# SQL (Structured Query Language)

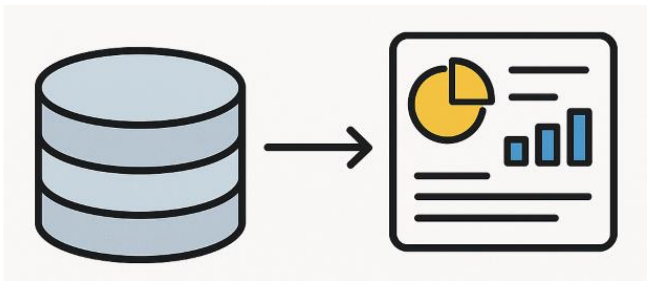
**Data**:- Data is a collection of raw facts, figures, observations, or symbols that represent information but have no meaning until they are processed.

**Examples of Data:**

- Numbers (e.g., 25, 89.5)
- Text (e.g., "Apple", "Student")
- Images (e.g., photos stored digitally)
- Videos
- Audio recordings
- Sensor readings (e.g., temperature = 35°C)
- True/False values

## WHAT IS DATABASE ?

- This Data will stored in DATABASE.
- A database is an organised collection of data that is stored and accessed electronically from a computer system.
- A database stores data in a structured way so it can be easily accessed, managed, updated, Protected, Modified, Analysed.
- Data includes images, text, videos, Excel files, PDFs, XML files, etc.



- Every website in the world is connected to a production server that serves as a collection of databases.
- Example: Facebook needs multiple databases (servers).

## Servers and Databases:

- **Servers** require a specific room called a server room when installed on-premises.
- **Cloud:** No physical server room needed; data is stored in data centres.

| **Server (On-premise)** | **Data centre (Cloud)** |
|---|---|
| Physical server room | Data stored in data centre |

**Security:** Every company or website must have its database connected and secured.

## Production Server:

- Multiple databases are connected together in production servers.
- Example: Companies like PhonePe, IRCTC.

## Types of Data:

- ✓ **Structured Database:**
  - Example: Numbers, tables
  - Structured Database==SQL databases ==relational DB ==RDBMS
  - One table relates to another using Primary Key and Foreign Key.
  - Structured DB- MySQL, Oracle, PostgreSQL

- ✓ **Unstructured Database:**
  - Includes images, videos, audio, streams, APIs, etc.
  - Structured Database==NoSQL databases ==non-relational DB.
  - Unstructured DB - MongoDB, Apache HBase, Cassandra

- ✓ **Vector Database:**
  - Every unstructured data converts to vectors before storing in vector DB.
  - Examples of vector databases: Pinecone, Qdrant, Weaviate, Milvus, Chromadb.
  - Used for RAG (Retrieval Augmented Generation).

## Who Uses Which Database?

| **Role** | **Database Used** |
|---|---|
| Data Engineer | NoSQL DB |
| Data Scientist / Data Analyst | SQL DB |
| ML Developer / Engineer | Vector DB |

## DBMS ( Database Management System ):-

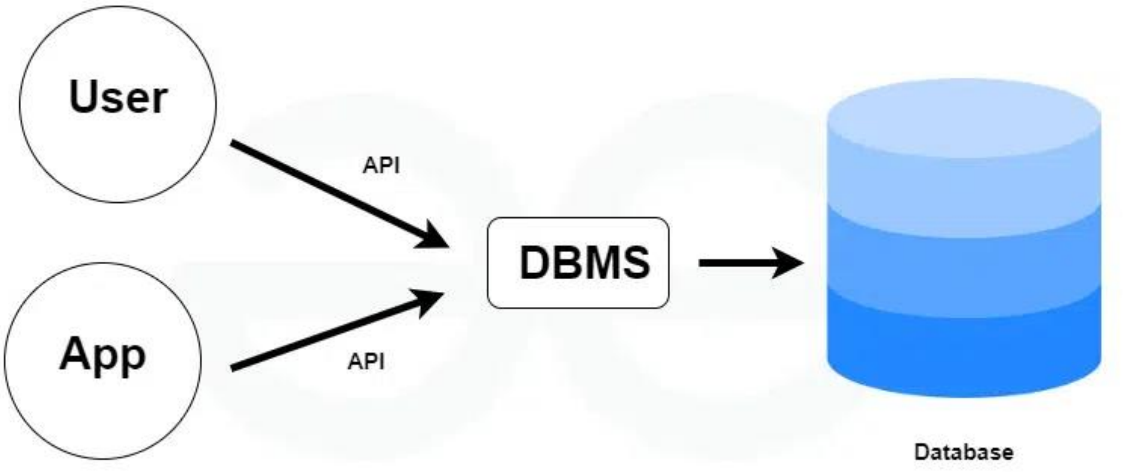
- DBMS ≠ Database
- DBMS (Database Management System) is a software tool used to pull data from the database.

**Popular DBMS Examples:**

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database
- SQLite
- MongoDB

**Functions of DBMS:**

- Retrieve data
- Create, modify, or delete databases



## EVOLUTION OF DATABASE :-

### 1. Flat File Database

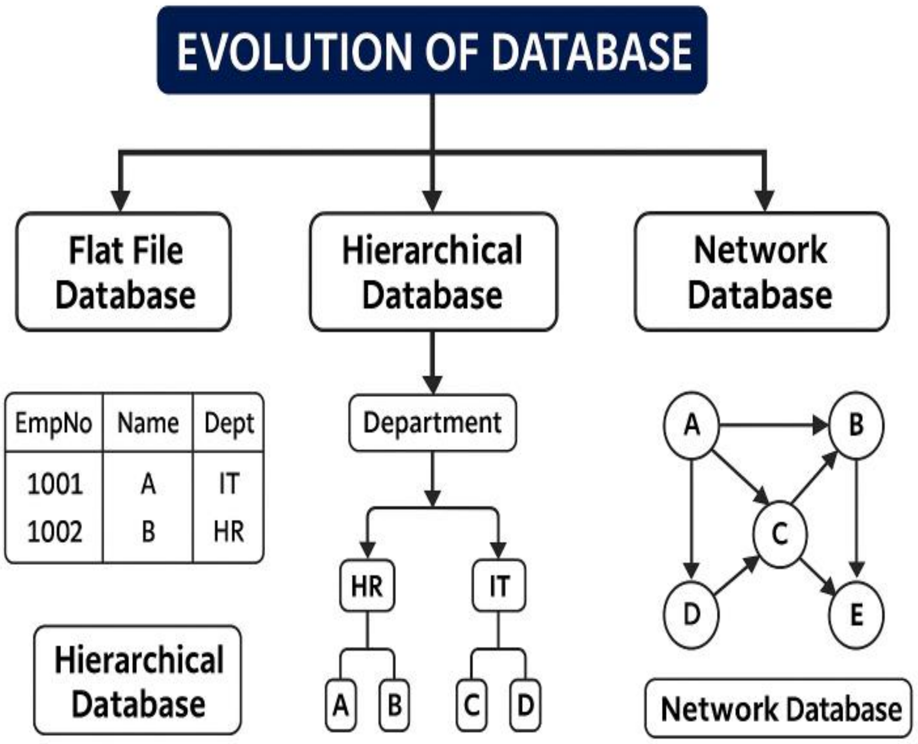
- Stores data in a simple, two-dimensional table (plain text file or spreadsheet).
- No multiple tables or complex relationships between tables.

### 2. Hierarchical Database

- Organises data in a tree-like (parent-child) structure.
- Each record/node has a single parent, except the root.
- Each parent can have multiple children.
- Structure resembles an upside-down tree or organisational chart.

**3. Network Database**

- Represents data using a network model.
- Data organised as collections of records with complex relationships.
- Unlike hierarchical DBs (single parent), each record can have multiple parents.
- Enables more flexible and complex relationships.



## Relational Database = SQL Database = Structured Database :

- Data stored in tables.
- Each table consists of rows and columns.

- Each column has a name and data type.
- Each row is treated as a record, formed by single or multiple columns.

**Employee Table :-** →**( Relationship)** → **Department Table:-**

| **EMPNO** | **EMPLNAME** | **DEPNO** (“foreign key”) | → | **DEPNO** (“primary key”) | **DNAME** | **Location** |
|---|---|---|---|---|---|---|
| 1001 | Sahil | 101 | | 101 | HR | Delhi |
| 1004 | Kavish | 102 | | 102 | Sales | Bangalore |
| 1006 | Aditya | 103 | | 103 | Marketing Specialist | Pune |
| 1005 | Atul | 104 | | 104 | Technical Engineer | Chennai |

**Relationships:**

- **One-to-One Relationship:**
  One table relates to only one other table.
- **One-to-Many Relationship:**
  One table relates to many records in another table.
- **Many-to-One Relationship:**
  Many tables relate to a single table by ID.

## Non-Relational Database = NoSQL DB = Unstructured DB:

**Types of NoSQL Databases:**

1. Key-Value Database
2. Document Database
3. Graph Database
4. Wide Column Database (Column-Family)
5. Search Engine Database
6. Time Series Database

## 1. Key-Value Database

- Examples: Redis, Amazon DynamoDB
- Data is stored as key-value pairs.

**Example:**

  - Hostname: Gonville
  - Port number: 1521

## 2. Document Database

- Examples: MongoDB, CouchDB
- Data is stored in JSON-like documents.

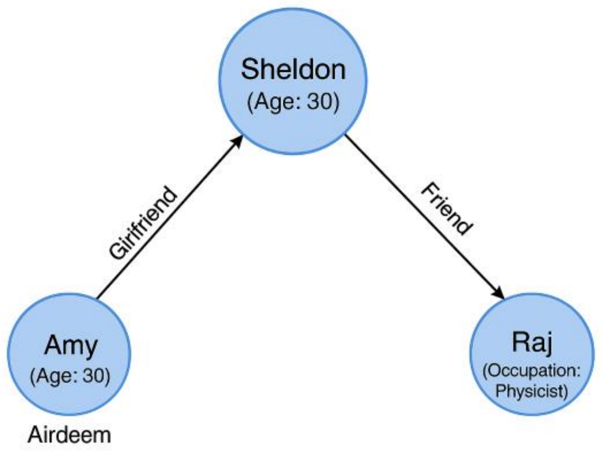
**Example :**

```
ID: iPad
{
"Type": "Tablet",
"Apps": ["Safari", "Facetime"]
}
```

## 3. Graph Database

- Examples: Neo4j, Amazon Neptune
- Data stored in nodes and edges representing parent-child relationships.

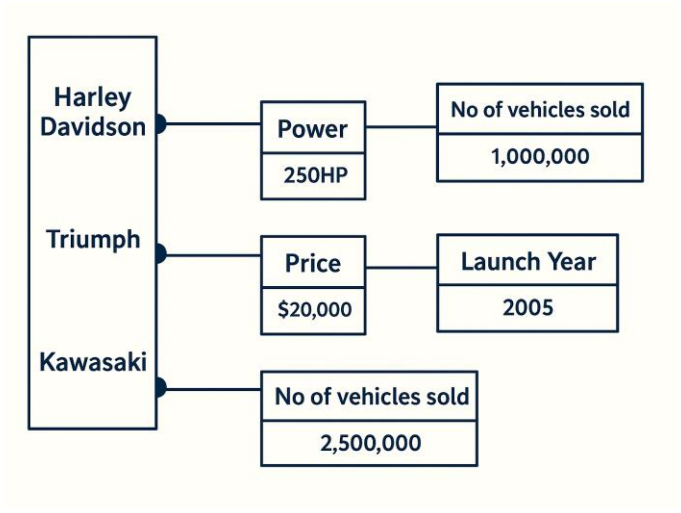
**Example:**



## 4. Wide Column Database (Column-Family Store)

- Examples: Apache Cassandra, Apache HBase
- Data is stored in columns rather than rows, suitable for large amounts of data with varying attributes.

**Example :**



**Applications of DBMS :-**

- Banking system
- Telecom
- Airlines
- Online shopping
- Educational institutions
- Manufacturing

## SQL Language or Types of SQL Commands :-

1. **DDL — Data Definition Language**

- Used to define/change structure of tables
- Commands:
  - CREATE
  - ALTER
  - DROP
  - TRUNCATE

2. **DCL — Data Control Language**

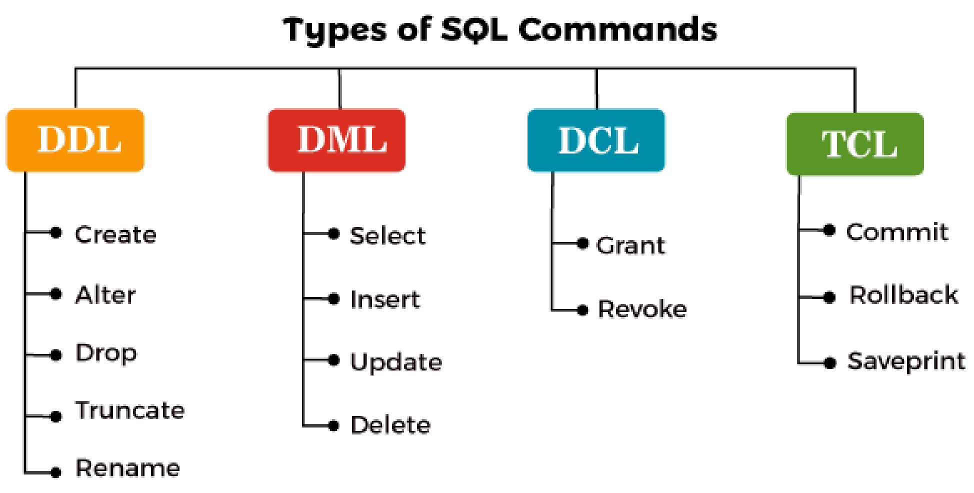
- Used to control access/permissions
- Commands:
  - GRANT
  - REVOKE

## 3. **DML — Data Manipulation Language**

- Used to manipulate data inside tables
- Commands:
  - SELECT
  - INSERT
  - UPDATE
  - DELETE

## 4. **TCL — Transaction Control Language**

- Used to manage transactions
- Commands:
  - COMMIT
  - ROLLBACK
  - SAVEPOINT
  - SET TRANSACTION



- **Most popular DB:** MySQL (widely used worldwide)

- Tools: MySQL Workbench, information_schema, mysql, performance_schema, sys
- **Schema:** A schema is a collection of tables & database objects

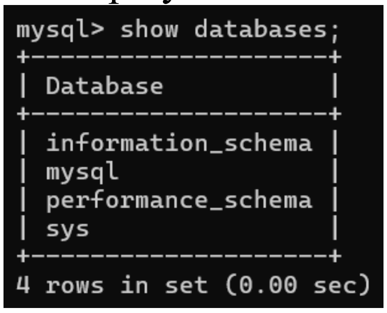
## How to Connect to SQL Server :-

**Three ways:**

1. **Using MySQL** — enter root password.
2. **Using Workbench** — connect to the table with GUI.
3. **Using Command Prompt (CMD):**
   - Command: mysql -u root -p

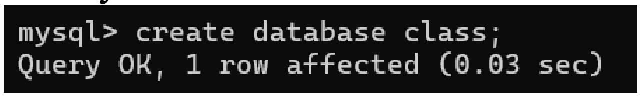
## MYSQL COMMANDS :-

➢ To display a list of all databases available in MySQL server.



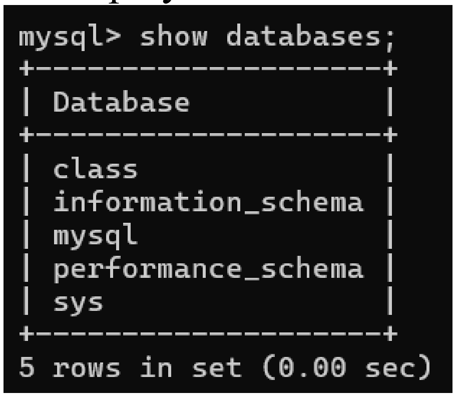
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

➢ Now you can create tables inside class database.



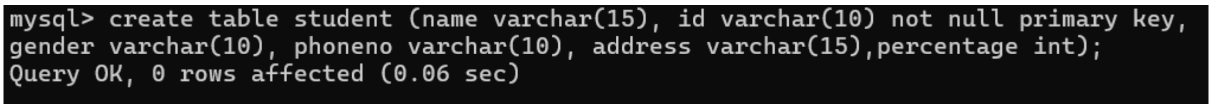
```
mysql> create database class;
Query OK, 1 row affected (0.03 sec)
```

➢ To display a list of all databases available on server.



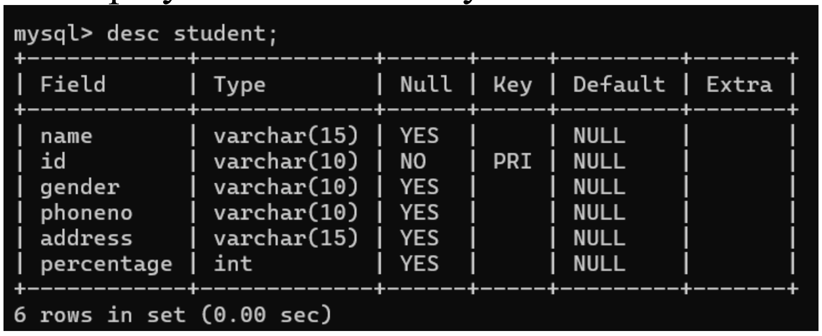
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| class              |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

➢ To creates a table named student.



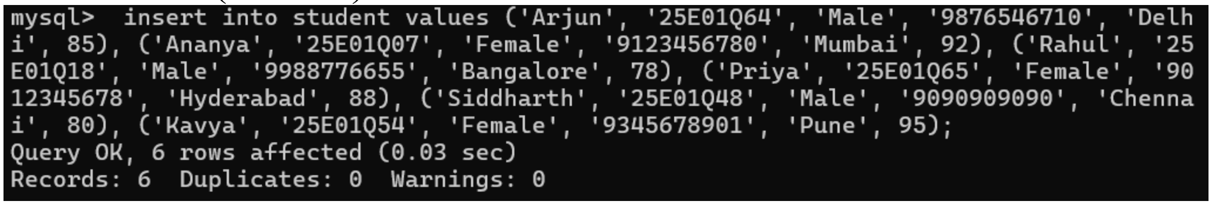
```
mysql> create table student (name varchar(15), id varchar(10) not null primary key,
gender varchar(10), phoneno varchar(10), address varchar(15),percentage int);
Query OK, 0 rows affected (0.06 sec)
```

➢ To display the structure of your student table**.**



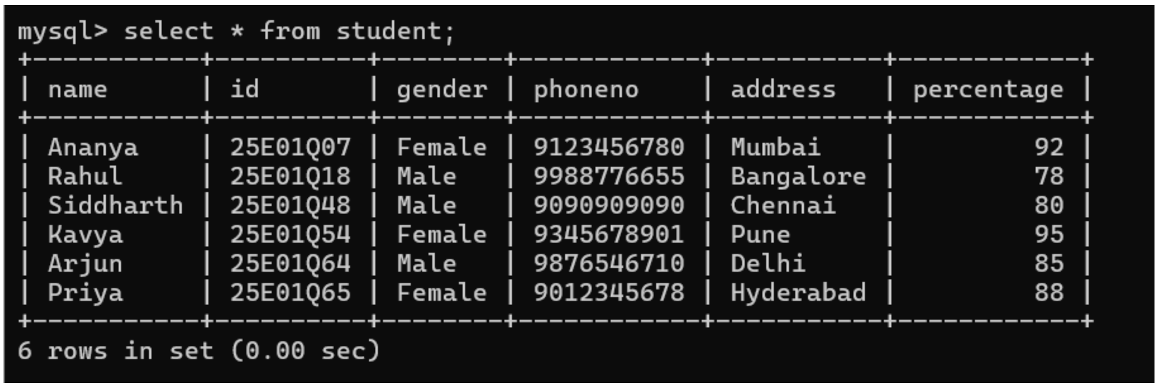
```
mysql> desc student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| name       | varchar(15) | YES  |     | NULL    |       |
| id         | varchar(10) | NO   | PRI | NULL    |       |
| gender     | varchar(10) | YES  |     | NULL    |       |
| phoneno    | varchar(10) | YES  |     | NULL    |       |
| address    | varchar(15) | YES  |     | NULL    |       |
| percentage | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

➢ To add data (records) into the table.



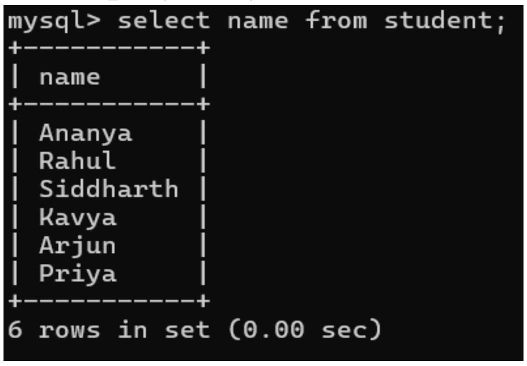
```
mysql>  insert into student values ('Arjun', '25E01Q64', 'Male', '9876546710', 'Delh
i', 85), ('Ananya', '25E01Q07', 'Female', '9123456780', 'Mumbai', 92), ('Rahul', '25
E01Q18', 'Male', '9988776655', 'Bangalore', 78), ('Priya', '25E01Q65', 'Female', '90
12345678', 'Hyderabad', 88), ('Siddharth', '25E01Q48', 'Male', '9090909090', 'Chenna
i', 80), ('Kavya', '25E01Q54', 'Female', '9345678901', 'Pune', 95);
Query OK, 6 rows affected (0.03 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

➢ To display all the records in your student table.



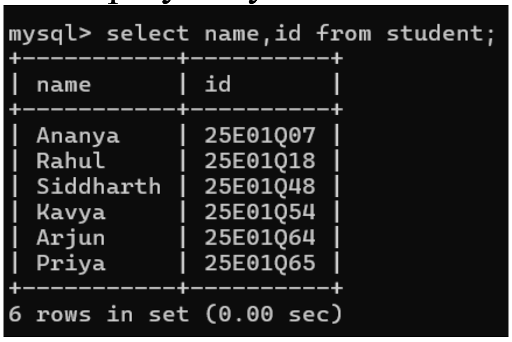
```
mysql> select * from student;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

- To display only the name column from your student table.



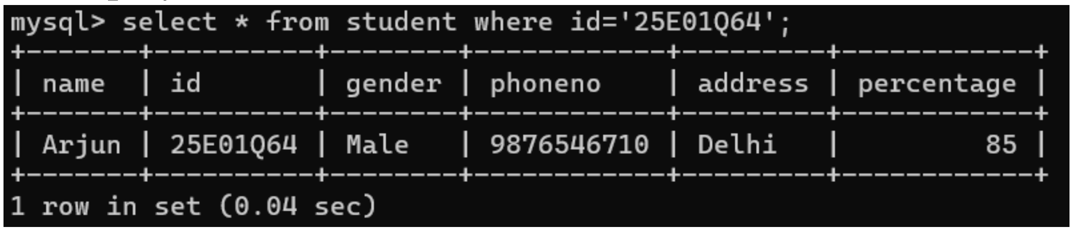
```
mysql> select name from student;
+-----------+
| name      |
+-----------+
| Ananya    |
| Rahul     |
| Siddharth |
| Kavya     |
| Arjun     |
| Priya     |
+-----------+
6 rows in set (0.00 sec)
```

- To display only the name and id columns from your student table.



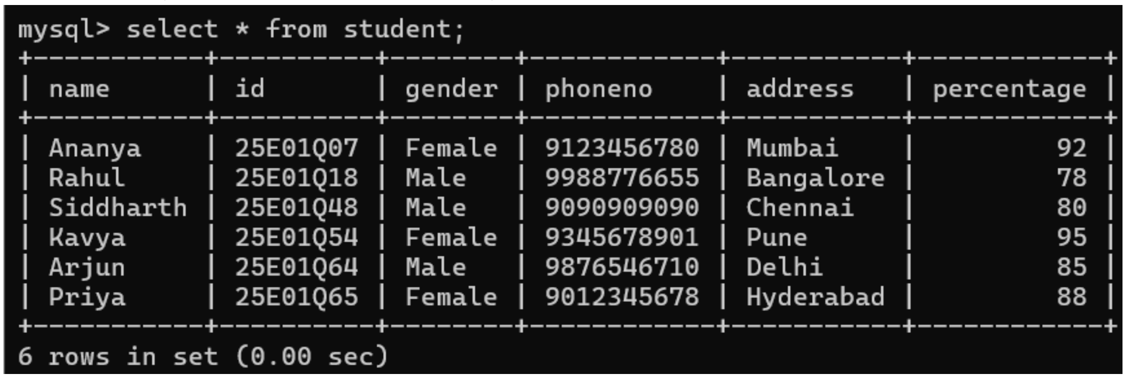
```
mysql> select name,id from student;
+-----------+----------+
| name      | id       |
+-----------+----------+
| Ananya    | 25E01Q07 |
| Rahul     | 25E01Q18 |
| Siddharth | 25E01Q48 |
| Kavya     | 25E01Q54 |
| Arjun     | 25E01Q64 |
| Priya     | 25E01Q65 |
+-----------+----------+
6 rows in set (0.00 sec)
```

- To display the record of the student whose id is '25E01Q64'



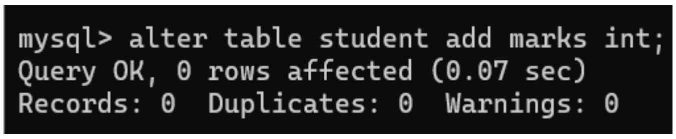
```
mysql> select * from student where id='25E01Q64';
+-------+----------+--------+------------+---------+------------+
| name  | id       | gender | phoneno    | address | percentage |
+-------+----------+--------+------------+---------+------------+
| Arjun | 25E01Q64 | Male   | 9876546710 | Delhi   |         85 |
+-------+----------+--------+------------+---------+------------+
1 row in set (0.04 sec)
```

- To display all the records in your student table



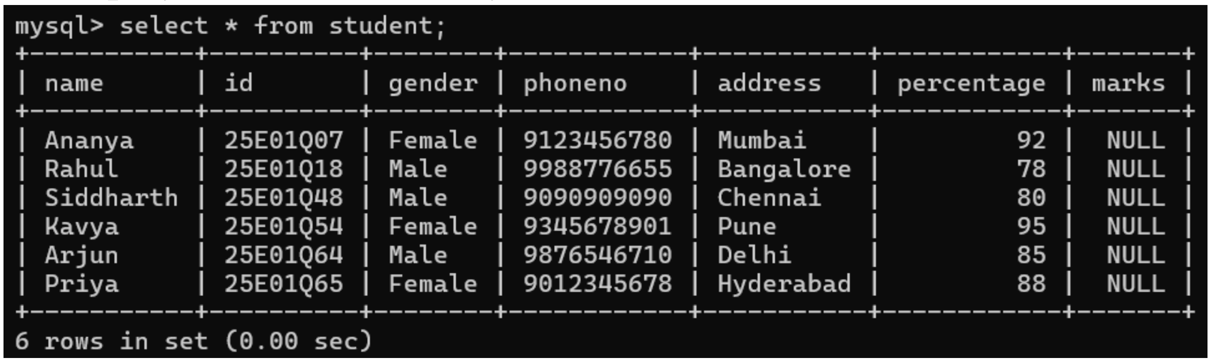
```
mysql> select * from student;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

- add a new column named marks of type INT to your existing student table.



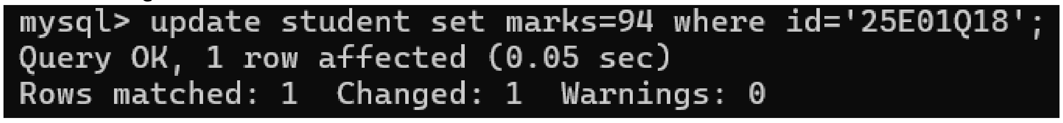
```
mysql> alter table student add marks int;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

➢ To display all the records in your student table.



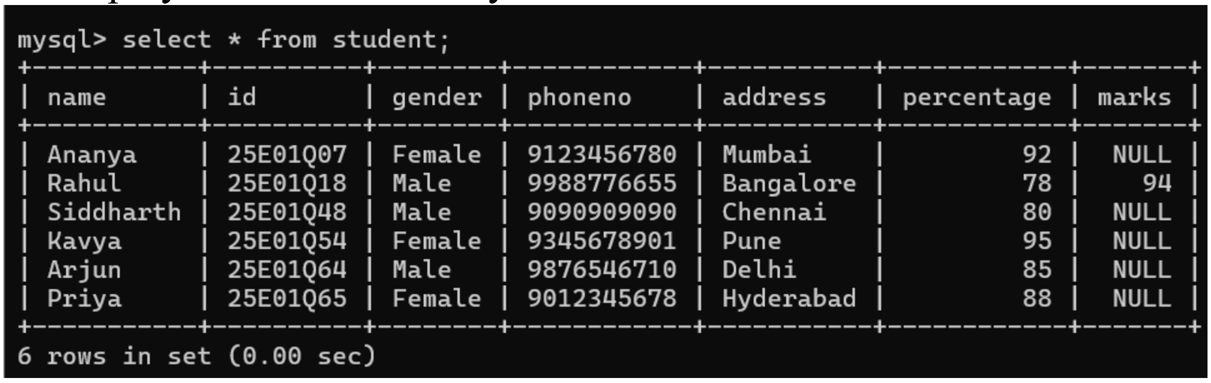
```
mysql> select * from student;
+-----------+----------+--------+------------+-----------+------------+-------+
| name      | id       | gender | phoneno    | address   | percentage | marks |
+-----------+----------+--------+------------+-----------+------------+-------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |  NULL |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |  NULL |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |  NULL |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |  NULL |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |  NULL |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |  NULL |
+-----------+----------+--------+------------+-----------+------------+-------+
6 rows in set (0.00 sec)
```

➢ update the marks column to 94 for the student whose id is '25E01Q18'.



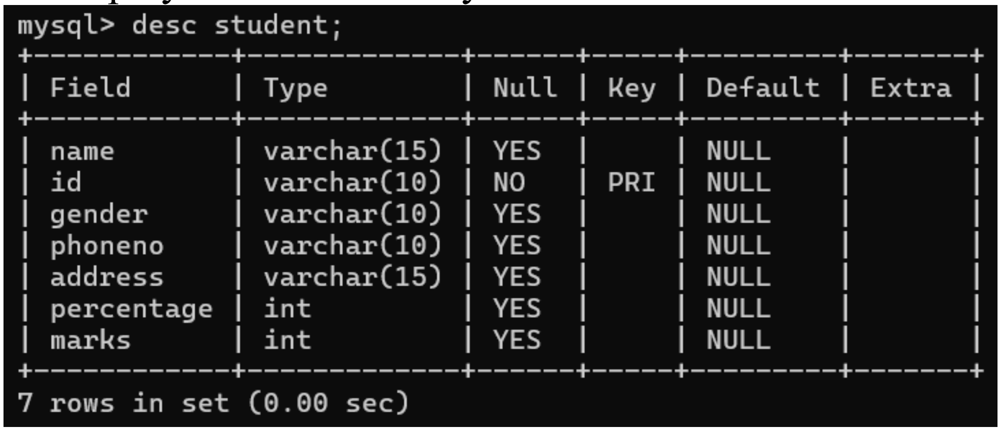
```
mysql> update student set marks=94 where id='25E01Q18';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

➢ To display all the records in your student table



```
mysql> select * from student;
+-----------+----------+--------+------------+-----------+------------+-------+
| name      | id       | gender | phoneno    | address   | percentage | marks |
+-----------+----------+--------+------------+-----------+------------+-------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |  NULL |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |    94 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |  NULL |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |  NULL |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |  NULL |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |  NULL |
+-----------+----------+--------+------------+-----------+------------+-------+
6 rows in set (0.00 sec)
```

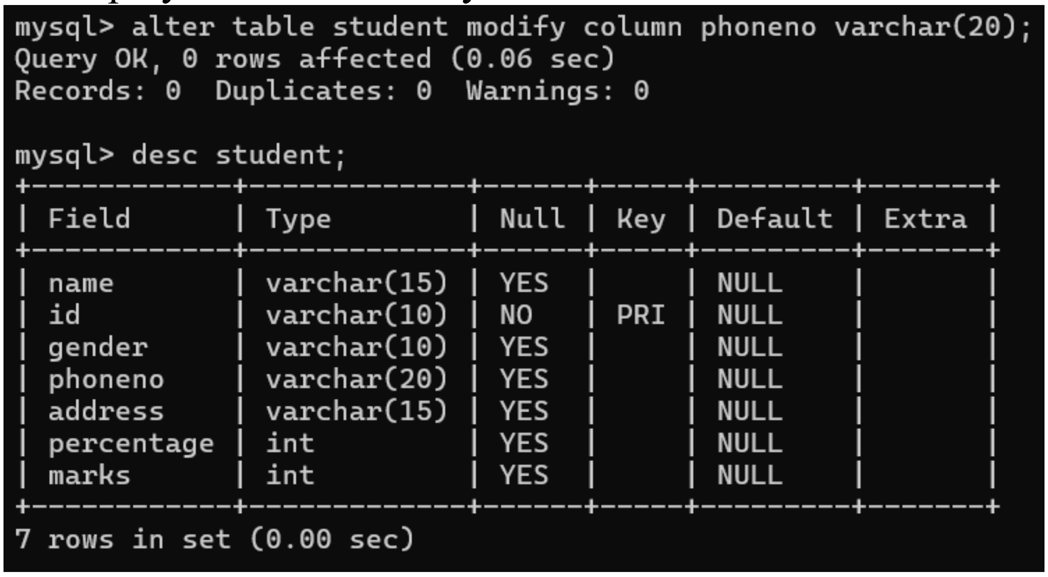
➢ To display the structure of your student table.



```
mysql> desc student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| name       | varchar(15) | YES  |     | NULL    |       |
| id         | varchar(10) | NO   | PRI | NULL    |       |
| gender     | varchar(10) | YES  |     | NULL    |       |
| phoneno    | varchar(10) | YES  |     | NULL    |       |
| address    | varchar(15) | YES  |     | NULL    |       |
| percentage | int         | YES  |     | NULL    |       |
| marks      | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

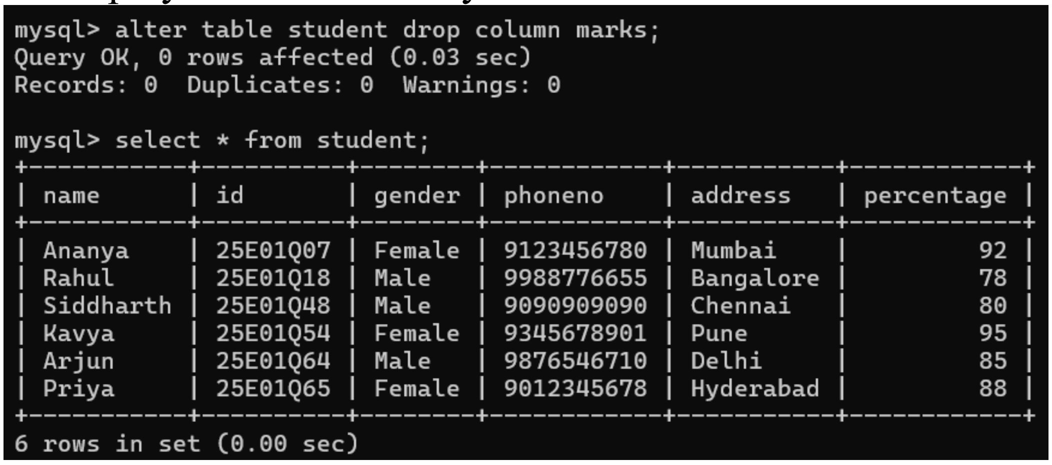
➢ The phoneno column data type was changed from VARCHAR(10) to VARCHAR(20).

- To display the structure of your student table.



```
mysql> alter table student modify column phoneno varchar(20);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| name       | varchar(15) | YES  |     | NULL    |       |
| id         | varchar(10) | NO   | PRI | NULL    |       |
| gender     | varchar(10) | YES  |     | NULL    |       |
| phoneno    | varchar(20) | YES  |     | NULL    |       |
| address    | varchar(15) | YES  |     | NULL    |       |
| percentage | int         | YES  |     | NULL    |       |
| marks      | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

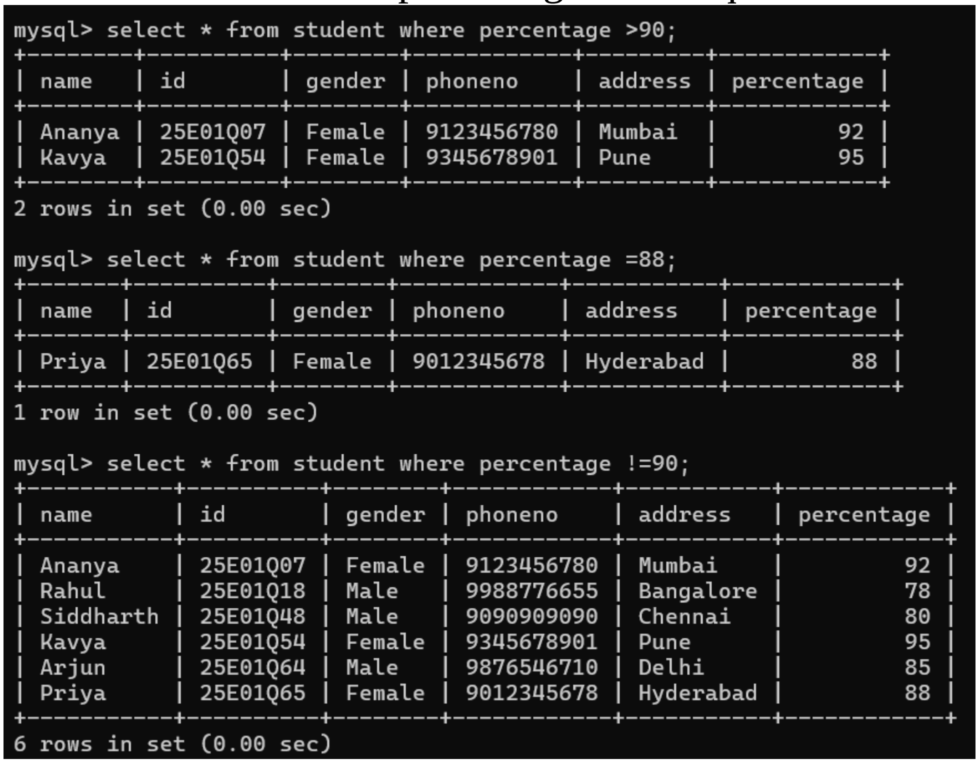
- The marks column was deleted from your student table.
- To display the structure of your student table.



```
mysql> alter table student drop column marks;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from student;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

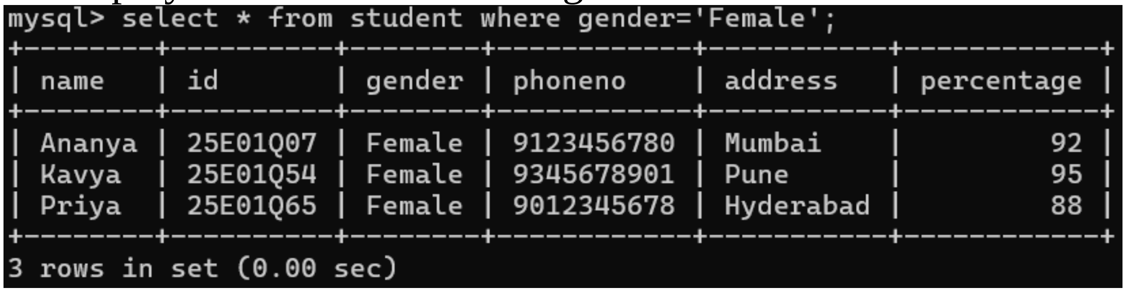
- Shows students having percentage greater than 90.
- Shows students having exactly 88%.

➢ Shows students whose percentage is not equal to 90.



```
mysql> select * from student where percentage >90;
+--------+----------+--------+------------+---------+------------+
| name   | id       | gender | phoneno    | address | percentage |
+--------+----------+--------+------------+---------+------------+
| Ananya | 25E01Q07 | Female | 9123456780 | Mumbai  |         92 |
| Kavya  | 25E01Q54 | Female | 9345678901 | Pune    |         95 |
+--------+----------+--------+------------+---------+------------+
2 rows in set (0.00 sec)

mysql> select * from student where percentage =88;
+-------+----------+--------+------------+-----------+------------+
| name  | id       | gender | phoneno    | address   | percentage |
+-------+----------+--------+------------+-----------+------------+
| Priya | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+-------+----------+--------+------------+-----------+------------+
1 row in set (0.00 sec)

mysql> select * from student where percentage !=90;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

➢ To display all records where the gender is 'Female'.



```
mysql> select * from student where gender='Female';
+--------+----------+--------+------------+-----------+------------+
| name   | id       | gender | phoneno    | address   | percentage |
+--------+----------+--------+------------+-----------+------------+
| Ananya | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Kavya  | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Priya  | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+--------+----------+--------+------------+-----------+------------+
3 rows in set (0.00 sec)
```

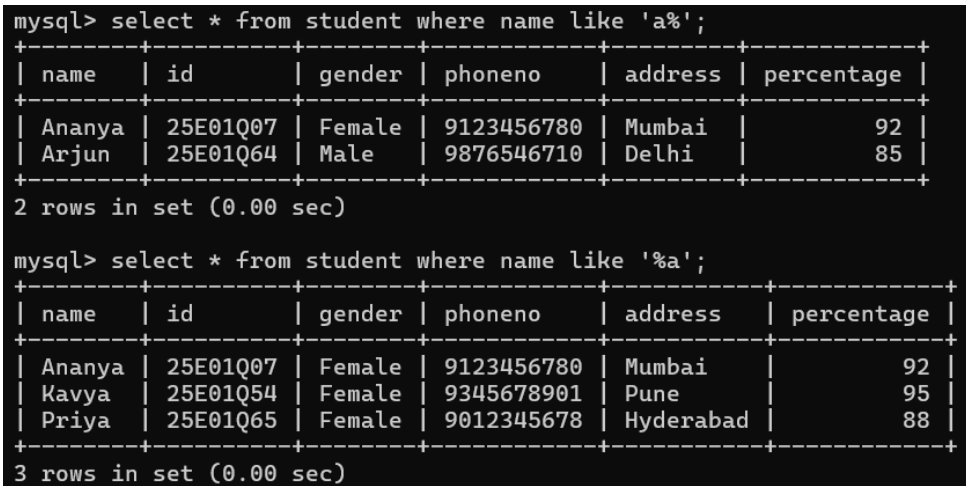
## Wild Card Characters :-

- In organization records are plenty. If you want to pull out the data we will use wildcard characters.
- The LIKE operator is used to compare a value to similar values using wildcard operators to filter records based on patterns.

Supported Wildcard Operators :-

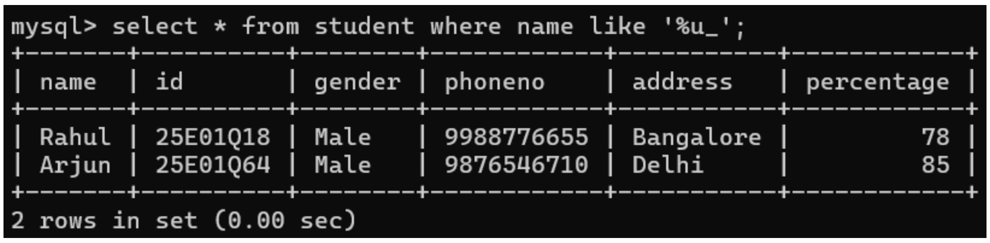
| Wildcard | Description |
|---|---|
| % | Matches **zero, one, or multiple characters**. Example: `'a%'` finds any value starting with 'a'. MS Access uses `*` instead of `%`. |
| _ | Matches **exactly one character**. Example: `'_a%'` finds any value with 'a' as the second character. MS Access uses `?` instead of `_`. |

- This shows all students whose names start with 'a' or 'A' (depending on collation).
- This shows all students whose names end with 'a'.



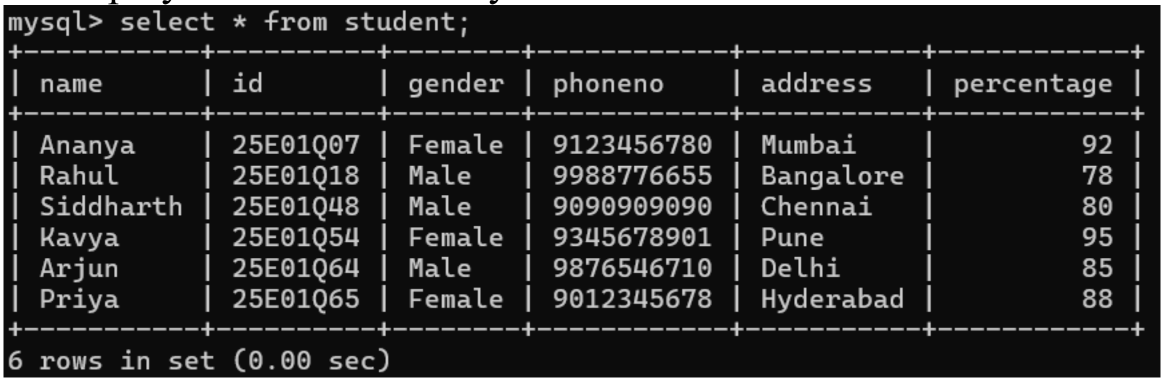
```
mysql> select * from student where name like 'a%';
+--------+----------+--------+------------+---------+------------+
| name   | id       | gender | phoneno    | address | percentage |
+--------+----------+--------+------------+---------+------------+
| Ananya | 25E01Q07 | Female | 9123456780 | Mumbai  |         92 |
| Arjun  | 25E01Q64 | Male   | 9876546710 | Delhi   |         85 |
+--------+----------+--------+------------+---------+------------+
2 rows in set (0.00 sec)

mysql> select * from student where name like '%a';
+--------+----------+--------+------------+-----------+------------+
| name   | id       | gender | phoneno    | address   | percentage |
+--------+----------+--------+------------+-----------+------------+
| Ananya | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Kavya  | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Priya  | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+--------+----------+--------+------------+-----------+------------+
3 rows in set (0.00 sec)
```

- To display all records where the second letter of the name is 'a'.
- To display all records where the name ends with any character that has 'u'.



```
mysql> select * from student where name like '%u_';
+-------+----------+--------+------------+-----------+------------+
| name  | id       | gender | phoneno    | address   | percentage |
+-------+----------+--------+------------+-----------+------------+
| Rahul | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Arjun | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
+-------+----------+--------+------------+-----------+------------+
2 rows in set (0.00 sec)
```

- To display all the records in your student table.



```
mysql> select * from student;
+------------+----------+--------+------------+-----------+------------+
| name       | id       | gender | phoneno    | address   | percentage |
+------------+----------+--------+------------+-----------+------------+
| Ananya     | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Rahul      | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth  | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Kavya      | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Arjun      | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya      | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+------------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

## SQL FUNCTIONS:

### 1. Aggregate Functions

- SUM() – total sum of a column
- AVG() – average value
- COUNT() – number of rows
- MAX() – highest value
- MIN() – lowest value

### 2. String Functions

- LENGTH() – length of string
- CHAR_LENGTH() – number of characters in string
- UPPER() / UCASE() – convert to uppercase
- LOWER() / LCASE() – convert to lowercase
- SUBSTRING() / SUBSTR() – extract part of string
- CONCAT() – combine strings
- TRIM() – remove spaces from both ends
- LTRIM() – remove spaces from left
- RTRIM() – remove spaces from right
- REPLACE() – replace part of string with another
- INSTR() – position of substring
- REVERSE() – reverses string

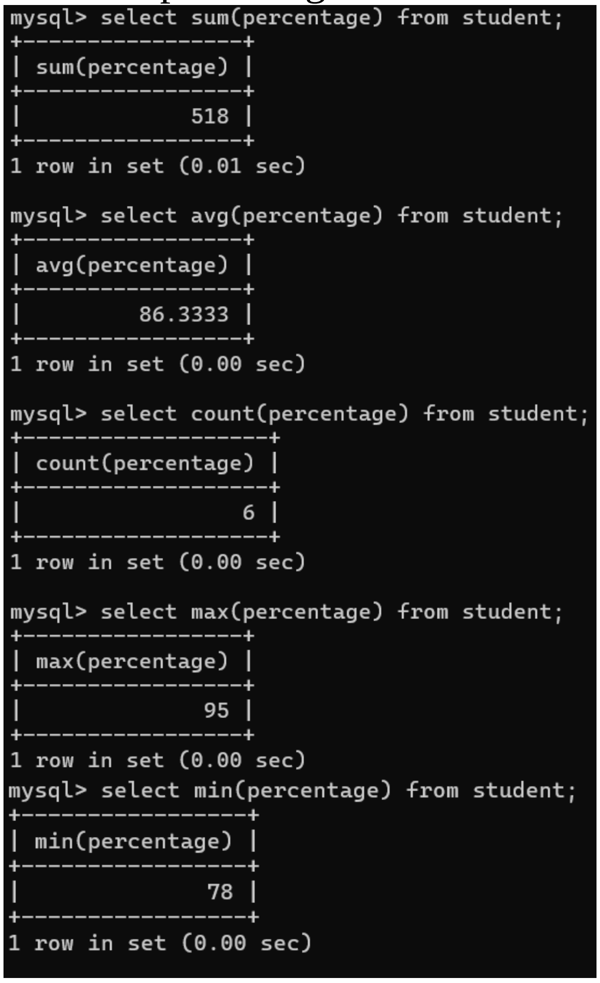
### 3. Date and Time Functions

- NOW() – current date and time
- CURDATE() – current date
- CURTIME() – current time
- DAY() – day from date
- MONTH() – month from date
- YEAR() – year from date
- DAYNAME() – name of weekday
- MONTHNAME() – name of month
- DATEDIFF() – difference between two dates
- DATE_ADD() – add to date
- DATE_SUB() – subtract from date
- TIME() – extract time part

## 4. Mathematical Functions

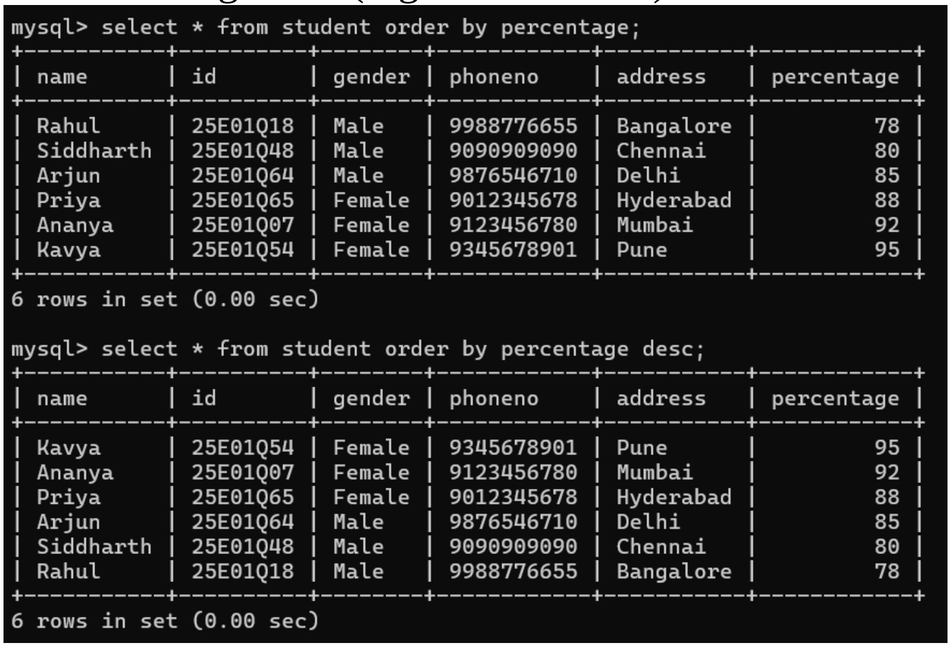
- ROUND() – rounds a number
- CEIL() / CEILING() – smallest integer >= number
- FLOOR() – largest integer <= number
- ABS() – absolute value
- MOD() – remainder
- POWER() – x to the power y
- SQRT() – square root
- EXP() – exponential value of x
- LOG() – natural log
- RAND() – random number

➢ Adds up all the percentage values.
➢ Calculates the average of all percentage values.
➢ Counts the number of entries in the percentage column.
➢ Finds the highest percentage value.the minimum (smallest) value from the percentage column in the student table



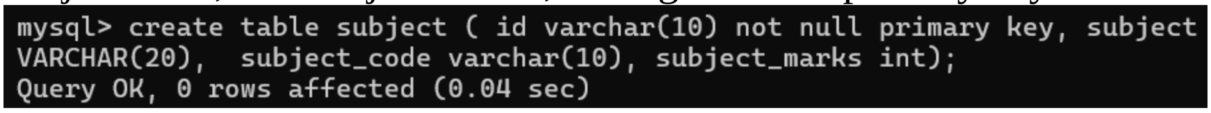
```
mysql> select sum(percentage) from student;
+-----------------+
| sum(percentage) |
+-----------------+
|             518 |
+-----------------+
1 row in set (0.01 sec)

mysql> select avg(percentage) from student;
+-----------------+
| avg(percentage) |
+-----------------+
|         86.3333 |
+-----------------+
1 row in set (0.00 sec)

mysql> select count(percentage) from student;
+-------------------+
| count(percentage) |
+-------------------+
|                 6 |
+-------------------+
1 row in set (0.00 sec)

mysql> select max(percentage) from student;
+-----------------+
| max(percentage) |
+-----------------+
|              95 |
+-----------------+
1 row in set (0.00 sec)
mysql> select min(percentage) from student;
+-----------------+
| min(percentage) |
+-----------------+
|              78 |
+-----------------+
1 row in set (0.00 sec)
```

- To displays all records from the student table sorted by percentage in ascending order (lowest to highest).
- To displays all records from the student table sorted by percentage in descending order (highest to lowest).



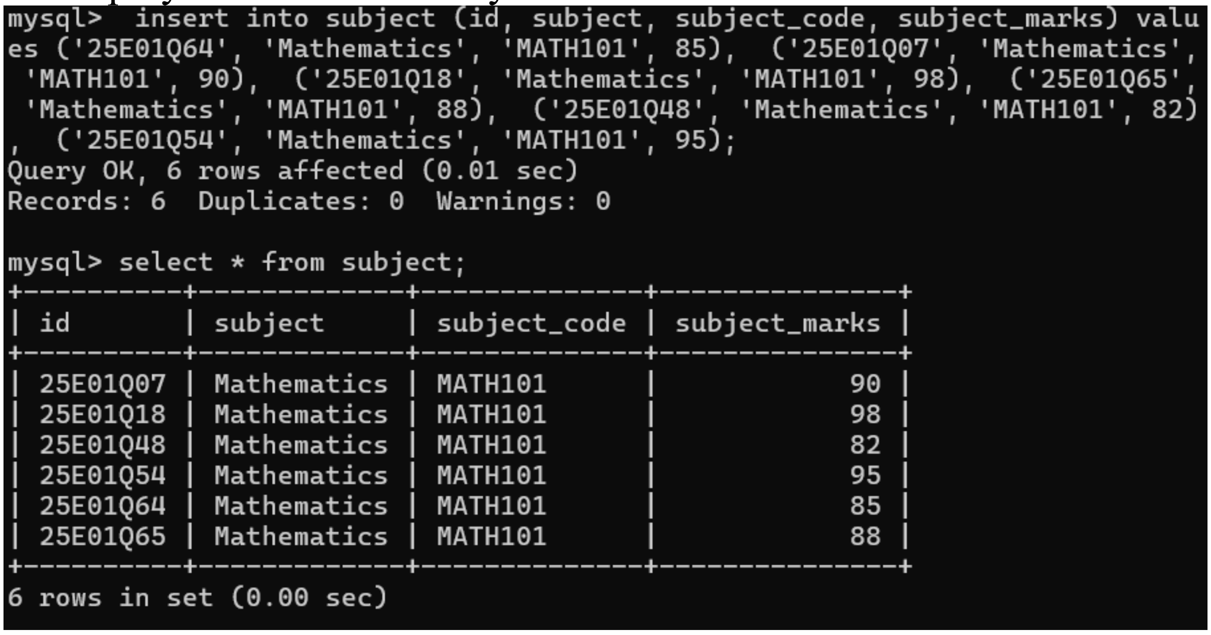
```
mysql> select * from student order by percentage;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)

mysql> select * from student order by percentage desc;
+-----------+----------+--------+------------+-----------+------------+
| name      | id       | gender | phoneno    | address   | percentage |
+-----------+----------+--------+------------+-----------+------------+
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

- creates a subject table with columns for student id, subject name, subject code, and subject marks, setting id as the primary key.



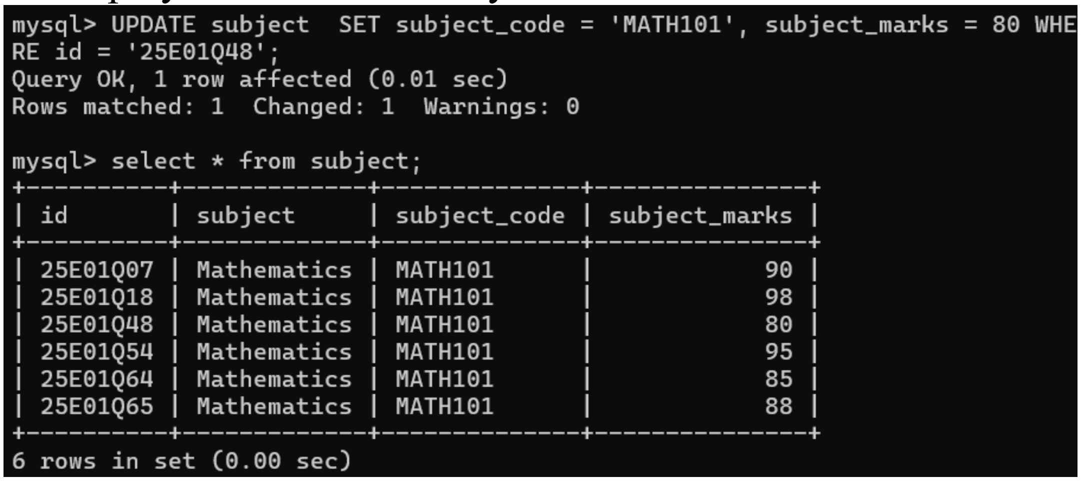
```
mysql> create table subject ( id varchar(10) not null primary key, subject
VARCHAR(20),  subject_code varchar(10), subject_marks int);
Query OK, 0 rows affected (0.04 sec)
```

- Inserted 6 rows into the subject table with student IDs, Mathematics subject, and their marks successfully.
- To display all the records in your student table.



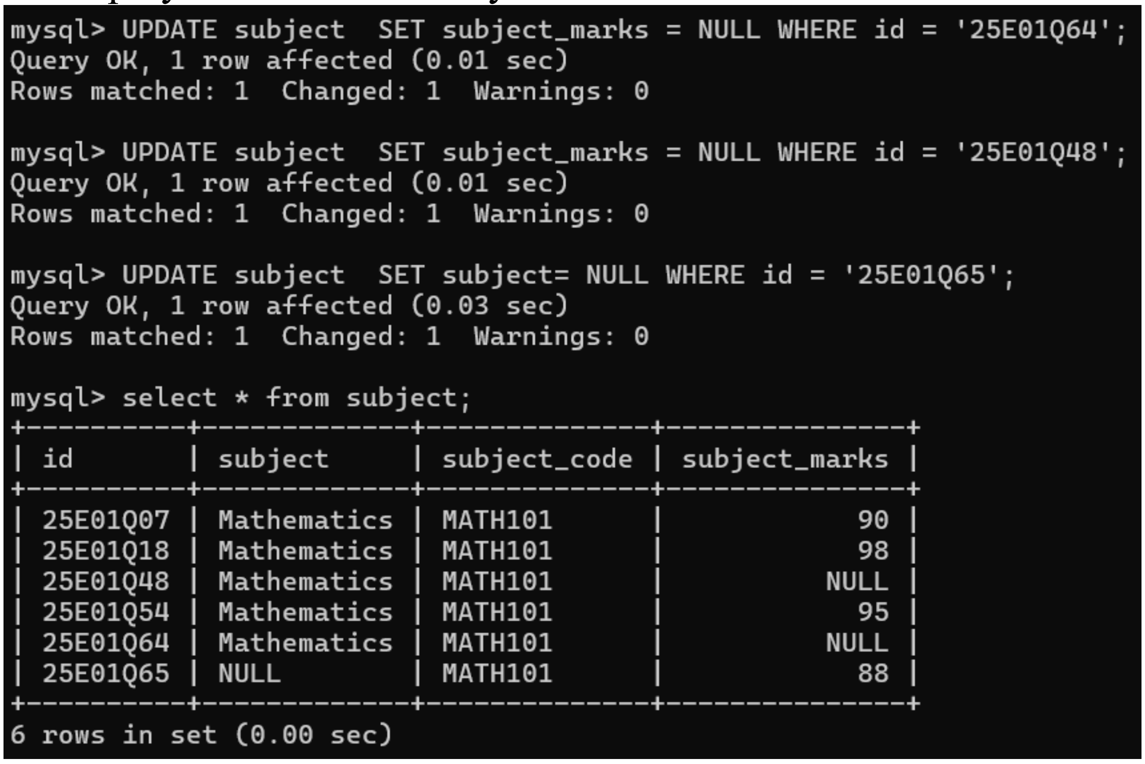
```
mysql>  insert into subject (id, subject, subject_code, subject_marks) valu
es ('25E01Q64', 'Mathematics', 'MATH101', 85),  ('25E01Q07', 'Mathematics',
 'MATH101', 90),  ('25E01Q18', 'Mathematics', 'MATH101', 98),  ('25E01Q65',
 'Mathematics', 'MATH101', 88),  ('25E01Q48', 'Mathematics', 'MATH101', 82)
,  ('25E01Q54', 'Mathematics', 'MATH101', 95);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from subject;
+----------+-------------+--------------+---------------+
| id       | subject     | subject_code | subject_marks |
+----------+-------------+--------------+---------------+
| 25E01Q07 | Mathematics | MATH101      |            90 |
| 25E01Q18 | Mathematics | MATH101      |            98 |
| 25E01Q48 | Mathematics | MATH101      |            82 |
| 25E01Q54 | Mathematics | MATH101      |            95 |
| 25E01Q64 | Mathematics | MATH101      |            85 |
| 25E01Q65 | Mathematics | MATH101      |            88 |
+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)
```

- updated the subject_code to 'MATH101' and subject_marks to 80 for the student with id '25E01Q48' in the subject table.
- To display all the records in your student table.



```
mysql> UPDATE subject  SET subject_code = 'MATH101', subject_marks = 80 WHE
RE id = '25E01Q48';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from subject;
+----------+-------------+--------------+---------------+
| id       | subject     | subject_code | subject_marks |
+----------+-------------+--------------+---------------+
| 25E01Q07 | Mathematics | MATH101      |            90 |
| 25E01Q18 | Mathematics | MATH101      |            98 |
| 25E01Q48 | Mathematics | MATH101      |            80 |
| 25E01Q54 | Mathematics | MATH101      |            95 |
| 25E01Q64 | Mathematics | MATH101      |            85 |
| 25E01Q65 | Mathematics | MATH101      |            88 |
+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)
```

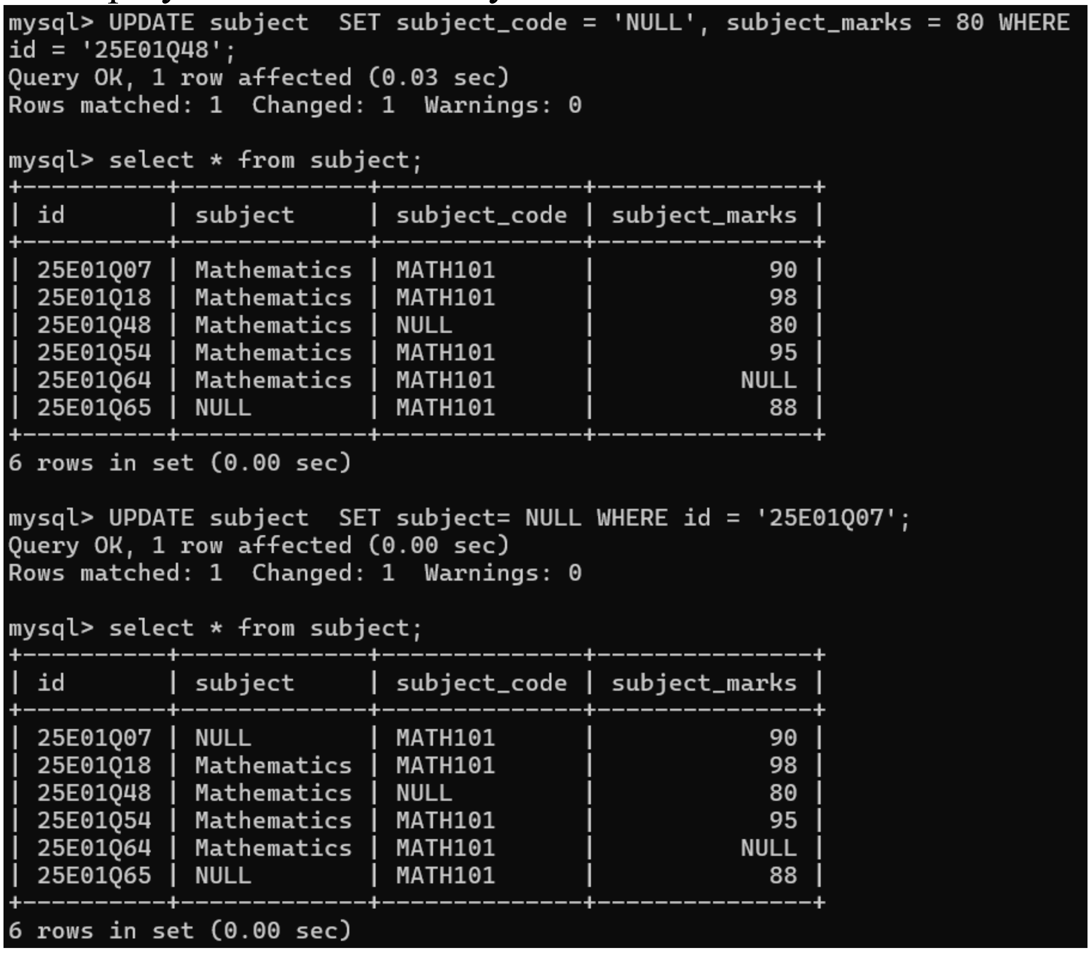
- Sets subject_marks to NULL for student ID 25E01Q64.
- Sets subject_marks to NULL for student ID 25E01Q48.
- Sets subject to NULL for student ID 25E01Q65.
- To display all the records in your student table.



```
mysql> UPDATE subject  SET subject_marks = NULL WHERE id = '25E01Q64';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE subject  SET subject_marks = NULL WHERE id = '25E01Q48';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE subject  SET subject= NULL WHERE id = '25E01Q65';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from subject;
+----------+-------------+--------------+---------------+
| id       | subject     | subject_code | subject_marks |
+----------+-------------+--------------+---------------+
| 25E01Q07 | Mathematics | MATH101      |            90 |
| 25E01Q18 | Mathematics | MATH101      |            98 |
| 25E01Q48 | Mathematics | MATH101      |          NULL |
| 25E01Q54 | Mathematics | MATH101      |            95 |
| 25E01Q64 | Mathematics | MATH101      |          NULL |
| 25E01Q65 | NULL        | MATH101      |            88 |
+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)
```

- set subject_code to the string 'NULL' (not actual NULL) and subject_marks to 80 for ID 25E01Q48.
- This sets subject column to NULL (no value) for ID 25E01Q07.

➢ To display all the records in your student table.



```
mysql> UPDATE subject  SET subject_code = 'NULL', subject_marks = 80 WHERE
id = '25E01Q48';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from subject;
+----------+-------------+--------------+---------------+
| id       | subject     | subject_code | subject_marks |
+----------+-------------+--------------+---------------+
| 25E01Q07 | Mathematics | MATH101      |            90 |
| 25E01Q18 | Mathematics | MATH101      |            98 |
| 25E01Q48 | Mathematics | NULL         |            80 |
| 25E01Q54 | Mathematics | MATH101      |            95 |
| 25E01Q64 | Mathematics | MATH101      |          NULL |
| 25E01Q65 | NULL        | MATH101      |            88 |
+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)

mysql> UPDATE subject  SET subject= NULL WHERE id = '25E01Q07';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from subject;
+----------+-------------+--------------+---------------+
| id       | subject     | subject_code | subject_marks |
+----------+-------------+--------------+---------------+
| 25E01Q07 | NULL        | MATH101      |            90 |
| 25E01Q18 | Mathematics | MATH101      |            98 |
| 25E01Q48 | Mathematics | NULL         |            80 |
| 25E01Q54 | Mathematics | MATH101      |            95 |
| 25E01Q64 | Mathematics | MATH101      |          NULL |
| 25E01Q65 | NULL        | MATH101      |            88 |
+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)
```

## SQL Joins Descriptions :-

### 1. INNER JOIN

- Returns only matching rows from both tables.
- Excludes non-matching rows.
- Shows common data between tables.

### 2. LEFT JOIN (LEFT OUTER JOIN)

- Returns all rows from the left table.
- Returns matching rows from the right table.
- Shows NULL for right table columns if no match found.

## 3. RIGHT JOIN (RIGHT OUTER JOIN)

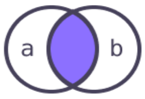
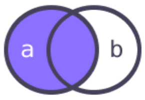
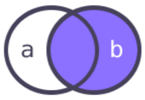
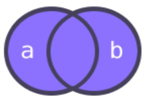
- Returns all rows from the right table.
- Returns matching rows from the left table.
- Shows NULL for left table columns if no match found.

## 4. FULL JOIN (FULL OUTER JOIN)

- Returns all rows from both tables.
- Combines matching rows.
- Shows NULL where there is no match in either table.

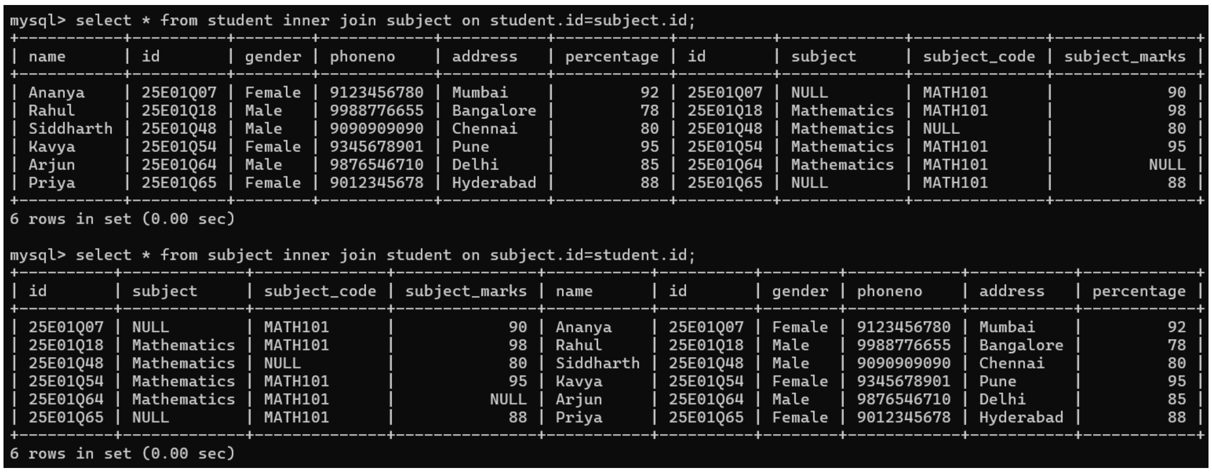
## 5. CROSS JOIN

- Returns the Cartesian product of both tables.
- Every row of first table combines with every row of second table.
- Number of rows in result = rows in table1 * rows in table2.

| Join type | Visually | Example usage |
|---|---|---|
| Inner join |  | `a JOIN b ON a.id = b.id` |
| Left join |  | `a LEFT JOIN b ON a.id = b.id` |
| Right join |  | `a RIGHT JOIN b ON a.id = b.id` |
| Full outer join |  | `a FULL OUTER JOIN b ON a.id = b.id` |

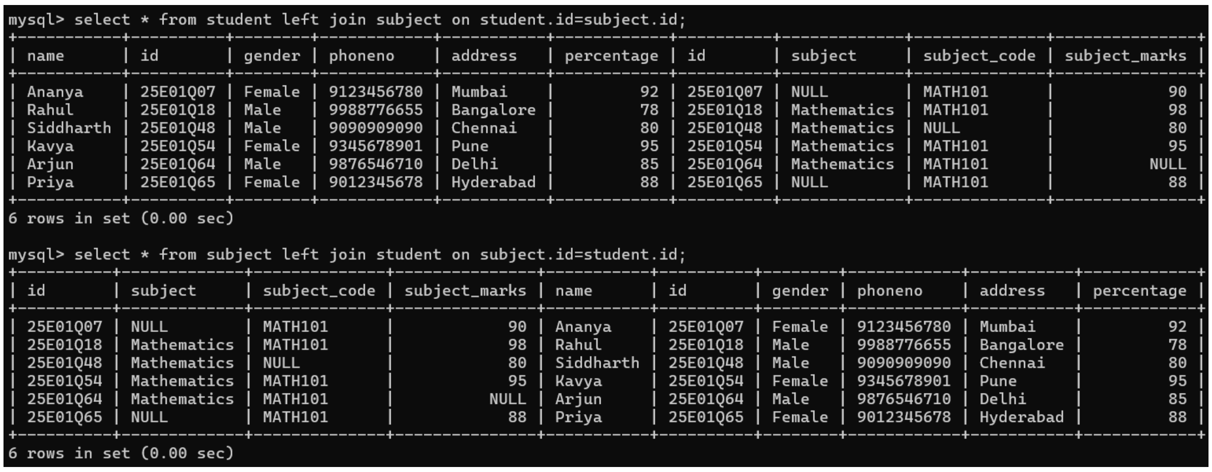
- ➢ INNER JOIN combines rows from both tables only where the id matches in both student and subject tables.
- ➢ It returns only the records with common id values in both tables.

- INNER JOIN combines rows from subject and student where subject.id = student.id.
- The output columns start with the subject table’s columns followed by student table’s columns (because subject is written first)



```
mysql> select * from student inner join subject on student.id=subject.id;
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
| name      | id       | gender | phoneno    | address   | percentage | id       | subject     | subject_code | subject_marks |
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 | 25E01Q07 | NULL        | MATH101      |            90 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 | 25E01Q18 | Mathematics | MATH101      |            98 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 | 25E01Q48 | Mathematics | NULL         |            80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 | 25E01Q54 | Mathematics | MATH101      |            95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 | 25E01Q64 | Mathematics | MATH101      |          NULL |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 | 25E01Q65 | NULL        | MATH101      |            88 |
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)

mysql> select * from subject inner join student on subject.id=student.id;
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
| id       | subject     | subject_code | subject_marks | name      | id       | gender | phoneno    | address   | percentage |
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
| 25E01Q07 | NULL        | MATH101      |            90 | Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| 25E01Q18 | Mathematics | MATH101      |            98 | Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| 25E01Q48 | Mathematics | NULL         |            80 | Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| 25E01Q54 | Mathematics | MATH101      |            95 | Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| 25E01Q64 | Mathematics | MATH101      |          NULL | Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| 25E01Q65 | NULL        | MATH101      |            88 | Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

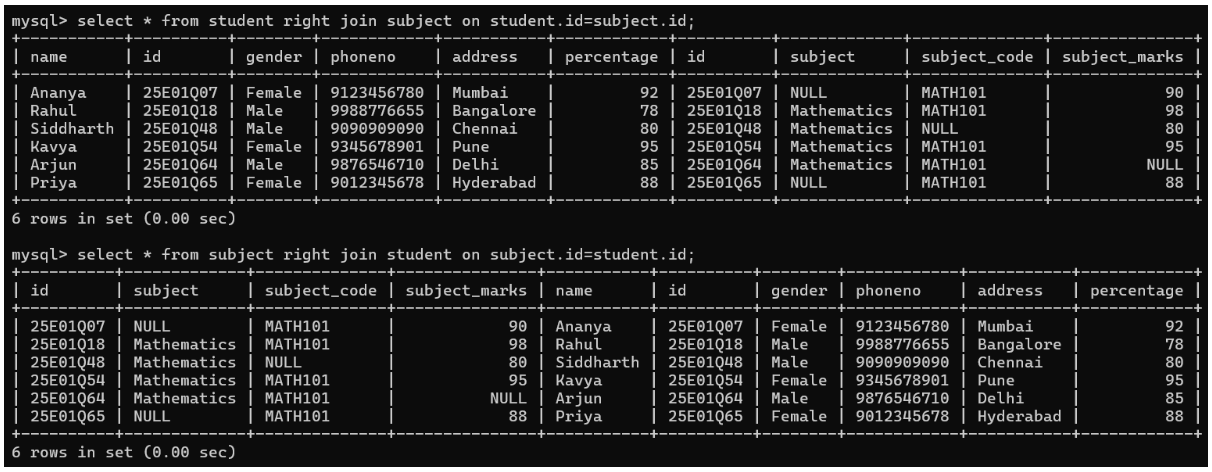
- LEFT JOIN returns all rows from the student table and matching rows from the subject table.
- If there is no matching id in subject, the result will show NULL for subject columns.
- LEFT JOIN returns all rows from the subject table and matching rows from the student table.
- If there is no matching id in student, the result will show NULL for student columns.



```
mysql> select * from student left join subject on student.id=subject.id;
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
| name      | id       | gender | phoneno    | address   | percentage | id       | subject     | subject_code | subject_marks |
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
| Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 | 25E01Q07 | NULL        | MATH101      |            90 |
| Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 | 25E01Q18 | Mathematics | MATH101      |            98 |
| Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 | 25E01Q48 | Mathematics | NULL         |            80 |
| Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 | 25E01Q54 | Mathematics | MATH101      |            95 |
| Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 | 25E01Q64 | Mathematics | MATH101      |          NULL |
| Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 | 25E01Q65 | NULL        | MATH101      |            88 |
+-----------+----------+--------+------------+-----------+------------+----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)

mysql> select * from subject left join student on subject.id=student.id;
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
| id       | subject     | subject_code | subject_marks | name      | id       | gender | phoneno    | address   | percentage |
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
| 25E01Q07 | NULL        | MATH101      |            90 | Ananya    | 25E01Q07 | Female | 9123456780 | Mumbai    |         92 |
| 25E01Q18 | Mathematics | MATH101      |            98 | Rahul     | 25E01Q18 | Male   | 9988776655 | Bangalore |         78 |
| 25E01Q48 | Mathematics | NULL         |            80 | Siddharth | 25E01Q48 | Male   | 9090909090 | Chennai   |         80 |
| 25E01Q54 | Mathematics | MATH101      |            95 | Kavya     | 25E01Q54 | Female | 9345678901 | Pune      |         95 |
| 25E01Q64 | Mathematics | MATH101      |          NULL | Arjun     | 25E01Q64 | Male   | 9876546710 | Delhi     |         85 |
| 25E01Q65 | NULL        | MATH101      |            88 | Priya     | 25E01Q65 | Female | 9012345678 | Hyderabad |         88 |
+----------+-------------+--------------+---------------+-----------+----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

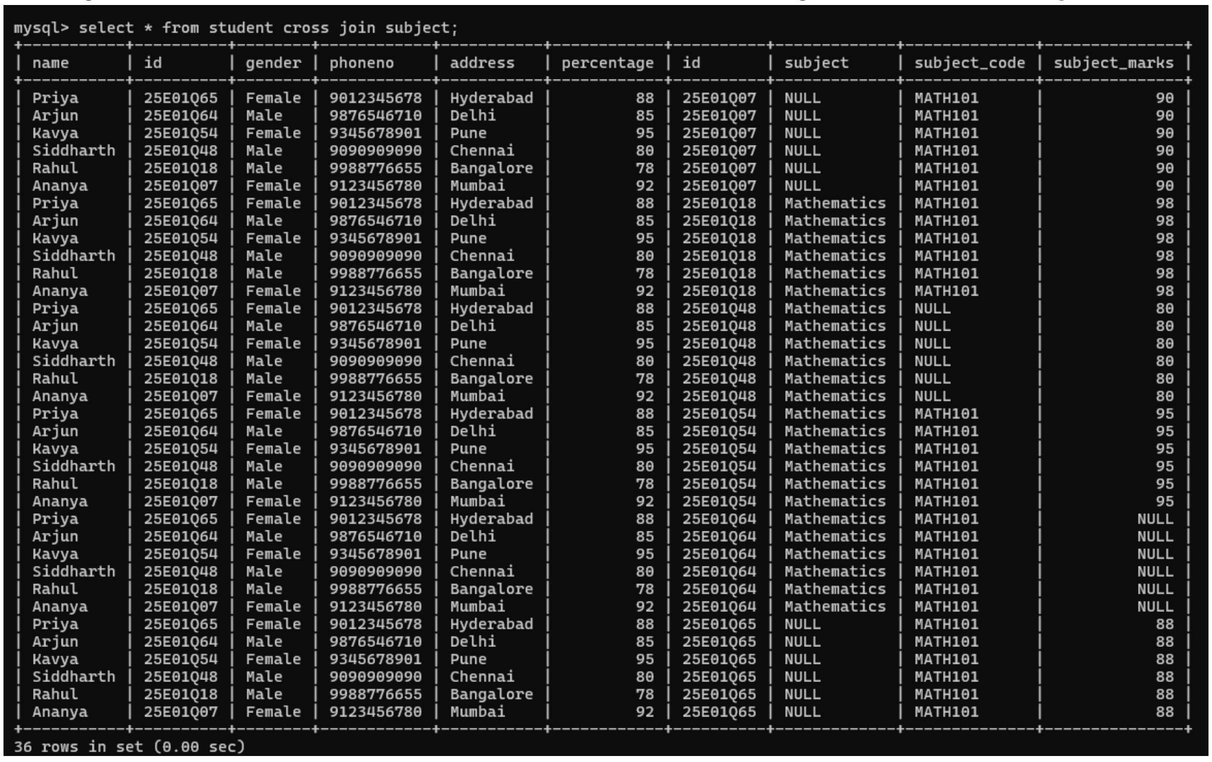
- RIGHT JOIN returns all rows from the subject table and matching rows from the student table.
- If there is no matching id in student, the result will show NULL for student columns.

- RIGHT JOIN returns all rows from the student table and matching rows from the subject table.
- If there is no matching id in subject, the result will show NULL for subject columns.



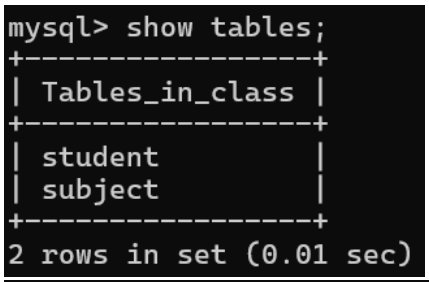
```
mysql> select * from student right join subject on student.id=subject.id;
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
| name      | id        | gender | phoneno    | address   | percentage | id        | subject     | subject_code | subject_marks |
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q07  | NULL        | MATH101      |            90 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q48  | Mathematics | NULL         |            80 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q65  | NULL        | MATH101      |            88 |
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
6 rows in set (0.00 sec)

mysql> select * from subject right join student on subject.id=student.id;
+-----------+-------------+--------------+---------------+-----------+-----------+--------+------------+-----------+------------+
| id        | subject     | subject_code | subject_marks | name      | id        | gender | phoneno    | address   | percentage |
+-----------+-------------+--------------+---------------+-----------+-----------+--------+------------+-----------+------------+
| 25E01Q07  | NULL        | MATH101      |            90 | Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 |
| 25E01Q18  | Mathematics | MATH101      |            98 | Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 |
| 25E01Q48  | Mathematics | NULL         |            80 | Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 |
| 25E01Q54  | Mathematics | MATH101      |            95 | Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 |
| 25E01Q64  | Mathematics | MATH101      |          NULL | Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 |
| 25E01Q65  | NULL        | MATH101      |            88 | Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 |
+-----------+-------------+--------------+---------------+-----------+-----------+--------+------------+-----------+------------+
6 rows in set (0.00 sec)
```

- CROSS JOIN returns the Cartesian product of the two tables.
- Every row from student is combined with every row from subject.

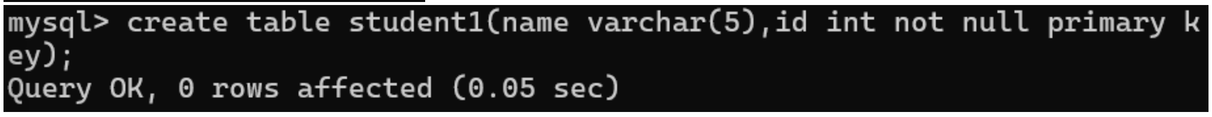


```
mysql> select * from student cross join subject;
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
| name      | id        | gender | phoneno    | address   | percentage | id        | subject     | subject_code | subject_marks |
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q07  | NULL        | MATH101      |            90 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q07  | NULL        | MATH101      |            90 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q07  | NULL        | MATH101      |            90 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q07  | NULL        | MATH101      |            90 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q07  | NULL        | MATH101      |            90 |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q07  | NULL        | MATH101      |            90 |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q18  | Mathematics | MATH101      |            98 |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q48  | Mathematics | NULL         |            80 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q48  | Mathematics | NULL         |            80 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q48  | Mathematics | NULL         |            80 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q48  | Mathematics | NULL         |            80 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q48  | Mathematics | NULL         |            80 |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q48  | Mathematics | NULL         |            80 |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q54  | Mathematics | MATH101      |            95 |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q64  | Mathematics | MATH101      |          NULL |
| Priya     | 25E01Q65  | Female | 9012345678 | Hyderabad |         88 | 25E01Q65  | NULL        | MATH101      |            88 |
| Arjun     | 25E01Q64  | Male   | 9876546710 | Delhi     |         85 | 25E01Q65  | NULL        | MATH101      |            88 |
| Kavya     | 25E01Q54  | Female | 9345678901 | Pune      |         95 | 25E01Q65  | NULL        | MATH101      |            88 |
| Siddharth | 25E01Q48  | Male   | 9090909090 | Chennai   |         80 | 25E01Q65  | NULL        | MATH101      |            88 |
| Rahul     | 25E01Q18  | Male   | 9988776655 | Bangalore |         78 | 25E01Q65  | NULL        | MATH101      |            88 |
| Ananya    | 25E01Q07  | Female | 9123456780 | Mumbai    |         92 | 25E01Q65  | NULL        | MATH101      |            88 |
+-----------+-----------+--------+------------+-----------+------------+-----------+-------------+--------------+---------------+
36 rows in set (0.00 sec)
```

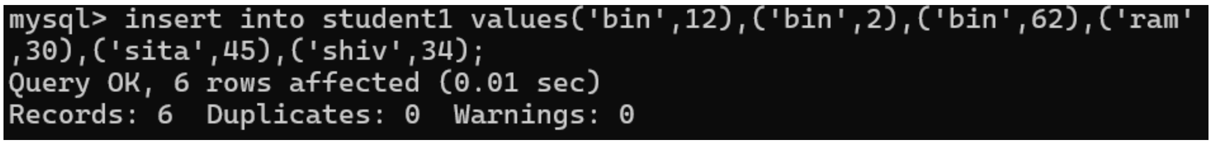
- Displays a list of all tables in your current database.
- Creates a new table called student1
- Inserts 6 rows into student1 table



```
mysql> show tables;
+-----------------+
| Tables_in_class |
+-----------------+
| student         |
| subject         |
+-----------------+
2 rows in set (0.01 sec)
```

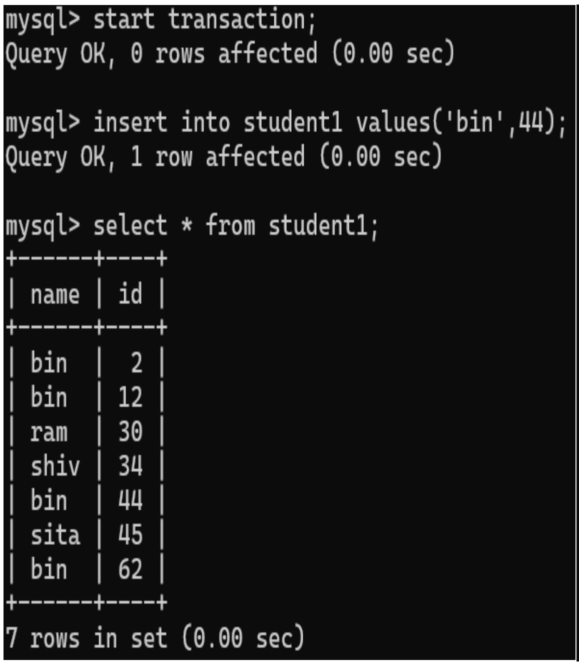


```
mysql> create table student1(name varchar(5),id int not null primary k
ey);
Query OK, 0 rows affected (0.05 sec)
```

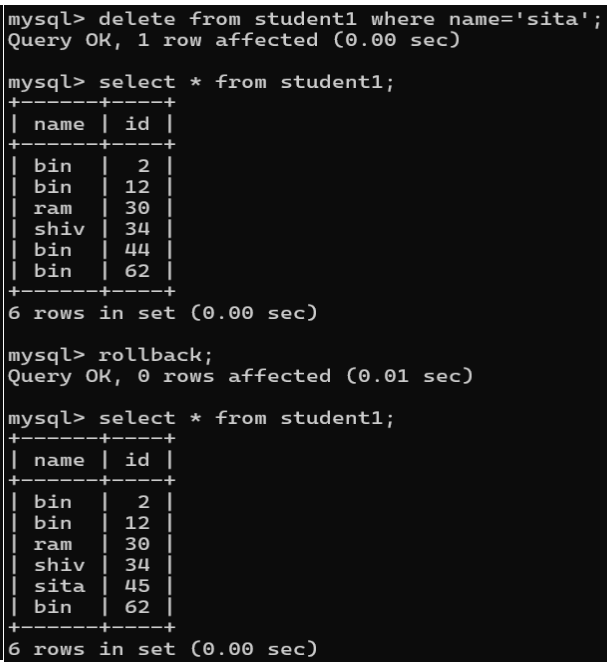


```
mysql> insert into student1 values('bin',12),('bin',2),('bin',62),('ram'
,30),('sita',45),('shiv',34);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

- Transaction started. Changes will not be permanent until commit.
- Added row (bin, 44).
- display all the records in your student table
- Deleted sita.
- All changes since start transaction; are undone
- Your table reverted to its state before starting the transaction



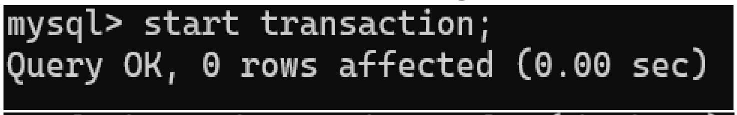
```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into student1 values('bin',44);
Query OK, 1 row affected (0.00 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| ram  | 30 |
| shiv | 34 |
| bin  | 44 |
| sita | 45 |
| bin  | 62 |
+------+----+
7 rows in set (0.00 sec)
```



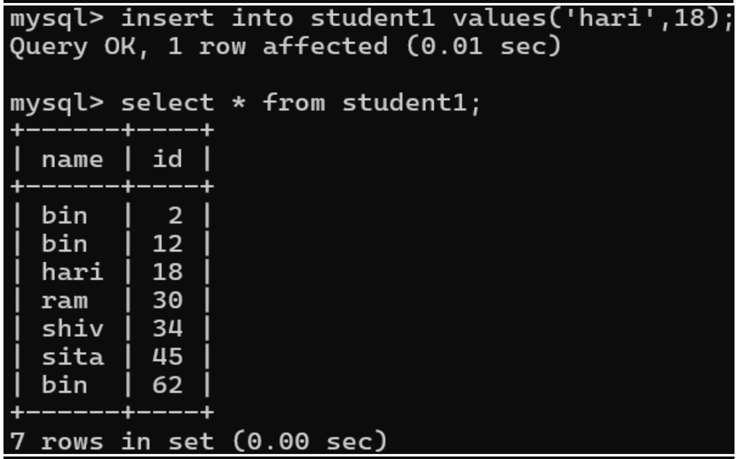
```
mysql> delete from student1 where name='sita';
Query OK, 1 row affected (0.00 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| ram  | 30 |
| shiv | 34 |
| bin  | 44 |
| bin  | 62 |
+------+----+
6 rows in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| ram  | 30 |
| shiv | 34 |
| sita | 45 |
| bin  | 62 |
+------+----+
6 rows in set (0.00 sec)
```

- Transaction started.
- Added row (hari, 18).
- Table now had 7 rows, including hari.
- Deleted the row (hari, 18). Table now back to 6 rows before committing
- This saved both actions (insert and delete) permanently:

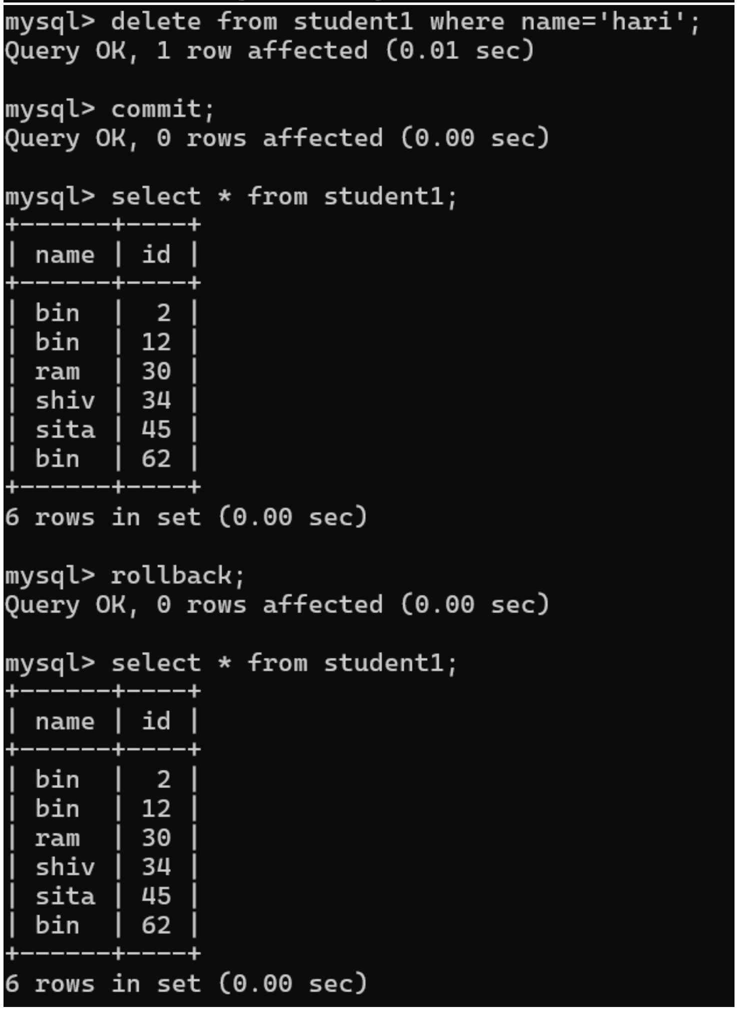
- hari was inserted
- Then hari was deleted
- So final table has no hari row.
- No effect because changes were already committed. Rollback only undoes uncommitted change.



```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)
```

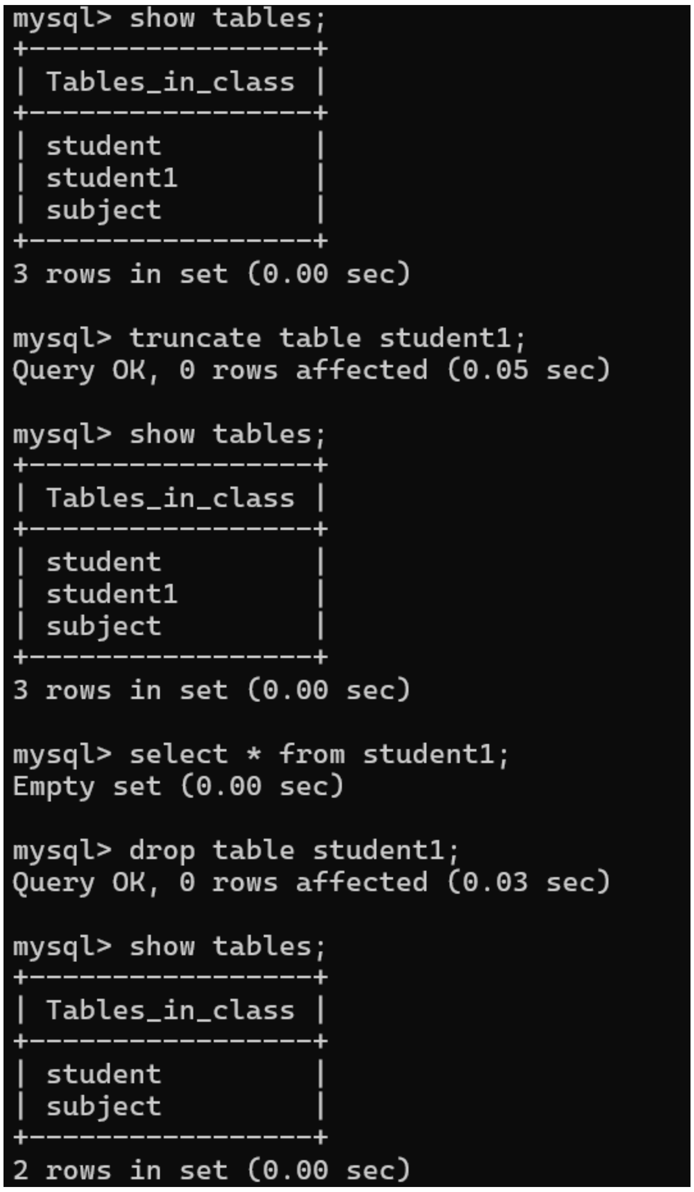


```
mysql> insert into student1 values('hari',18);
Query OK, 1 row affected (0.01 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| hari | 18 |
| ram  | 30 |
| shiv | 34 |
| sita | 45 |
| bin  | 62 |
+------+----+
7 rows in set (0.00 sec)
```



```
mysql> delete from student1 where name='hari';
Query OK, 1 row affected (0.01 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| ram  | 30 |
| shiv | 34 |
| sita | 45 |
| bin  | 62 |
+------+----+
6 rows in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from student1;
+------+----+
| name | id |
+------+----+
| bin  |  2 |
| bin  | 12 |
| ram  | 30 |
| shiv | 34 |
| sita | 45 |
| bin  | 62 |
+------+----+
6 rows in set (0.00 sec)
```

- Displayed all tables.
- **Deleted all rows** from student1.**Table structure remains** (empty table).
- student1 still exists (but now empty).
- Result: **Empty set** (no rows).
- Deleted **entire table structure and data** permanently from the database.
- Now only **student** and **subject** remain.



```
mysql> show tables;
+-----------------+
| Tables_in_class |
+-----------------+
| student         |
| student1        |
| subject         |
+-----------------+
3 rows in set (0.00 sec)

mysql> truncate table student1;
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+-----------------+
| Tables_in_class |
+-----------------+
| student         |
| student1        |
| subject         |
+-----------------+
3 rows in set (0.00 sec)

mysql> select * from student1;
Empty set (0.00 sec)

mysql> drop table student1;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----------------+
| Tables_in_class |
+-----------------+
| student         |
| subject         |
+-----------------+
2 rows in set (0.00 sec)
```