentID INT AUTO_INCREMENT PRIMARY KEY,
    ->   FullName VARCHAR(100) NOT NULL,
    ->   Email VARCHAR(120) UNIQUE NOT NULL,
    ->   Address VARCHAR(200)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> 
```

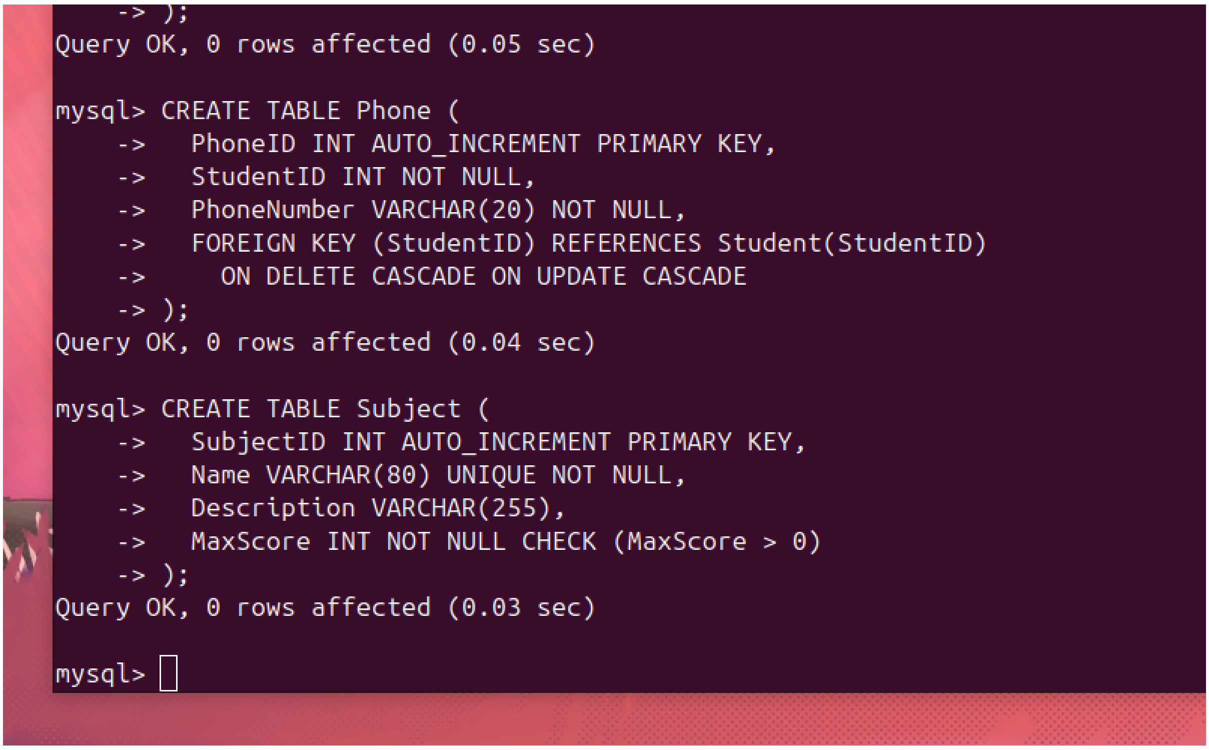
```
CREATE TABLE Phone (
 PhoneID INT AUTO_INCREMENT PRIMARY KEY,
 StudentID INT NOT NULL,
 PhoneNumber VARCHAR(20) NOT NULL,
 FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
  ON DELETE CASCADE ON UPDATE CASCADE
```

);



```
mysql>
mysql> USE iti_grades;
Database changed
mysql> CREATE TABLE Student (
    ->   StudentID INT AUTO_INCREMENT PRIMARY KEY,
    ->   FullName VARCHAR(100) NOT NULL,
    ->   Email VARCHAR(120) UNIQUE NOT NULL,
    ->   Address VARCHAR(200)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Phone (
    ->   PhoneID INT AUTO_INCREMENT PRIMARY KEY,
    ->   StudentID INT NOT NULL,
    ->   PhoneNumber VARCHAR(20) NOT NULL,
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> 
```

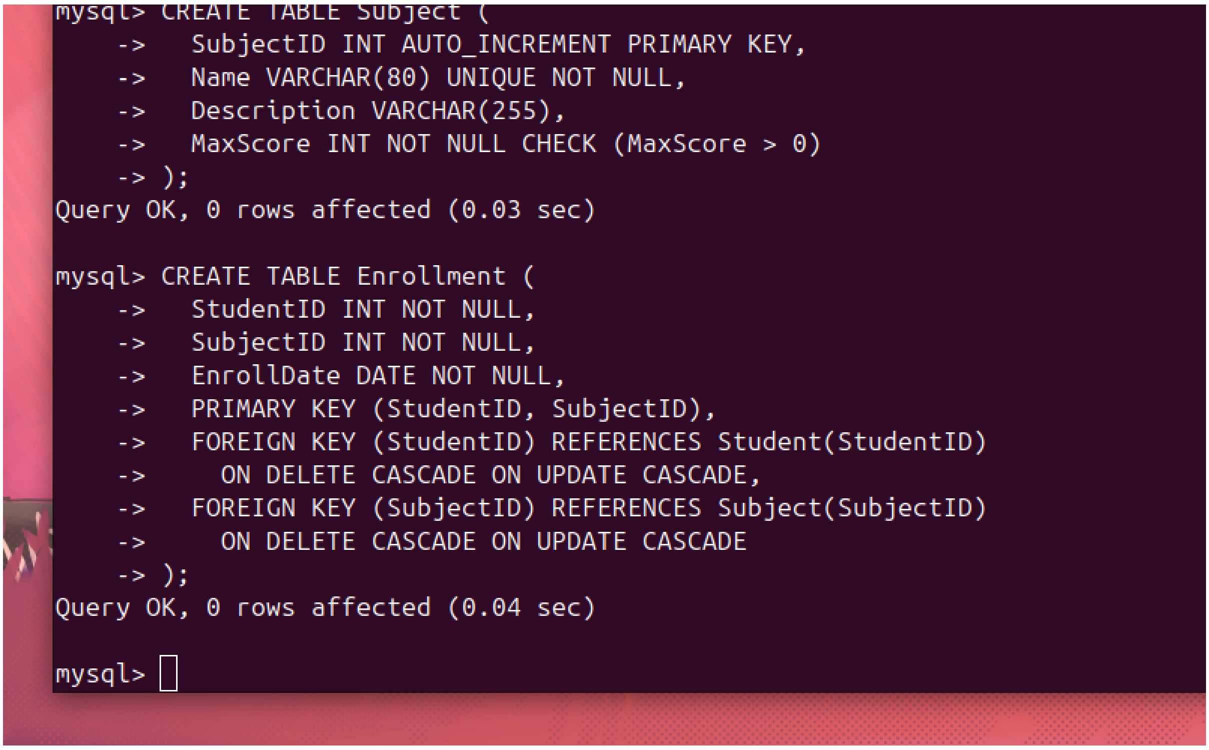
```
CREATE TABLE Subject (

 SubjectID INT AUTO_INCREMENT PRIMARY KEY,

 Name VARCHAR(80) UNIQUE NOT NULL,

 Description VARCHAR(255),

 MaxScore INT NOT NULL CHECK (MaxScore > 0)

);
```



```
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Phone (
    ->   PhoneID INT AUTO_INCREMENT PRIMARY KEY,
    ->   StudentID INT NOT NULL,
    ->   PhoneNumber VARCHAR(20) NOT NULL,
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Subject (
    ->   SubjectID INT AUTO_INCREMENT PRIMARY KEY,
    ->   Name VARCHAR(80) UNIQUE NOT NULL,
    ->   Description VARCHAR(255),
    ->   MaxScore INT NOT NULL CHECK (MaxScore > 0)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> 
```

```
CREATE TABLE Enrollment (
 StudentID INT NOT NULL,
 SubjectID INT NOT NULL,
 EnrollDate DATE NOT NULL,
 PRIMARY KEY (StudentID, SubjectID),
 FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
  ON DELETE CASCADE ON UPDATE CASCADE,
 FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
  ON DELETE CASCADE ON UPDATE CASCADE
```

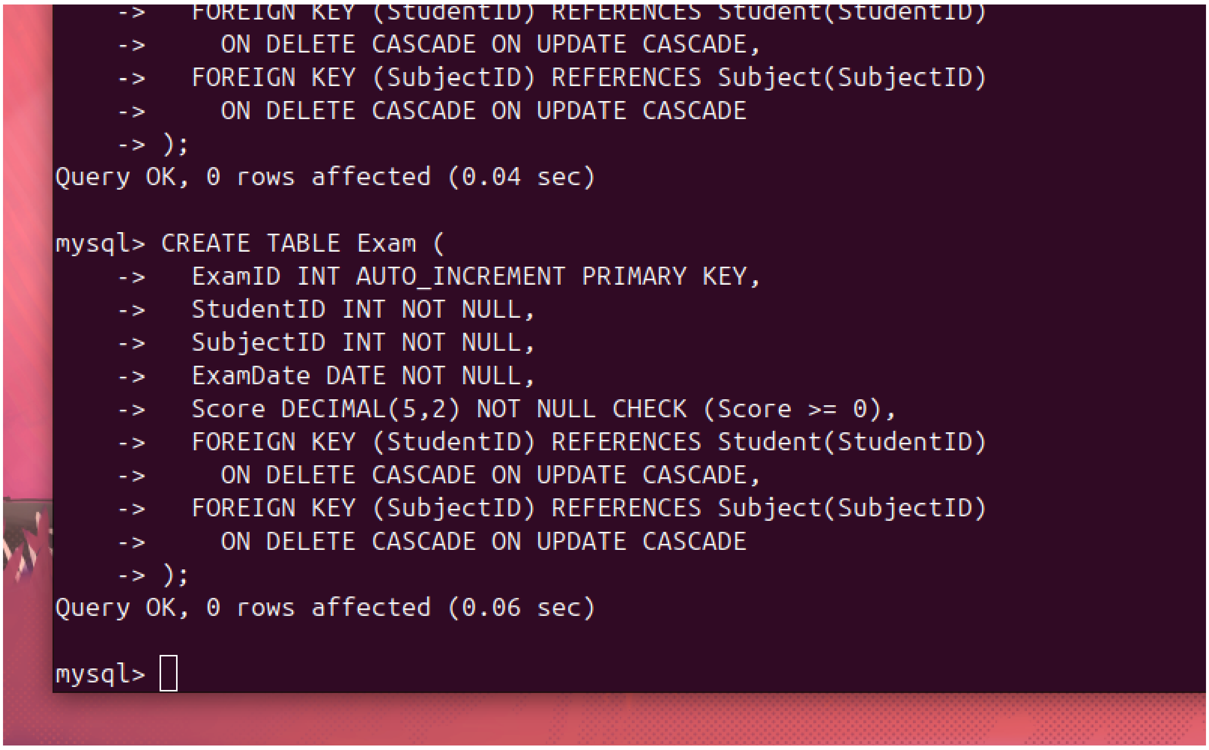
);



```
mysql> CREATE TABLE Subject (
    ->   SubjectID INT AUTO_INCREMENT PRIMARY KEY,
    ->   Name VARCHAR(80) UNIQUE NOT NULL,
    ->   Description VARCHAR(255),
    ->   MaxScore INT NOT NULL CHECK (MaxScore > 0)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Enrollment (
    ->   StudentID INT NOT NULL,
    ->   SubjectID INT NOT NULL,
    ->   EnrollDate DATE NOT NULL,
    ->   PRIMARY KEY (StudentID, SubjectID),
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE,
    ->   FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> 
```

```
CREATE TABLE Exam (
 ExamID INT AUTO_INCREMENT PRIMARY KEY,
 StudentID INT NOT NULL,
 SubjectID INT NOT NULL,
 ExamDate DATE NOT NULL,
 Score DECIMAL(5,2) NOT NULL CHECK (Score >= 0),
 FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
  ON DELETE CASCADE ON UPDATE CASCADE,
 FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
  ON DELETE CASCADE ON UPDATE CASCADE
```

);



```
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE,
    ->   FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Exam (
    ->   ExamID INT AUTO_INCREMENT PRIMARY KEY,
    ->   StudentID INT NOT NULL,
    ->   SubjectID INT NOT NULL,
    ->   ExamDate DATE NOT NULL,
    ->   Score DECIMAL(5,2) NOT NULL CHECK (Score >= 0),
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE,
    ->   FOREIGN KEY (SubjectID) REFERENCES Subject(SubjectID)
    ->     ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> 
```

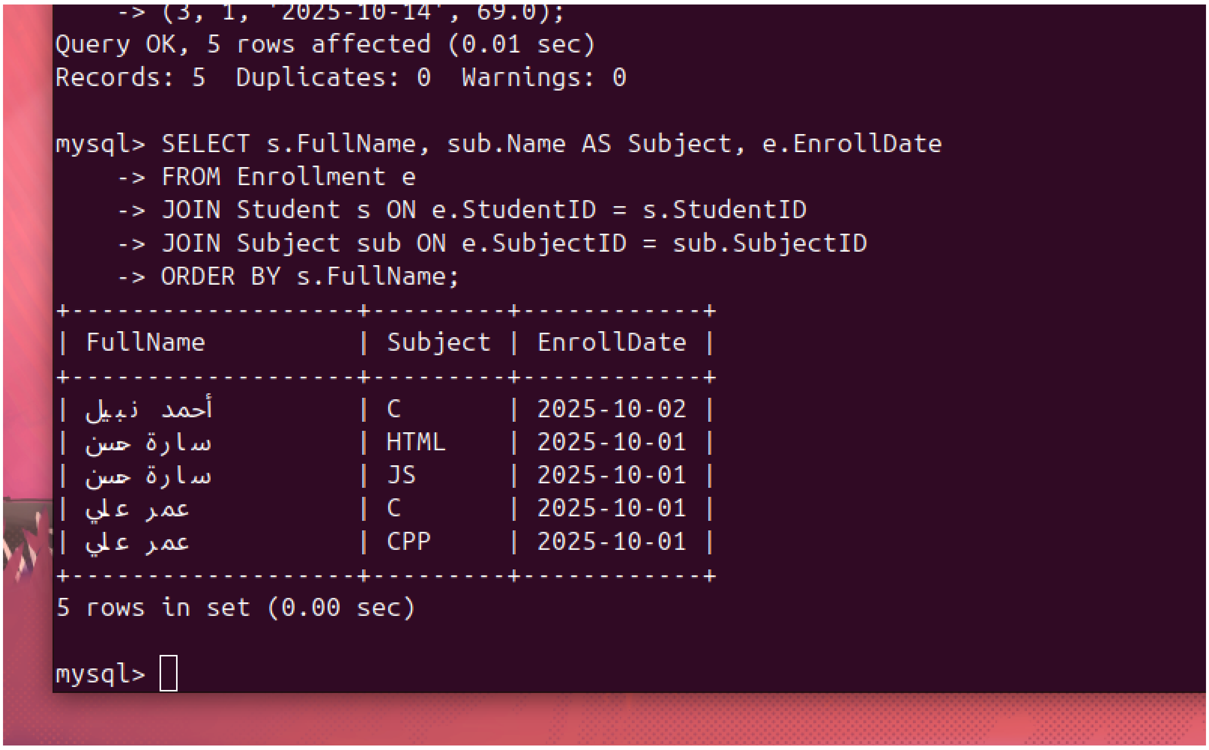
SELECT s.FullName, sub.Name AS Subject, e.EnrollDate

FROM Enrollment e

JOIN Student s ON e.StudentID = s.StudentID

JOIN Subject sub ON e.SubjectID = sub.SubjectID

ORDER BY s.FullName;



```
    -> (3, 1, '2025-10-14', 69.0);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT s.FullName, sub.Name AS Subject, e.EnrollDate
    -> FROM Enrollment e
    -> JOIN Student s ON e.StudentID = s.StudentID
    -> JOIN Subject sub ON e.SubjectID = sub.SubjectID
    -> ORDER BY s.FullName;
+-------------------+---------+------------+
| FullName          | Subject | EnrollDate |
+-------------------+---------+------------+
| أحمد نبيل         | C       | 2025-10-02 |
| سارة حسن          | HTML    | 2025-10-01 |
| سارة حسن          | JS      | 2025-10-01 |
| عمر علي           | C       | 2025-10-01 |
| عمر علي           | CPP     | 2025-10-01 |
+-------------------+---------+------------+
5 rows in set (0.00 sec)

mysql> 
```

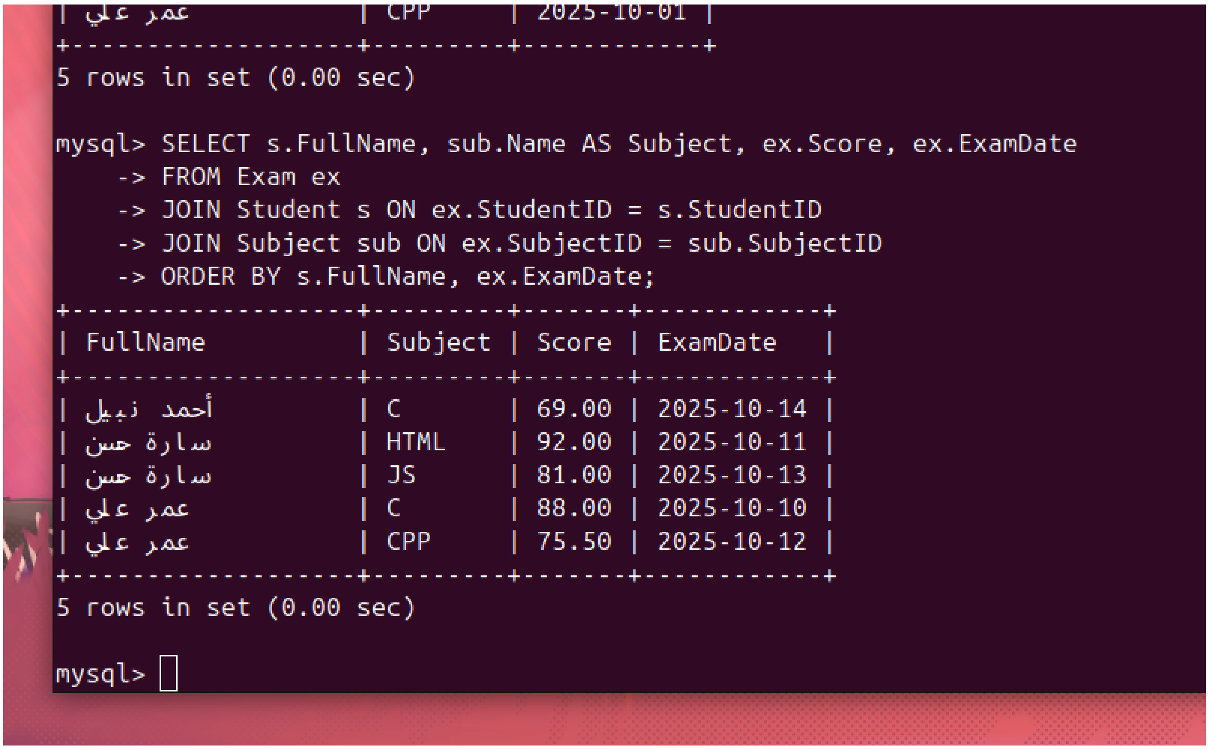
SELECT s.FullName, sub.Name AS Subject, ex.Score, ex.ExamDate

FROM Exam ex

JOIN Student s ON ex.StudentID = s.StudentID

JOIN Subject sub ON ex.SubjectID = sub.SubjectID

ORDER BY s.FullName, ex.ExamDate;



```
| عمر علي            | CPP     | 2025-10-01 |
+--------------------+---------+------------+
5 rows in set (0.00 sec)

mysql> SELECT s.FullName, sub.Name AS Subject, ex.Score, ex.ExamDate
    -> FROM Exam ex
    -> JOIN Student s ON ex.StudentID = s.StudentID
    -> JOIN Subject sub ON ex.SubjectID = sub.SubjectID
    -> ORDER BY s.FullName, ex.ExamDate;
+--------------------+---------+-------+------------+
| FullName           | Subject | Score | ExamDate   |
+--------------------+---------+-------+------------+
| أحمد نبيل          | C       | 69.00 | 2025-10-14 |
| سارة حسن           | HTML    | 92.00 | 2025-10-11 |
| سارة حسن           | JS      | 81.00 | 2025-10-13 |
| عمر علي            | C       | 88.00 | 2025-10-10 |
| عمر علي            | CPP     | 75.50 | 2025-10-12 |
+--------------------+---------+-------+------------+
5 rows in set (0.00 sec)

mysql> 
```

SELECT sub.Name AS Subject,

MAX(ex.Score) AS MaxScore,

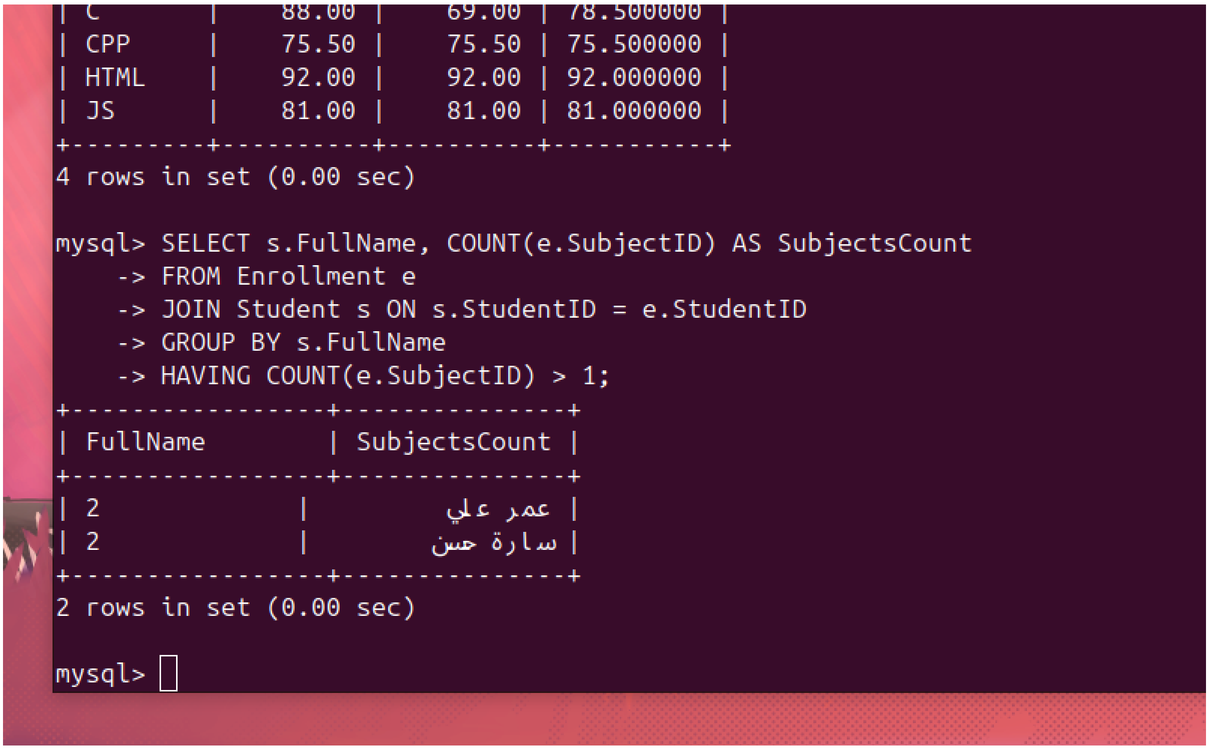
MIN(ex.Score) AS MinScore,

AVG(ex.Score) AS AvgScore

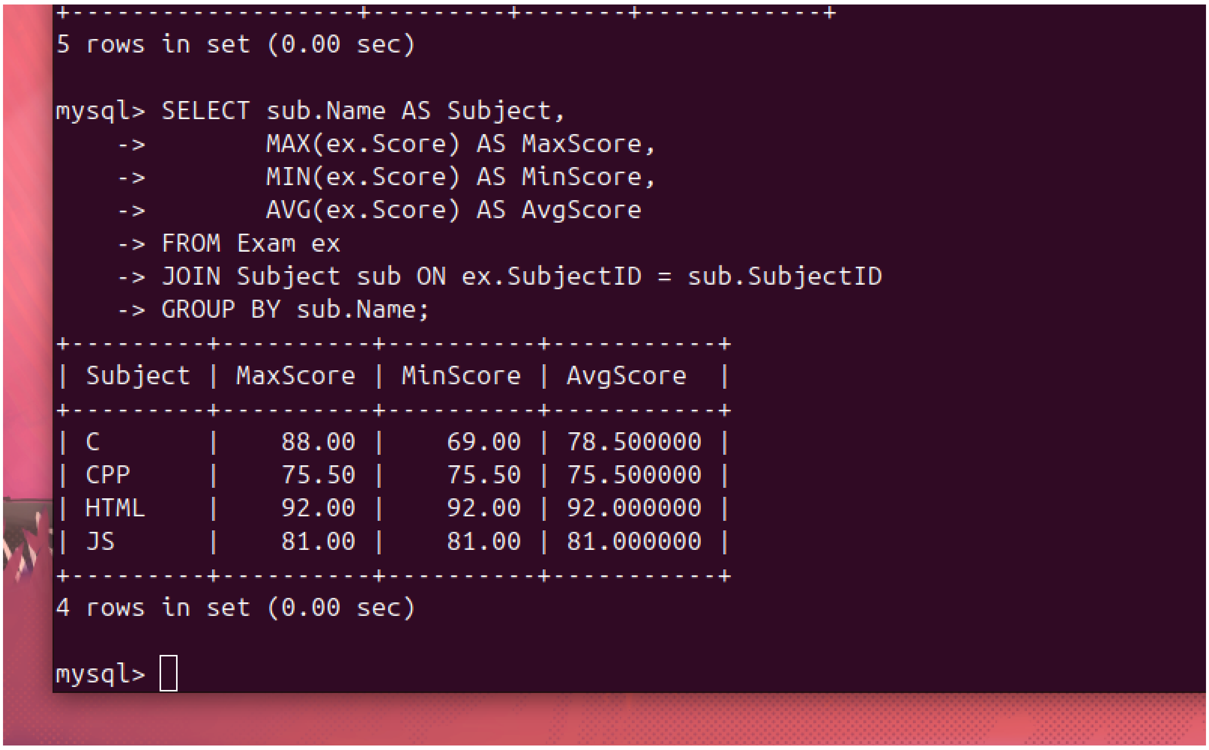
FROM Exam ex

JOIN Subject sub ON ex.SubjectID = sub.SubjectID

```
GROUP BY sub.Name;
```



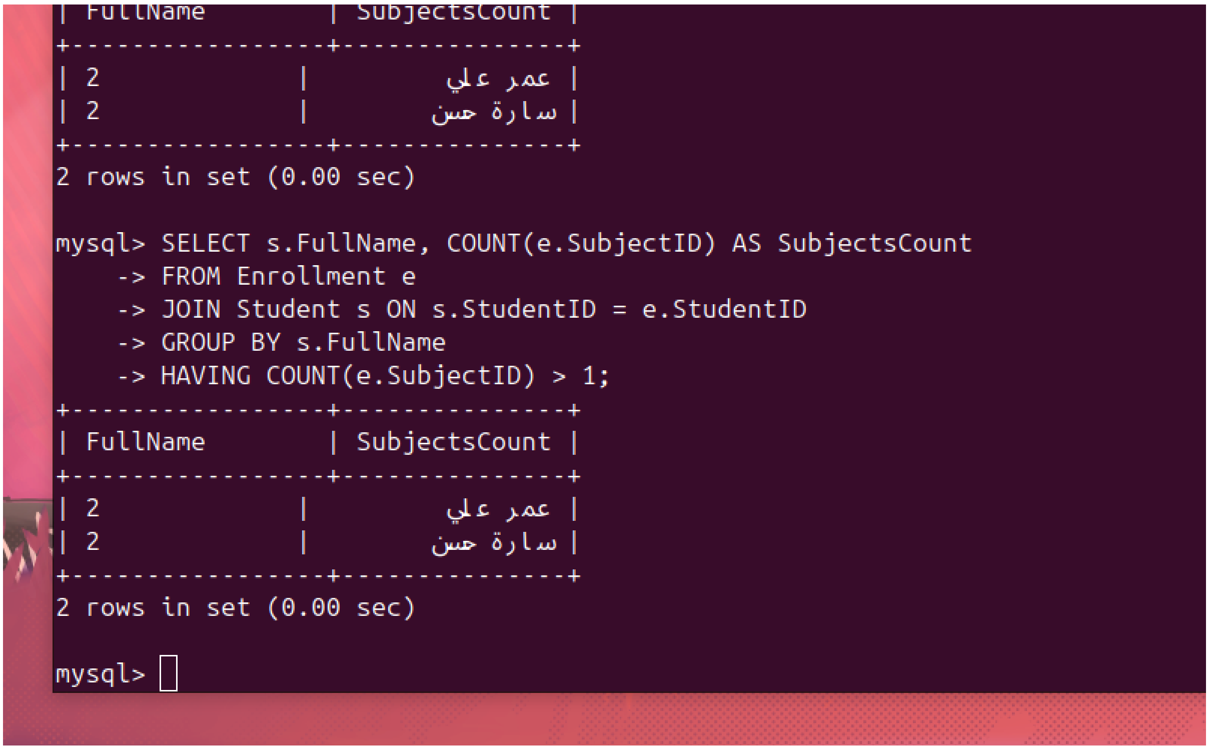
```
| C        |    88.00 |    69.00 | 78.500000 |
| CPP      |    75.50 |    75.50 | 75.500000 |
| HTML     |    92.00 |    92.00 | 92.000000 |
| JS       |    81.00 |    81.00 | 81.000000 |
+---------+---------+---------+----------+
4 rows in set (0.00 sec)

mysql> SELECT s.FullName, COUNT(e.SubjectID) AS SubjectsCount
    -> FROM Enrollment e
    -> JOIN Student s ON s.StudentID = e.StudentID
    -> GROUP BY s.FullName
    -> HAVING COUNT(e.SubjectID) > 1;
+----------------+---------------+
| FullName         | SubjectsCount |
+----------------+---------------+
| 2               |         عمر علي |
| 2               |        سارة حسن |
+----------------+---------------+
2 rows in set (0.00 sec)

mysql> 
```



```
+-----------------+--------+------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT sub.Name AS Subject,
    ->        MAX(ex.Score) AS MaxScore,
    ->        MIN(ex.Score) AS MinScore,
    ->        AVG(ex.Score) AS AvgScore
    -> FROM Exam ex
    -> JOIN Subject sub ON ex.SubjectID = sub.SubjectID
    -> GROUP BY sub.Name;
+---------+----------+----------+-----------+
| Subject | MaxScore | MinScore | AvgScore  |
+---------+----------+----------+-----------+
| C       |    88.00 |    69.00 | 78.500000 |
| CPP     |    75.50 |    75.50 | 75.500000 |
| HTML    |    92.00 |    92.00 | 92.000000 |
| JS      |    81.00 |    81.00 | 81.000000 |
+---------+----------+----------+-----------+
4 rows in set (0.00 sec)

mysql> 
```

```
SELECT s.FullName, COUNT(e.SubjectID) AS SubjectsCount

FROM Enrollment e

JOIN Student s ON s.StudentID = e.StudentID

GROUP BY s.FullName
```

HAVING COUNT(e.SubjectID) > 1;



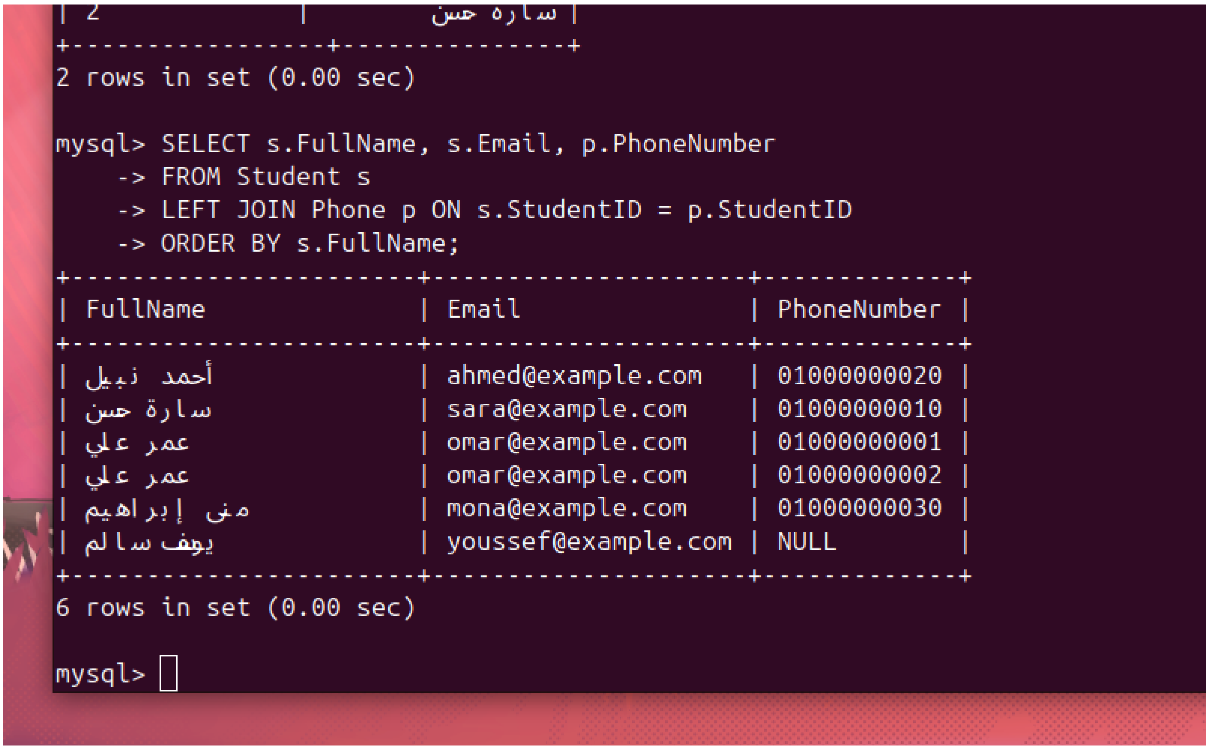
```
| FullName         | SubjectsCount |
+------------------+---------------+
| 2              |         عمر علي |
| 2              |        سارة حسن |
+------------------+---------------+
2 rows in set (0.00 sec)

mysql> SELECT s.FullName, COUNT(e.SubjectID) AS SubjectsCount
    -> FROM Enrollment e
    -> JOIN Student s ON s.StudentID = e.StudentID
    -> GROUP BY s.FullName
    -> HAVING COUNT(e.SubjectID) > 1;
+------------------+---------------+
| FullName         | SubjectsCount |
+------------------+---------------+
| 2              |         عمر علي |
| 2              |        سارة حسن |
+------------------+---------------+
2 rows in set (0.00 sec)

mysql> 
```

SELECT s.FullName, s.Email, p.PhoneNumber

FROM Student s

LEFT JOIN Phone p ON s.StudentID = p.StudentID

ORDER BY s.FullName;

- 

```
| 2               |         سارة حسن |
+-----------------+--------------+
2 rows in set (0.00 sec)

mysql> SELECT s.FullName, s.Email, p.PhoneNumber
    -> FROM Student s
    -> LEFT JOIN Phone p ON s.StudentID = p.StudentID
    -> ORDER BY s.FullName;
+-----------------------+---------------------+-------------+
| FullName              | Email               | PhoneNumber |
+-----------------------+---------------------+-------------+
| أحمد نبيل             | ahmed@example.com   | 01000000020 |
| سارة حسن              | sara@example.com    | 01000000010 |
| عمر علي               | omar@example.com    | 01000000001 |
| عمر علي               | omar@example.com    | 01000000002 |
| منى إبراهيم           | mona@example.com    | 01000000030 |
| يوسف سالم             | youssef@example.com | NULL        |
+-----------------------+---------------------+-------------+
6 rows in set (0.00 sec)

mysql> 
```

NoSQL؟
(Non-Relational)

تستخدم **Documents** أو **Key-Value** أو **Graph**.

Document (مثل MongoDB)

- Key-Value (مثل Redis)
- Column (مثل Cassandra)
- Graph (مثل Neo4j)

---

DBMS

| النوع | الوصف | أمثلة |
|---|---|---|
| **Hierarchical** | شجري (Tree-based) | IBM IMS |
| **Network** | شبكة علاقات متعددة | IDMS |
| **Relational (RDBMS)** | يعتمد على الجداول والعلاقات | MySQL, Oracle, PostgreSQL |
| **NoSQL** | مرن في البنية، غير علائقي | MongoDB, Redis |
| **NewSQL** | يجمع بين السرعة والـACID | CockroachDB, TiDB |