Data Manipulation Language:

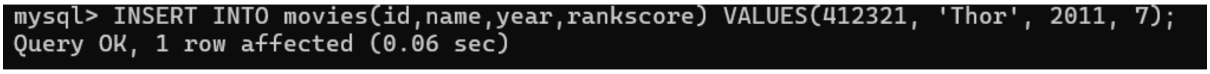
SELECT, INSERT, UPDATE, DELETE

Data Manipulation Language (DML) in SQL is a subset of commands that lets you interact with, retrieve, and modify the data stored within your database tables. In essence, DML focuses on querying and managing the data rather than defining the structure of the database (which is handled by Data Definition Language, or DDL).

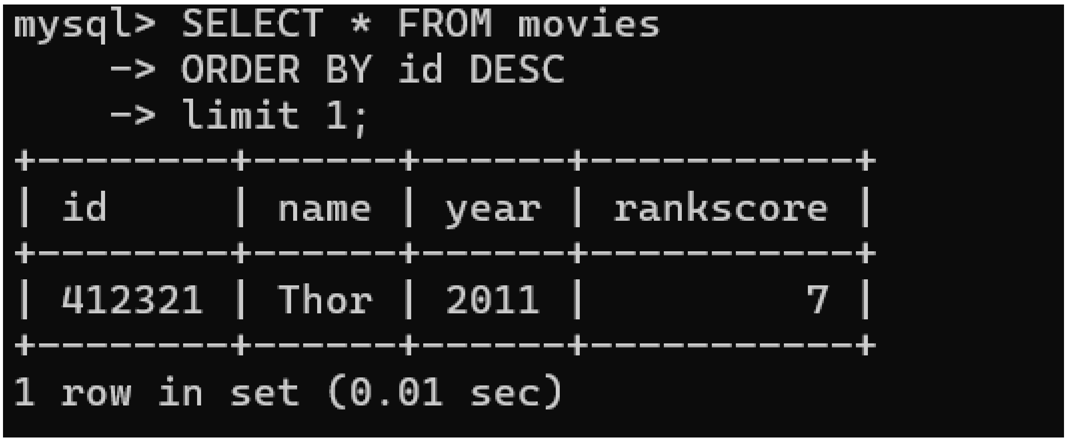
Here are the key aspects and commands of SQL DML:

INSERT

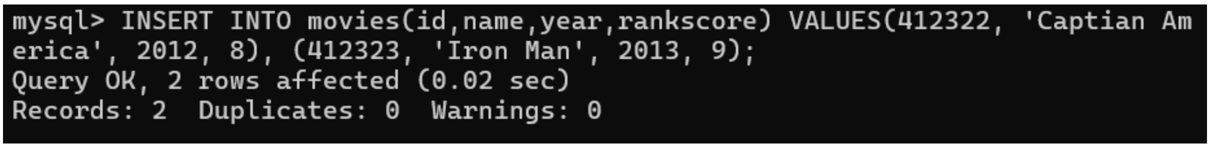
The INSERT command adds new rows of data into a table. You can insert data into all columns or only a subset, depending on how your table is defined. For example**:**



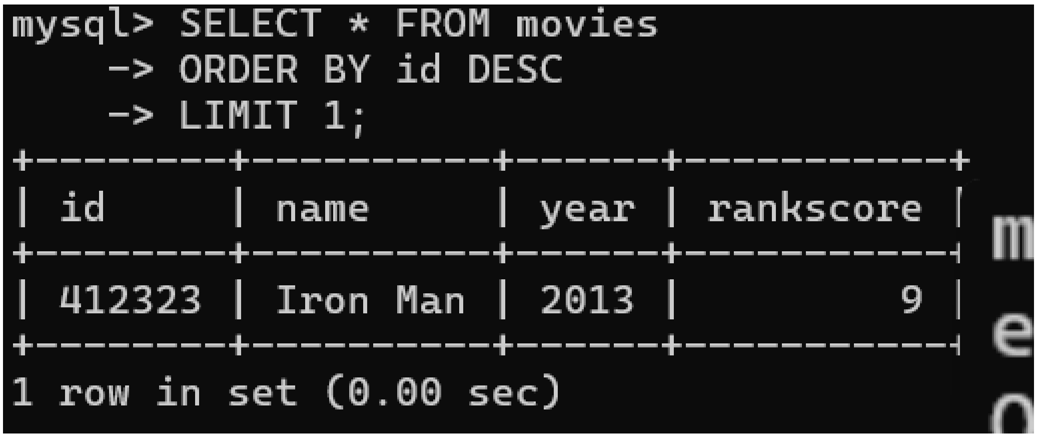
```
mysql> INSERT INTO movies(id,name,year,rankscore) VALUES(412321, 'Thor', 2011, 7);
Query OK, 1 row affected (0.06 sec)
```



```
mysql> SELECT * FROM movies
    -> ORDER BY id DESC
    -> limit 1;
+--------+------+------+-----------+
| id     | name | year | rankscore |
+--------+------+------+-----------+
| 412321 | Thor | 2011 |         7 |
+--------+------+------+-----------+
1 row in set (0.01 sec)
```



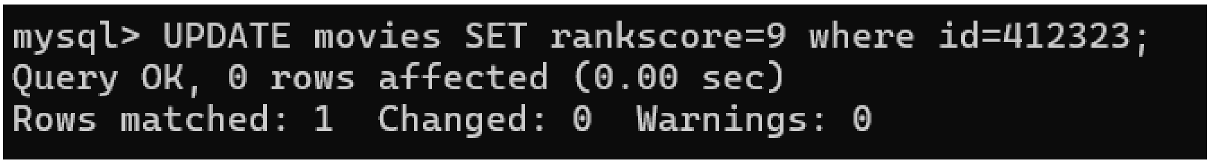
```
mysql> INSERT INTO movies(id,name,year,rankscore) VALUES(412322, 'Captian Am
erica', 2012, 8), (412323, 'Iron Man', 2013, 9);
Query OK, 2 rows affected (0.02 sec)
Records: 2  Duplicates: 0  Warnings: 0
```



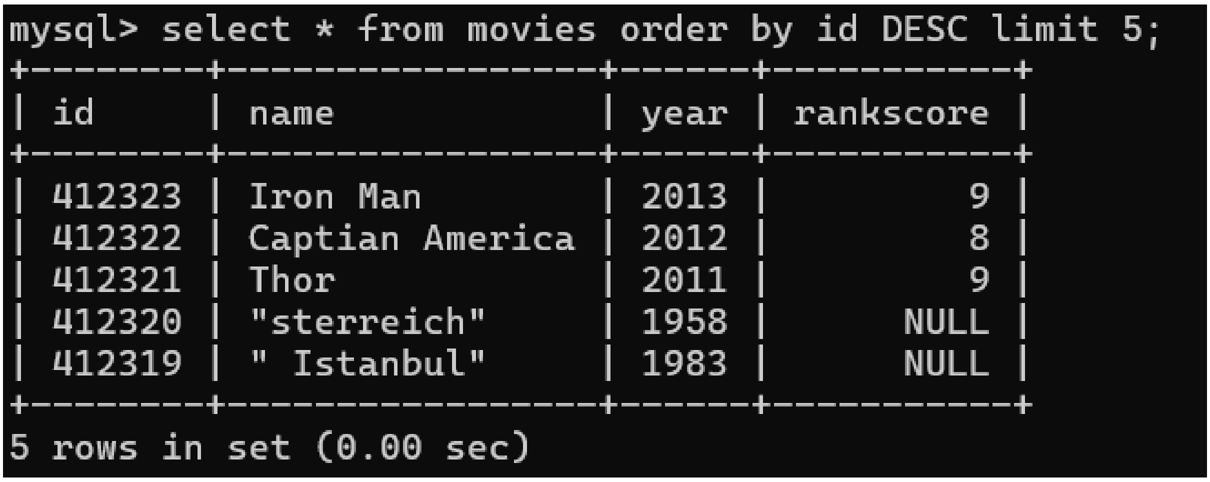
```
mysql> SELECT * FROM movies
    -> ORDER BY id DESC
    -> LIMIT 1;
+--------+----------+------+-----------+
| id     | name     | year | rankscore |
+--------+----------+------+-----------+
| 412323 | Iron Man | 2013 |         9 |
+--------+----------+------+-----------+
1 row in set (0.00 sec)
```

UPDATE:

With the UPDATE statement, you can modify existing data within a table. This command allows you to change values for one or more records that meet certain conditions. For example:



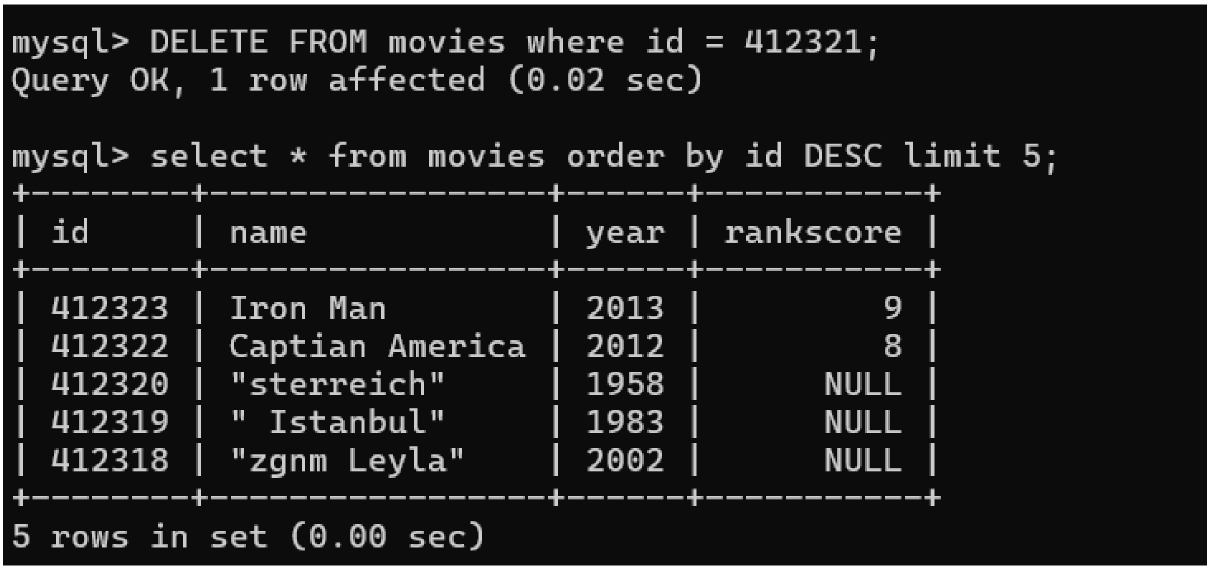
```
mysql> UPDATE movies SET rankscore=9 where id=412323;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```



```
mysql> select * from movies order by id DESC limit 5;
+--------+-----------------+------+-----------+
| id     | name            | year | rankscore |
+--------+-----------------+------+-----------+
| 412323 | Iron Man        | 2013 |         9 |
| 412322 | Captian America | 2012 |         8 |
| 412321 | Thor            | 2011 |         9 |
| 412320 | "sterreich"     | 1958 |      NULL |
| 412319 | " Istanbul"     | 1983 |      NULL |
+--------+-----------------+------+-----------+
5 rows in set (0.00 sec)
```

DELETE:

The DELETE statement removes one or more rows from a table based on specified criteria. For example:



```
mysql> DELETE FROM movies where id = 412321;
Query OK, 1 row affected (0.02 sec)

mysql> select * from movies order by id DESC limit 5;
+--------+-----------------+------+-----------+
| id     | name            | year | rankscore |
+--------+-----------------+------+-----------+
| 412323 | Iron Man        | 2013 |         9 |
| 412322 | Captian America | 2012 |         8 |
| 412320 | "sterreich"     | 1958 |      NULL |
| 412319 | " Istanbul"     | 1983 |      NULL |
| 412318 | "zgnm Leyla"    | 2002 |      NULL |
+--------+-----------------+------+-----------+
5 rows in set (0.00 sec)
```

Data Definition Language:

DDL, or Data Definition Language, in SQL is used to define and manage the structure of a database. It consists of commands that allow you to create, modify, and delete database objects like tables, indexes, and schemas.

Data Types:

SQL data types define the type of value that can be stored in a table column, For example, if we want a column to store only integer values. Then we can define it’s data type as int.

Constraints:

NOT NULL: Ensures that a column cannot have a NULL value

UNIQUE: Ensures that all values in a column are different

PRIMARY KEY: A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.

FOREIGN KEY: Uniquely identifies a row/record in another table.

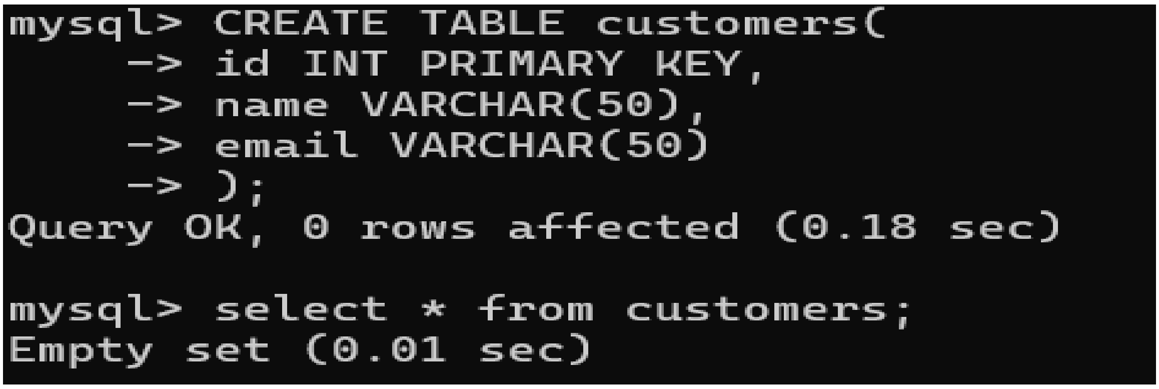
CHECK: Ensures that all values in a column satisfies a specifiec condition.

DEFAULT: Sets a default value in a column when no value is specified.

INDEX: Used to create and retrieve data from the database very quickly.

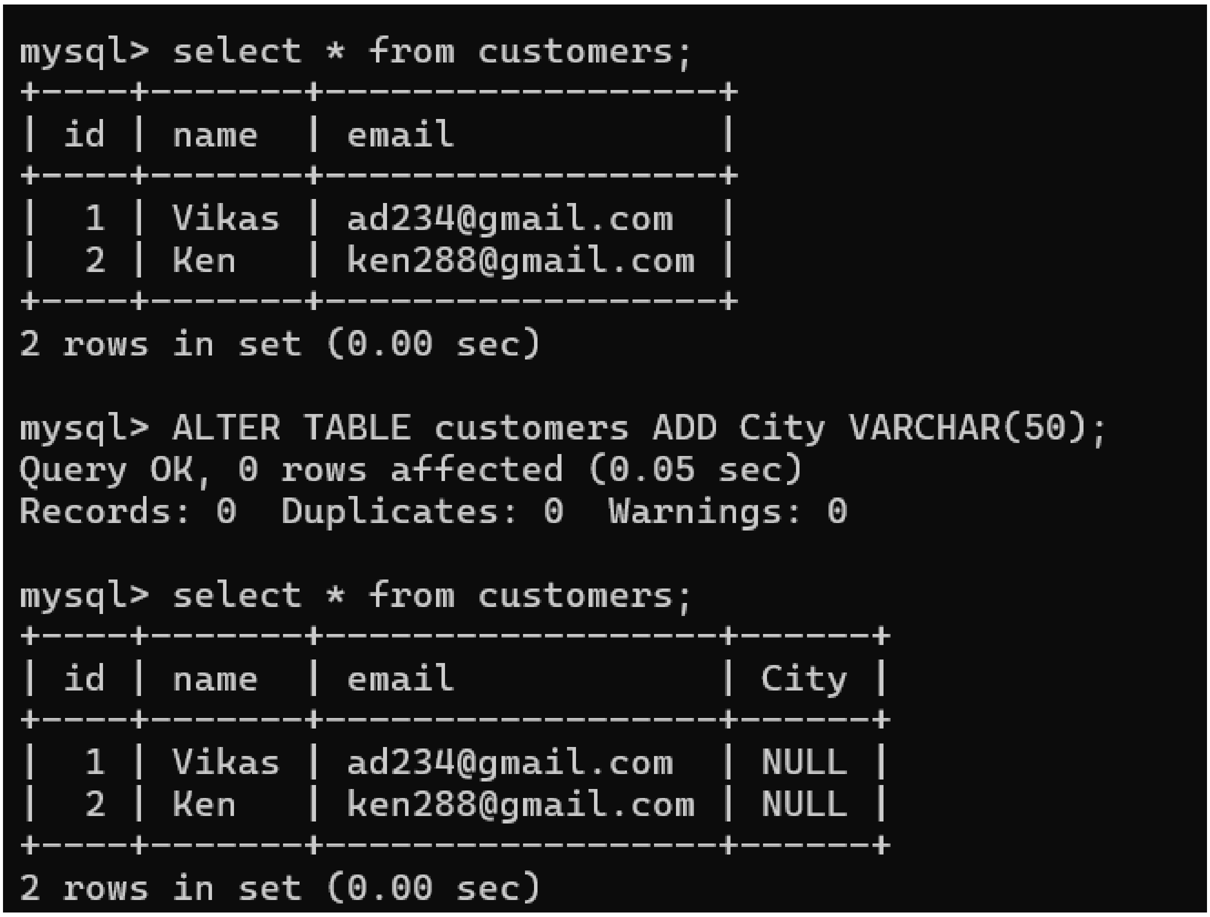
Here are the key DDL commands:

CREATE - Used to create database objects like tables, views, indexes, and schemas.



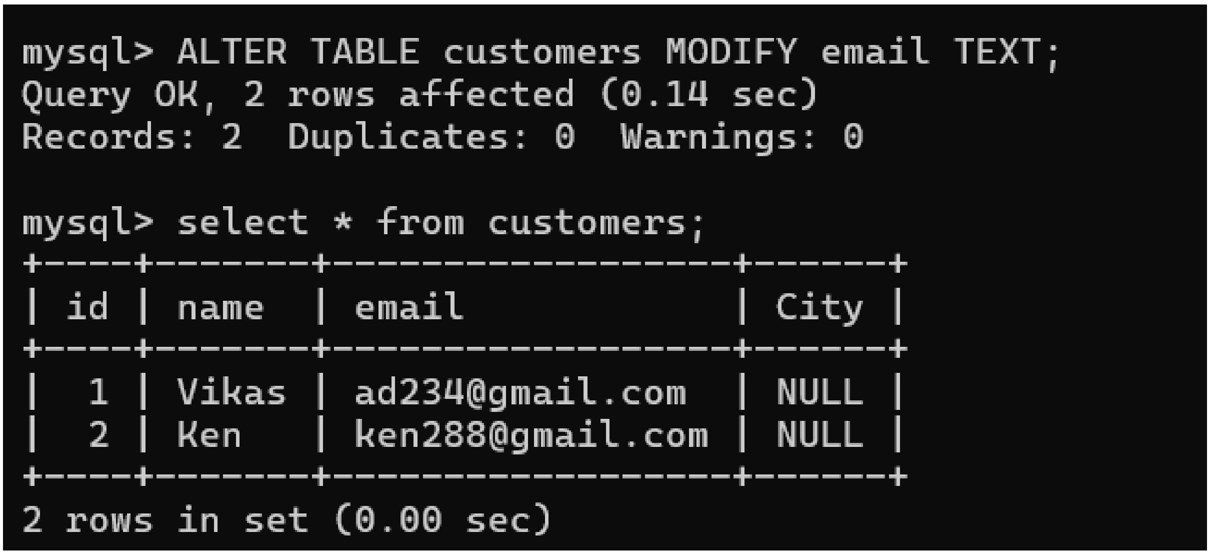
```
mysql> CREATE TABLE customers(
    -> id INT PRIMARY KEY,
    -> name VARCHAR(50),
    -> email VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.18 sec)

mysql> select * from customers;
Empty set (0.01 sec)
```

ALTER - Used to modify existing database structures, such as adding or removing columns.



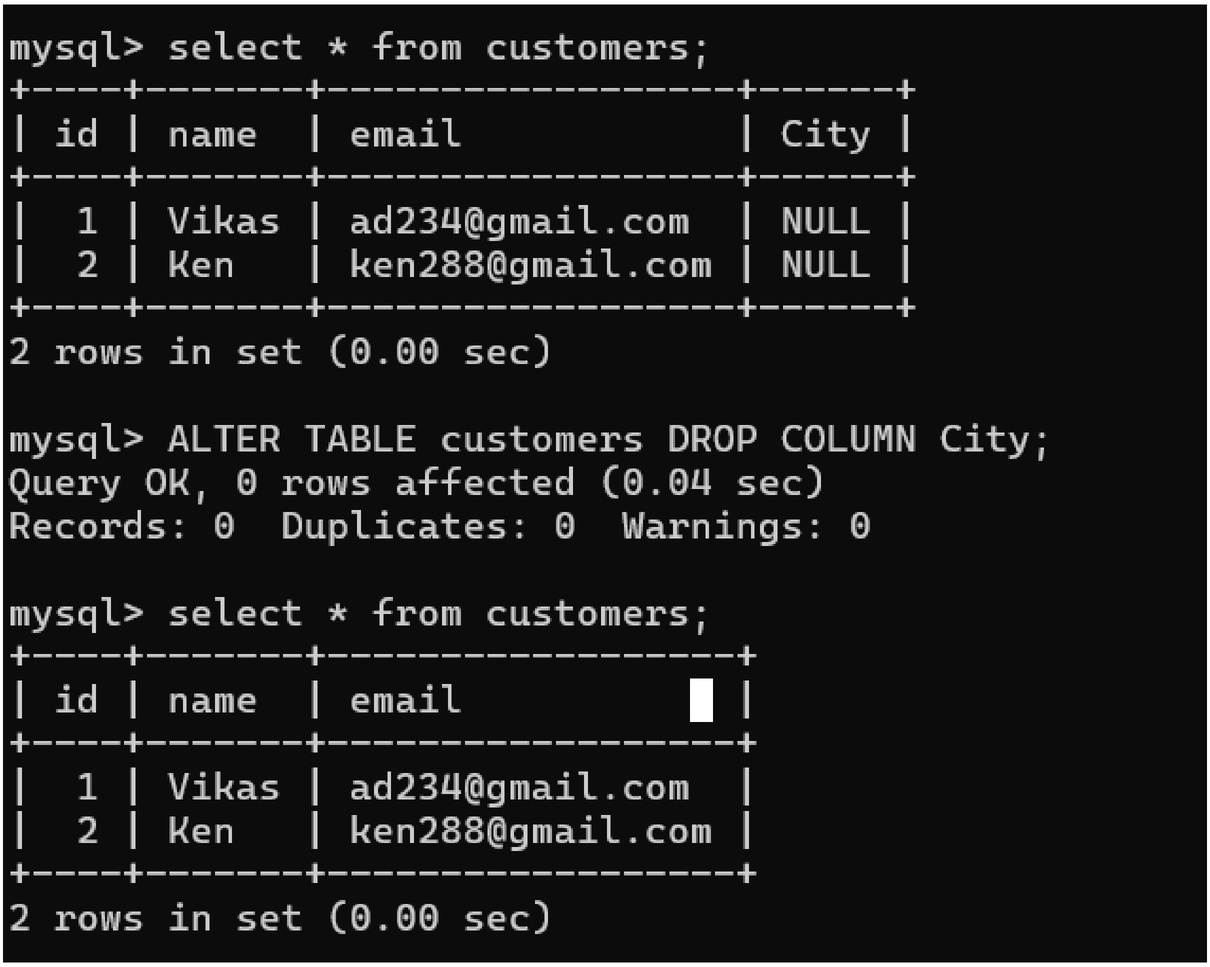
```
mysql> select * from customers;
+----+-------+-----------------+
| id | name  | email           |
+----+-------+-----------------+
|  1 | Vikas | ad234@gmail.com  |
|  2 | Ken   | ken288@gmail.com |
+----+-------+-----------------+
2 rows in set (0.00 sec)

mysql> ALTER TABLE customers ADD City VARCHAR(50);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from customers;
+----+-------+-----------------+------+
| id | name  | email           | City |
+----+-------+-----------------+------+
|  1 | Vikas | ad234@gmail.com  | NULL |
|  2 | Ken   | ken288@gmail.com | NULL |
+----+-------+-----------------+------+
2 rows in set (0.00 sec)
```

MODIFY: Used to change the data type or properties of an existing column.



```
mysql> ALTER TABLE customers MODIFY email TEXT;
Query OK, 2 rows affected (0.14 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from customers;
+----+-------+-----------------+------+
| id | name  | email           | City |
+----+-------+-----------------+------+
|  1 | Vikas | ad234@gmail.com  | NULL |
|  2 | Ken   | ken288@gmail.com | NULL |
+----+-------+-----------------+------+
2 rows in set (0.00 sec)
```

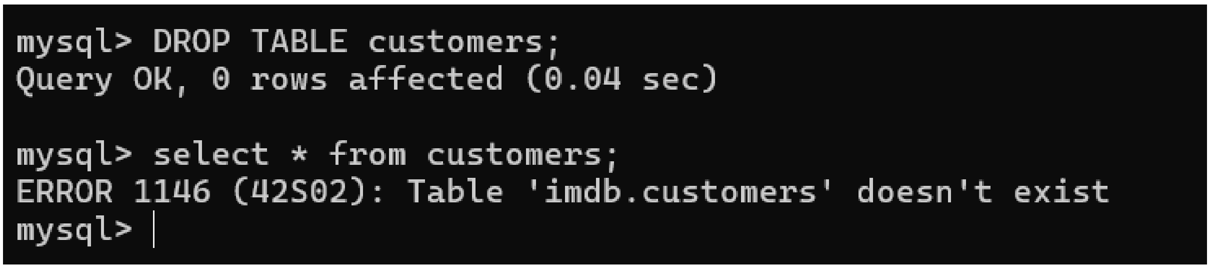
DROP - Used to delete entire database objects, like tables or schemas.



```
mysql> select * from customers;
+----+-------+------------------+------+
| id | name  | email            | City |
+----+-------+------------------+------+
|  1 | Vikas | ad234@gmail.com  | NULL |
|  2 | Ken   | ken288@gmail.com | NULL |
+----+-------+------------------+------+
2 rows in set (0.00 sec)

mysql> ALTER TABLE customers DROP COLUMN City;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from customers;
+----+-------+------------------+
| id | name  | email            |
+----+-------+------------------+
|  1 | Vikas | ad234@gmail.com  |
|  2 | Ken   | ken288@gmail.com |
+----+-------+------------------+
2 rows in set (0.00 sec)
```

Drop Table:

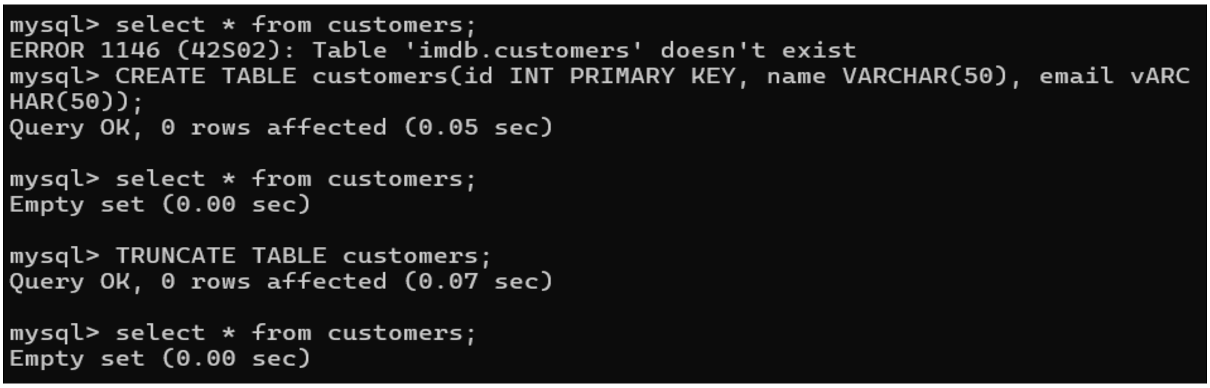
- Removes the entire table along with its structure.
- Once dropped, the table cannot be recovered unless recreated.
- Example:



```
mysql> DROP TABLE customers;
Query OK, 0 rows affected (0.04 sec)

mysql> select * from customers;
ERROR 1146 (42S02): Table 'imdb.customers' doesn't exist
mysql> |
```

TRUNCATE:

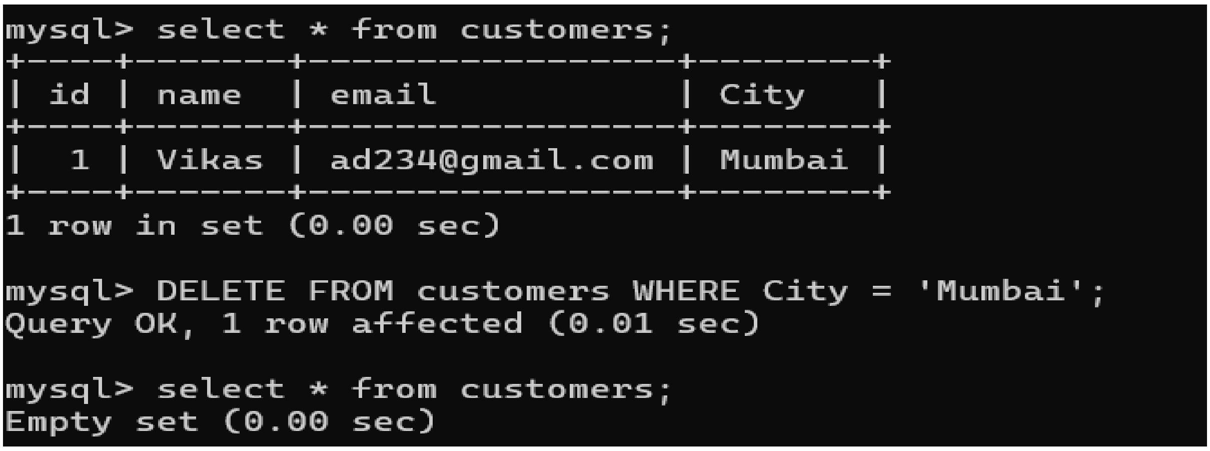
- Deletes all rows in a table but keeps its structure intact.
- It's faster than DELETE since it doesn't log individual row deletions.
- Example:



```
mysql> select * from customers;
ERROR 1146 (42S02): Table 'imdb.customers' doesn't exist
mysql> CREATE TABLE customers(id INT PRIMARY KEY, name VARCHAR(50), email vARC
HAR(50));
Query OK, 0 rows affected (0.05 sec)

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

mysql> TRUNCATE TABLE customers;
Query OK, 0 rows affected (0.07 sec)

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

DELETE:

- Removes specific rows based on a condition.
- Allows recovery with transactions ROLLBACK.
- Example(deleting records of customers from Mumbai).



```
mysql> select * from customers;
+----+-------+-----------------+--------+
| id | name  | email           | City   |
+----+-------+-----------------+--------+
|  1 | Vikas | ad234@gmail.com | Mumbai |
+----+-------+-----------------+--------+
1 row in set (0.00 sec)

mysql> DELETE FROM customers WHERE City = 'Mumbai';
Query OK, 1 row affected (0.01 sec)

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

Data Control Language

(DCL) is used to manage **user permissions** and control access to database objects. The two main commands are **GRANT** and **REVOKE**:

1. **GRANT(Giving Permissions)**

- Used to assign specific privileges to users or roles.
- Examples: Giving a user permission to SELECT and INSERT on a table.

GRANT SELECT, INSERT ON customers TO 'user1'@'localhost';

- User1 can now read and insert data into the customers table.

2. **REVOKE(Removing Permissions)**

- Used to take away previously granted privileges.
- Example: Revoking INSERT permission:

REVOKE INSERT ON customers FROM 'user1'@'localhost';

- User1 can still read SELECT data but can no longer insert records.