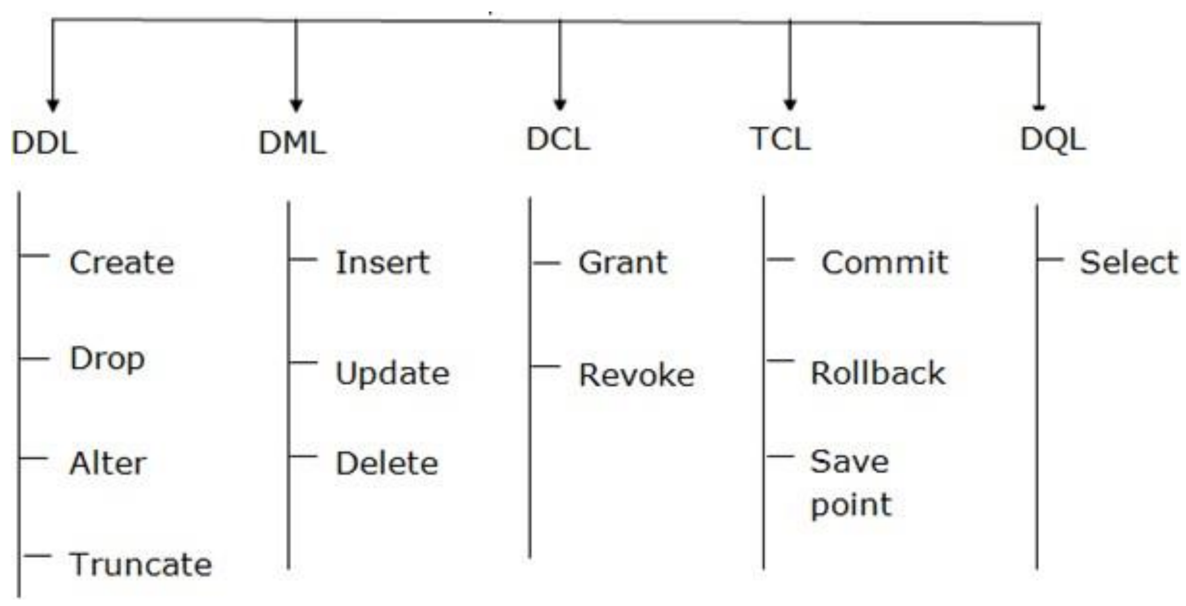
SQL Commands

SQL stands for structured Query Language. SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data. SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

- **Types of commands**
  There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.

SQL COMMANDS



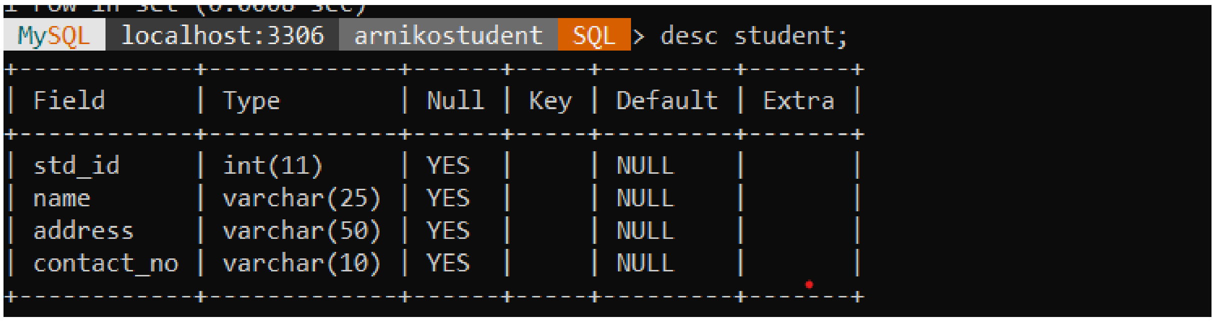
CREATE Command
It is used to create a new table in the database.
Syntax:

```
CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[, ...................]);
```

Example 1:

```
Create table student(std_id integer,name varchar(25),address
varchar(50),contact_no varchar(10));
```

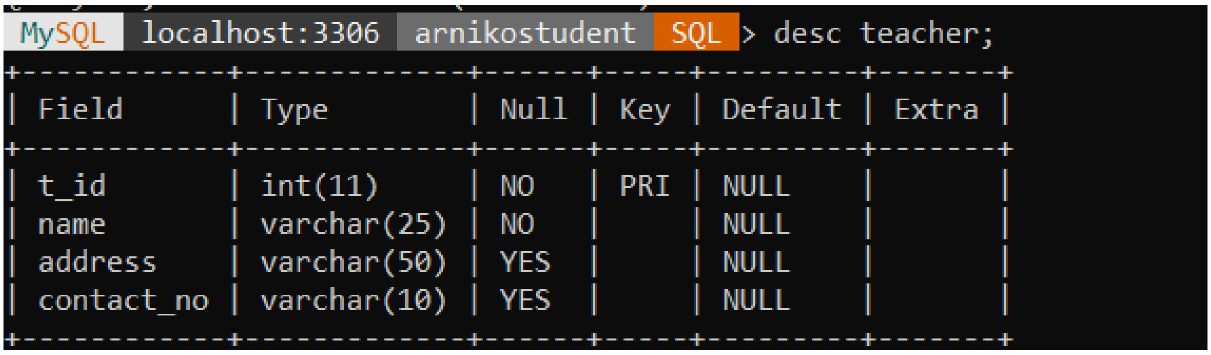
Output 1:



```
MySQL  localhost:3306  arnikostudent  SQL > desc student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| std_id     | int(11)     | YES  |     | NULL    |       |
| name       | varchar(25) | YES  |     | NULL    |       |
| address    | varchar(50) | YES  |     | NULL    |       |
| contact_no | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Example 2:
Create table student(t_id integer,name varchar(25),address varchar(50),contact_no varchar(10), primary key(t_id));

Output 2:



```
MySQL  localhost:3306  arnikostudent  SQL > desc teacher;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| t_id       | int(11)     | NO   | PRI | NULL    |       |
| name       | varchar(25) | NO   |     | NULL    |       |
| address    | varchar(50) | YES  |     | NULL    |       |
| contact_no | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Insert Command

The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

INSERT INTO TABLE_NAME(col1, col2, col3,.... col N) VALUES

(value1,value2,value 3,.....valueN); **OR**

INSERT INTO TABLE_NAME VALUES (value1,value2,value 3, ...valueN);

Example:

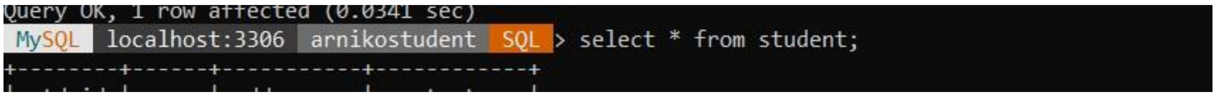
Insert into student values (1,”Junu”,”Kathmandu”,”9812343567”);

To see the output of the insert, we have to use the select command.

Select Command

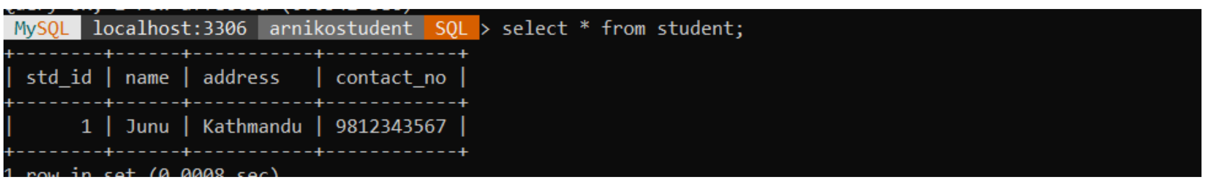
This is the same as the projection operation of relational algebra.

Example:



```
Query OK, 1 row affected (0.0341 sec)
MySQL  localhost:3306  arnikostudent  SQL > select * from student;
+--------+------+----------+------------+
```

Output:



```
MySQL  localhost:3306  arnikostudent  SQL > select * from student;
+--------+------+----------+------------+
| std_id | name | address  | contact_no |
+--------+------+----------+------------+
|      1 | Junu | Kathmandu | 9812343567 |
+--------+------+----------+------------+
1 row in set (0.0008 sec)
```

Alter Command

It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax:

ALTER TABLE table_name MODIFY(column_definitions ....);

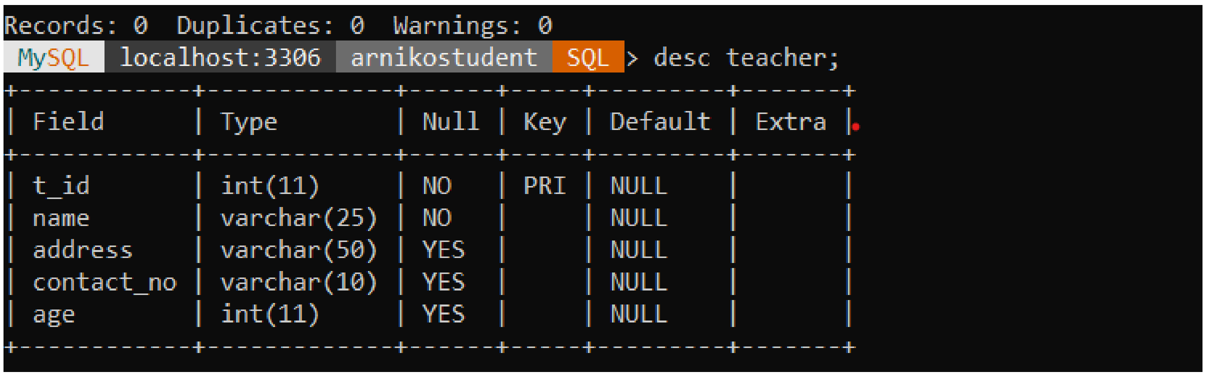
OR

ALTER TABLE table_name ADD(column_definitions ....);

Example:

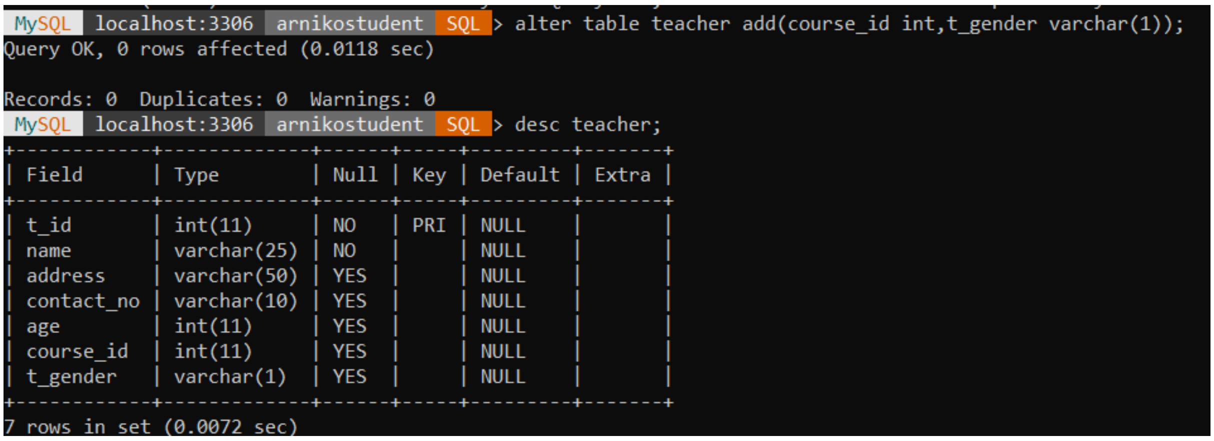
Alter table teacher add(age int);

Output:



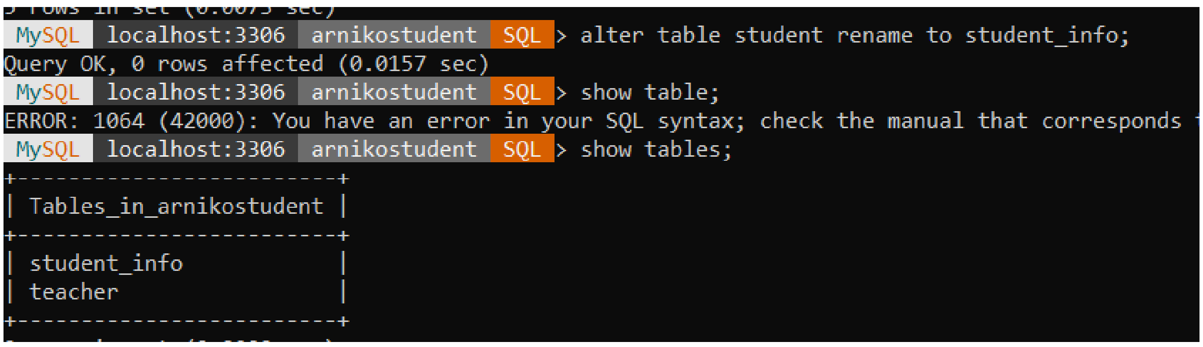
```
Records: 0  Duplicates: 0  Warnings: 0
MySQL  localhost:3306  arnikostudent  SQL > desc teacher;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| t_id       | int(11)     | NO   | PRI | NULL    |       |
| name       | varchar(25) | NO   |     | NULL    |       |
| address    | varchar(50) | YES  |     | NULL    |       |
| contact_no | varchar(10) | YES  |     | NULL    |       |
| age        | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Example & Output: 2



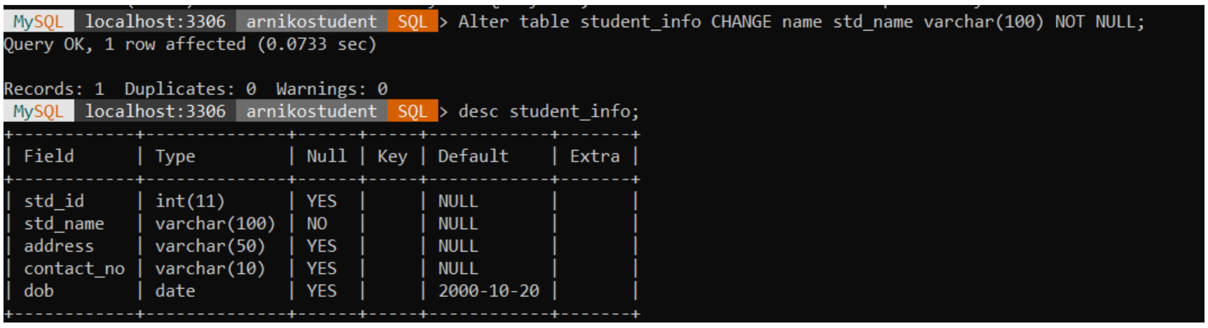
```
 MySQL  localhost:3306  arnikostudent  SQL > alter table teacher add(course_id int,t_gender varchar(1));
Query OK, 0 rows affected (0.0118 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  localhost:3306  arnikostudent  SQL > desc teacher;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| t_id       | int(11)     | NO   | PRI | NULL    |       |
| name       | varchar(25) | NO   |     | NULL    |       |
| address    | varchar(50) | YES  |     | NULL    |       |
| contact_no | varchar(10) | YES  |     | NULL    |       |
| age        | int(11)     | YES  |     | NULL    |       |
| course_id  | int(11)     | YES  |     | NULL    |       |
| t_gender   | varchar(1)  | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.0072 sec)
```

Example & Output: 3



```
5 rows in set (0.0073 sec)
 MySQL  localhost:3306  arnikostudent  SQL > alter table student rename to student_info;
Query OK, 0 rows affected (0.0157 sec)
 MySQL  localhost:3306  arnikostudent  SQL > show table;
ERROR: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
 MySQL  localhost:3306  arnikostudent  SQL > show tables;
+-------------------------+
| Tables_in_arnikostudent |
+-------------------------+
| student_info            |
| teacher                 |
+-------------------------+
```

Example & Output: 4



```
 MySQL  localhost:3306  arnikostudent  SQL > Alter table student_info CHANGE name std_name varchar(100) NOT NULL;
Query OK, 1 row affected (0.0733 sec)

Records: 1  Duplicates: 0  Warnings: 0
 MySQL  localhost:3306  arnikostudent  SQL > desc student_info;
+------------+--------------+------+-----+------------+-------+
| Field      | Type         | Null | Key | Default    | Extra |
+------------+--------------+------+-----+------------+-------+
| std_id     | int(11)      | YES  |     | NULL       |       |
| std_name   | varchar(100) | NO   |     | NULL       |       |
| address    | varchar(50)  | YES  |     | NULL       |       |
| contact_no | varchar(10)  | YES  |     | NULL       |       |
| dob        | date         | YES  |     | 2000-10-20 |       |
+------------+--------------+------+-----+------------+-------+
```

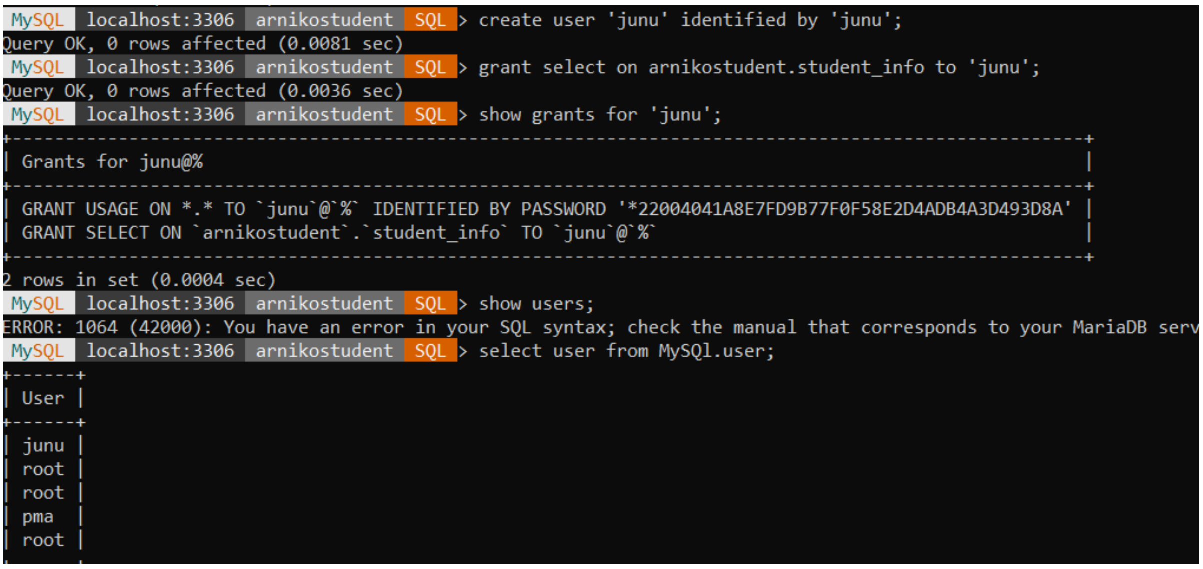
Grant Command

It is used to give user access privileges to a database.

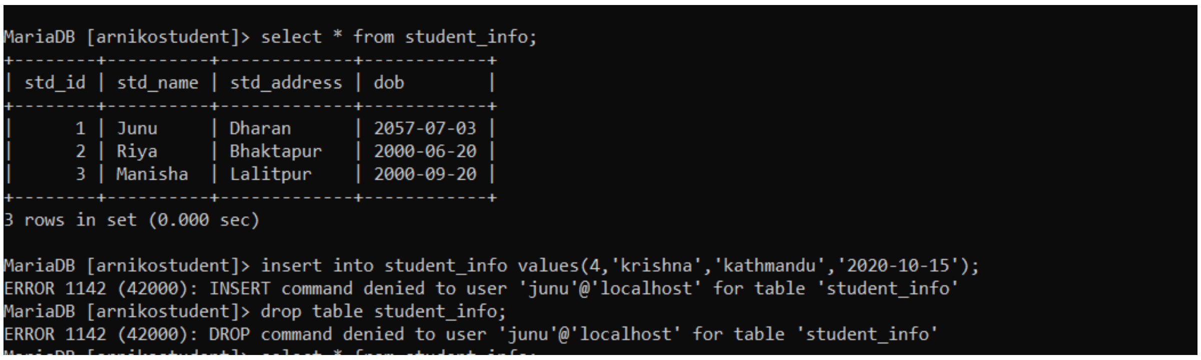
Syntax:

GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

Example with output:



```
MySQL  localhost:3306  arnikostudent  SQL > create user 'junu' identified by 'junu';
Query OK, 0 rows affected (0.0081 sec)
MySQL  localhost:3306  arnikostudent  SQL > grant select on arnikostudent.student_info to 'junu';
Query OK, 0 rows affected (0.0036 sec)
MySQL  localhost:3306  arnikostudent  SQL > show grants for 'junu';
+----------------------------------------------------------------------------------------------------+
| Grants for junu@%                                                                                  |
+----------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `junu`@`%` IDENTIFIED BY PASSWORD '*22004041A8E7FD9B77F0F58E2D4ADB4A3D493D8A' |
| GRANT SELECT ON `arnikostudent`.`student_info` TO `junu`@`%`                                       |
+----------------------------------------------------------------------------------------------------+
2 rows in set (0.0004 sec)
MySQL  localhost:3306  arnikostudent  SQL > show users;
ERROR: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB serv
MySQL  localhost:3306  arnikostudent  SQL > select user from MySQl.user;
+------+
| User |
+------+
| junu |
| root |
| root |
| pma  |
| root |
```



```
MariaDB [arnikostudent]> select * from student_info;
+--------+----------+-------------+------------+
| std_id | std_name | std_address | dob        |
+--------+----------+-------------+------------+
|      1 | Junu     | Dharan      | 2057-07-03 |
|      2 | Riya     | Bhaktapur   | 2000-06-20 |
|      3 | Manisha  | Lalitpur    | 2000-09-20 |
+--------+----------+-------------+------------+
3 rows in set (0.000 sec)

MariaDB [arnikostudent]> insert into student_info values(4,'krishna','kathmandu','2020-10-15');
ERROR 1142 (42000): INSERT command denied to user 'junu'@'localhost' for table 'student_info'
MariaDB [arnikostudent]> drop table student_info;
ERROR 1142 (42000): DROP command denied to user 'junu'@'localhost' for table 'student_info'
```

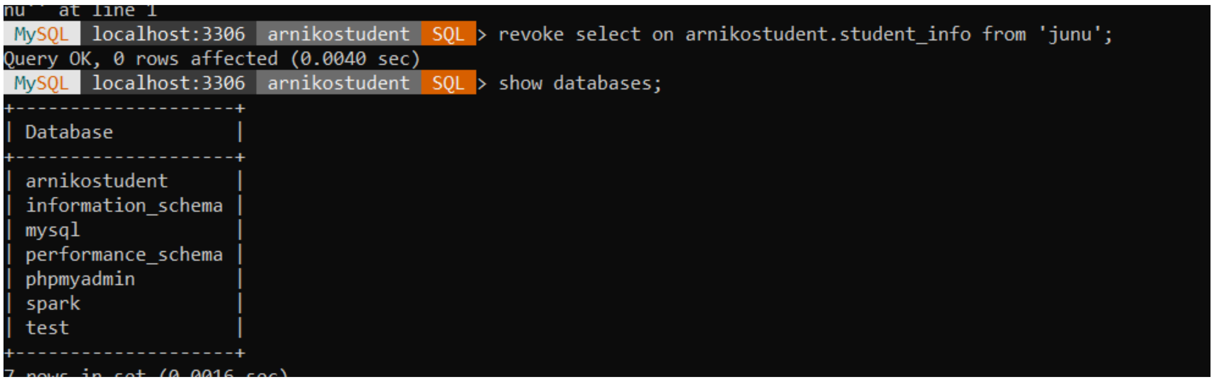
Revoke Command

It is used to take back permissions from the user.

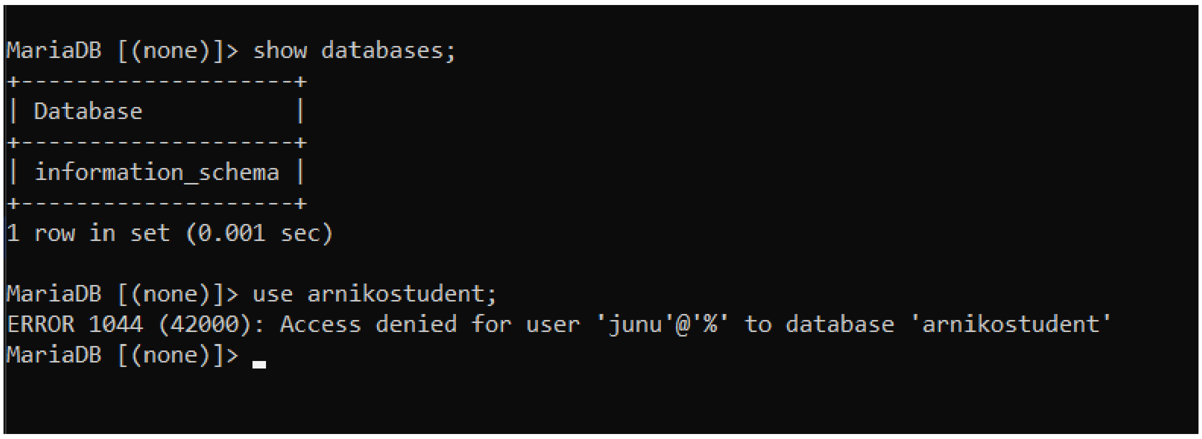
Syntax:

REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

Example with Output:



```
nu'` at line 1
 MySQL  localhost:3306  arnikostudent  SQL > revoke select on arnikostudent.student_info from 'junu';
Query OK, 0 rows affected (0.0040 sec)
 MySQL  localhost:3306  arnikostudent  SQL > show databases;
+--------------------+
| Database           |
+--------------------+
| arnikostudent      |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| spark              |
| test               |
+--------------------+
7 rows in set (0.0016 sec)
```



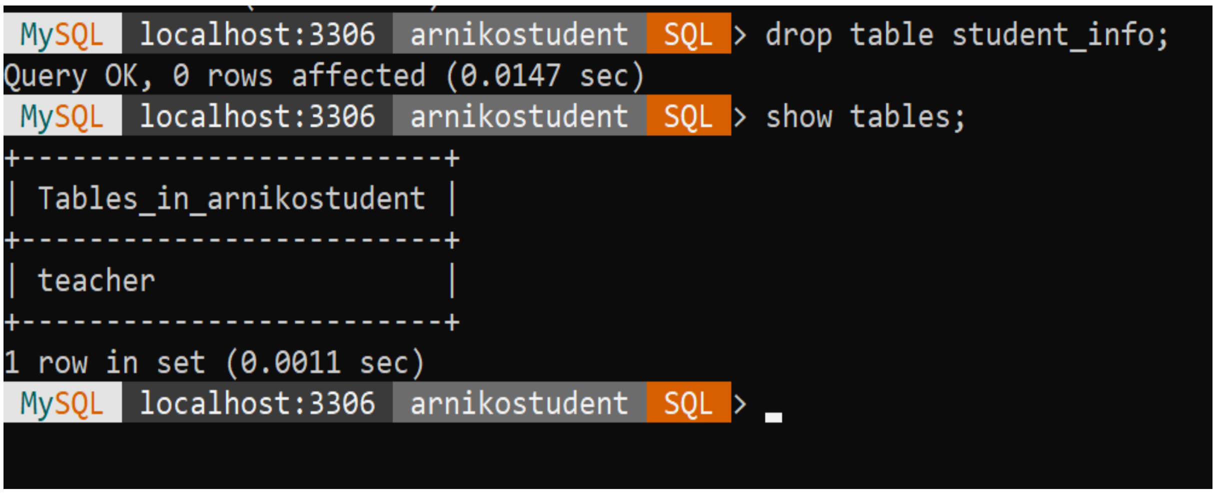
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.001 sec)

MariaDB [(none)]> use arnikostudent;
ERROR 1044 (42000): Access denied for user 'junu'@'%' to database 'arnikostudent'
MariaDB [(none)]> _
```

Drop Command

DROP is used **to delete a whole database or just a table**. The DROP statement destroys objects like an existing database, table, index, or view.

Syntax: DROP TABLE table_name;

Example with output:

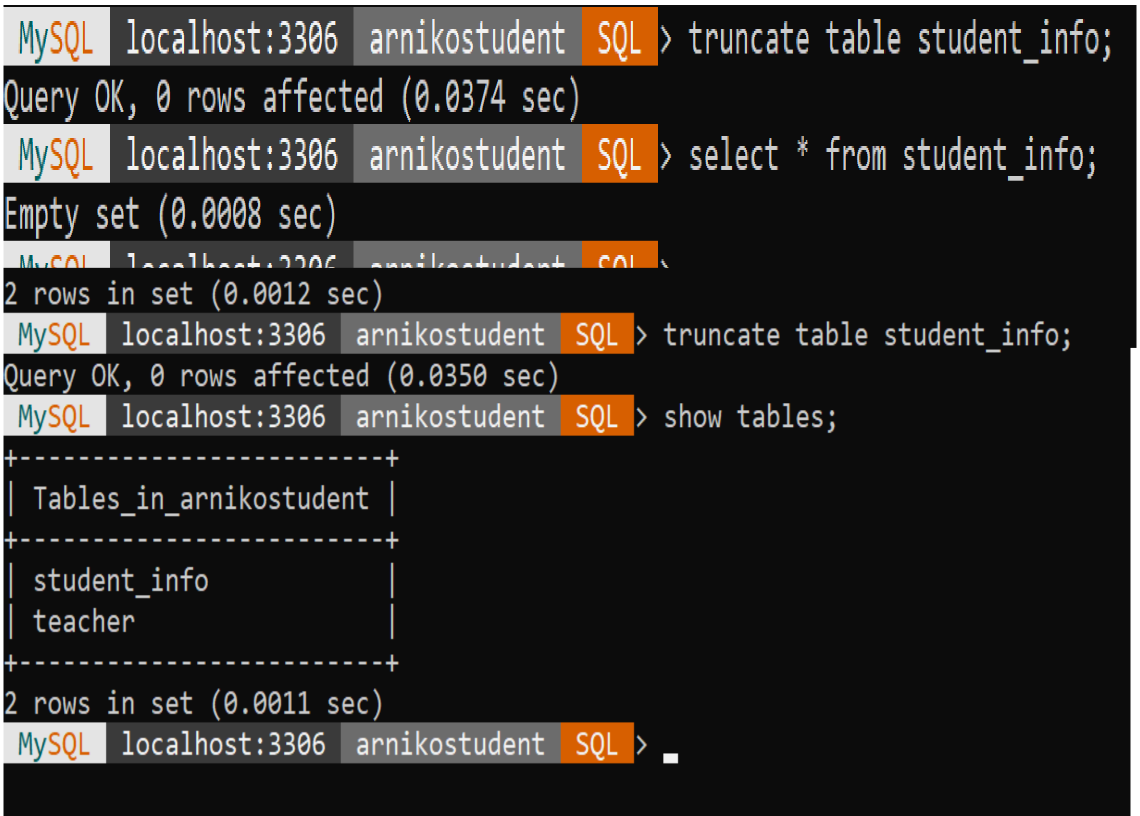


```
MySQL  localhost:3306  arnikostudent  SQL > drop table student_info;
Query OK, 0 rows affected (0.0147 sec)
MySQL  localhost:3306  arnikostudent  SQL > show tables;
+-------------------------+
| Tables_in_arnikostudent |
+-------------------------+
| teacher                 |
+-------------------------+
1 row in set (0.0011 sec)
MySQL  localhost:3306  arnikostudent  SQL > _
```

Truncate Command

The **TRUNCATE** command deletes the data inside a table, but not the table itself.

Syntax:
TRUNCATE TABLE table_name;



```
MySQL  localhost:3306  arnikostudent  SQL > truncate table student_info;
Query OK, 0 rows affected (0.0374 sec)
MySQL  localhost:3306  arnikostudent  SQL > select * from student_info;
Empty set (0.0008 sec)
```

```
2 rows in set (0.0012 sec)
MySQL  localhost:3306  arnikostudent  SQL > truncate table student_info;
Query OK, 0 rows affected (0.0350 sec)
MySQL  localhost:3306  arnikostudent  SQL > show tables;
+-------------------------+
| Tables_in_arnikostudent |
+-------------------------+
| student_info            |
| teacher                 |
+-------------------------+
2 rows in set (0.0011 sec)
MySQL  localhost:3306  arnikostudent  SQL > _
```

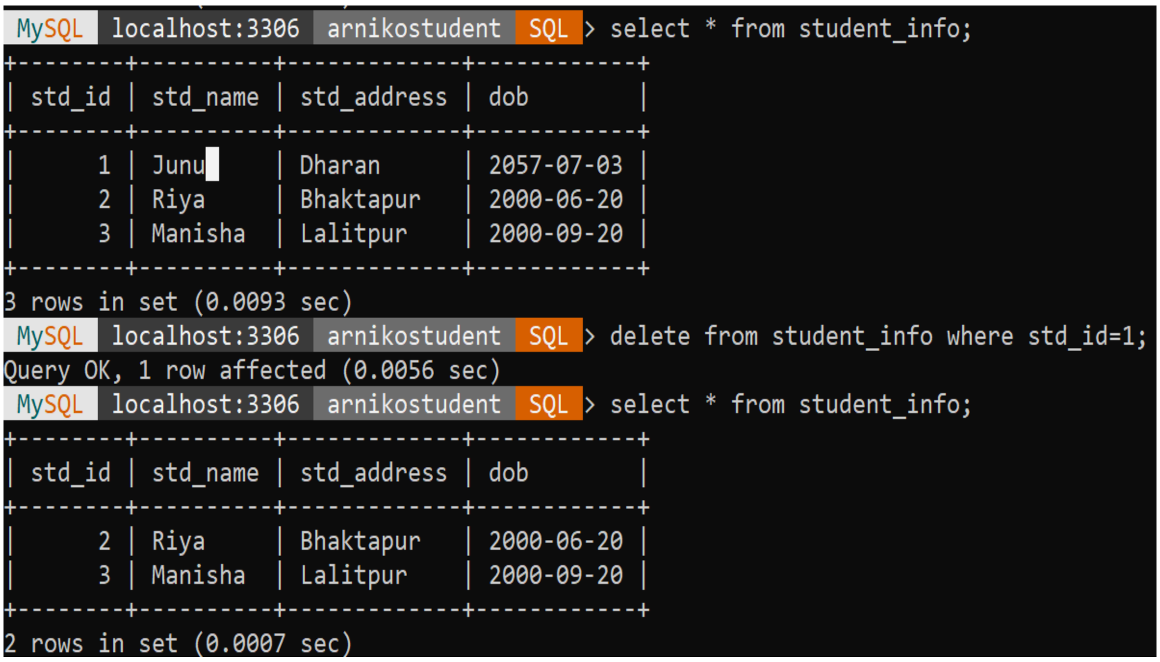
Delete Command

The `DELETE` command is used to delete existing records in a table.

Syntax:

`DELETE FROM` *`table_name`* `WHERE` *`condition`*`;`

Example with output:



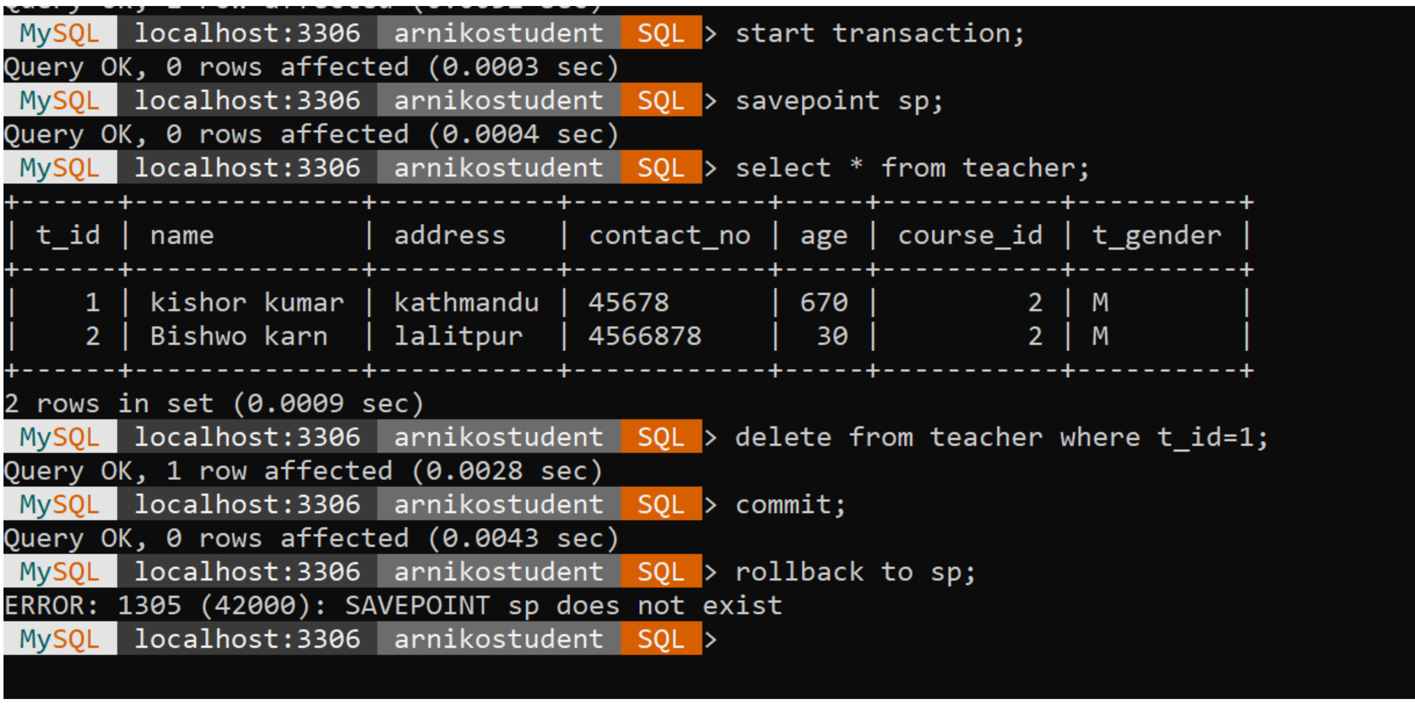
```
MySQL  localhost:3306  arnikostudent  SQL > select * from student_info;
+--------+----------+-------------+------------+
| std_id | std_name | std_address | dob        |
+--------+----------+-------------+------------+
|      1 | Junu     | Dharan      | 2057-07-03 |
|      2 | Riya     | Bhaktapur   | 2000-06-20 |
|      3 | Manisha  | Lalitpur    | 2000-09-20 |
+--------+----------+-------------+------------+
3 rows in set (0.0093 sec)
MySQL  localhost:3306  arnikostudent  SQL > delete from student_info where std_id=1;
Query OK, 1 row affected (0.0056 sec)
MySQL  localhost:3306  arnikostudent  SQL > select * from student_info;
+--------+----------+-------------+------------+
| std_id | std_name | std_address | dob        |
+--------+----------+-------------+------------+
|      2 | Riya     | Bhaktapur   | 2000-06-20 |
|      3 | Manisha  | Lalitpur    | 2000-09-20 |
+--------+----------+-------------+------------+
2 rows in set (0.0007 sec)
```

10) Commit Command

It is used to end your current transaction and make permanent all changes performed in the transaction.

Syntax: COMMIT;

Example with output:



```
 MySQL  localhost:3306  arnikostudent  SQL > start transaction;
Query OK, 0 rows affected (0.0003 sec)
 MySQL  localhost:3306  arnikostudent  SQL > savepoint sp;
Query OK, 0 rows affected (0.0004 sec)
 MySQL  localhost:3306  arnikostudent  SQL > select * from teacher;
+------+--------------+-----------+------------+-----+-----------+----------+
| t_id | name         | address   | contact_no | age | course_id | t_gender |
+------+--------------+-----------+------------+-----+-----------+----------+
|    1 | kishor kumar | kathmandu | 45678      | 670 |         2 | M        |
|    2 | Bishwo karn  | lalitpur  | 4566878    |  30 |         2 | M        |
+------+--------------+-----------+------------+-----+-----------+----------+
2 rows in set (0.0009 sec)
 MySQL  localhost:3306  arnikostudent  SQL > delete from teacher where t_id=1;
Query OK, 1 row affected (0.0028 sec)
 MySQL  localhost:3306  arnikostudent  SQL > commit;
Query OK, 0 rows affected (0.0043 sec)
 MySQL  localhost:3306  arnikostudent  SQL > rollback to sp;
ERROR: 1305 (42000): SAVEPOINT sp does not exist
 MySQL  localhost:3306  arnikostudent  SQL >
```

11) Roll Back Command

ROLLBACK in SQL is a transactional control language that is used to undo the transactions that have not been saved in the database. The command is only been used to undo changes since the last COMMIT.

Syntax:

ROLLBACK;

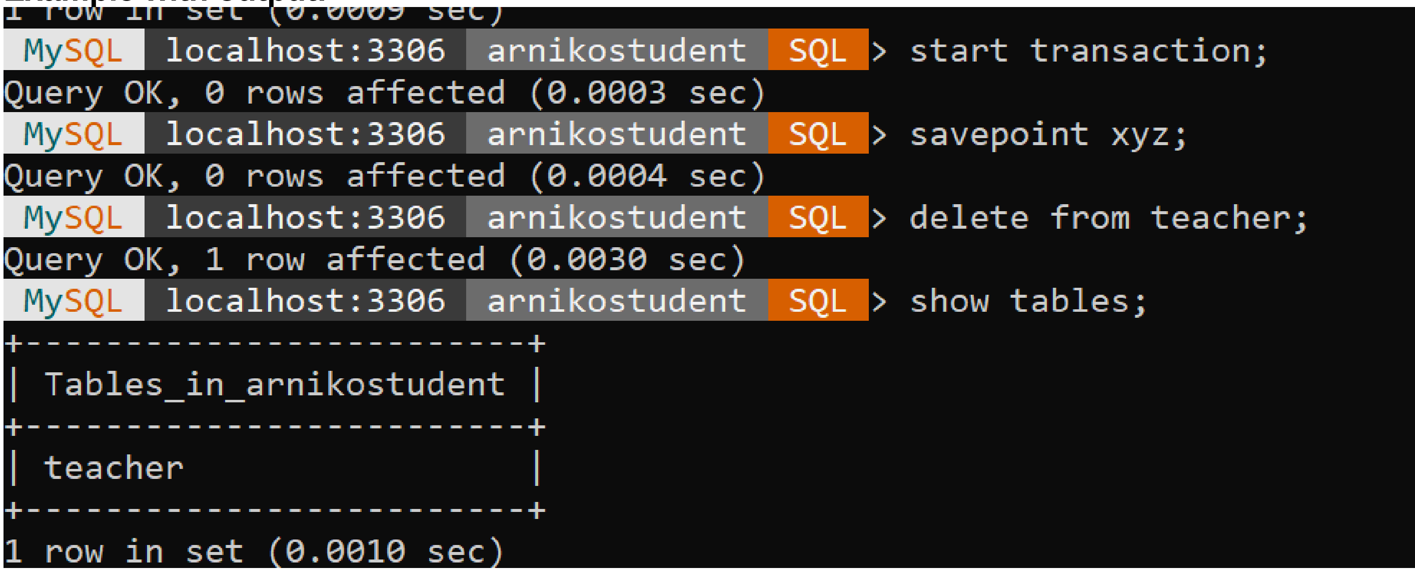
12) Save point Command

A **SAVEPOINT** is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

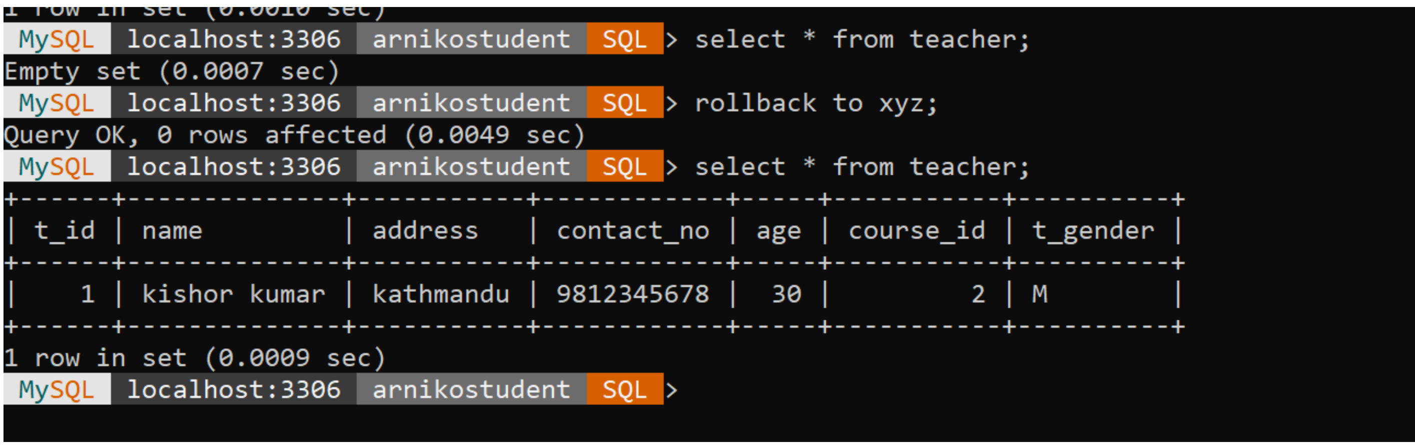
Syntax:

SAVEPOINT SAVEPOINT_NAME;

Example with output:



```
1 row in set (0.0009 sec)
 MySQL  localhost:3306  arnikostudent  SQL > start transaction;
Query OK, 0 rows affected (0.0003 sec)
 MySQL  localhost:3306  arnikostudent  SQL > savepoint xyz;
Query OK, 0 rows affected (0.0004 sec)
 MySQL  localhost:3306  arnikostudent  SQL > delete from teacher;
Query OK, 1 row affected (0.0030 sec)
 MySQL  localhost:3306  arnikostudent  SQL > show tables;
+-------------------------+
| Tables_in_arnikostudent |
+-------------------------+
| teacher                 |
+-------------------------+
1 row in set (0.0010 sec)
```



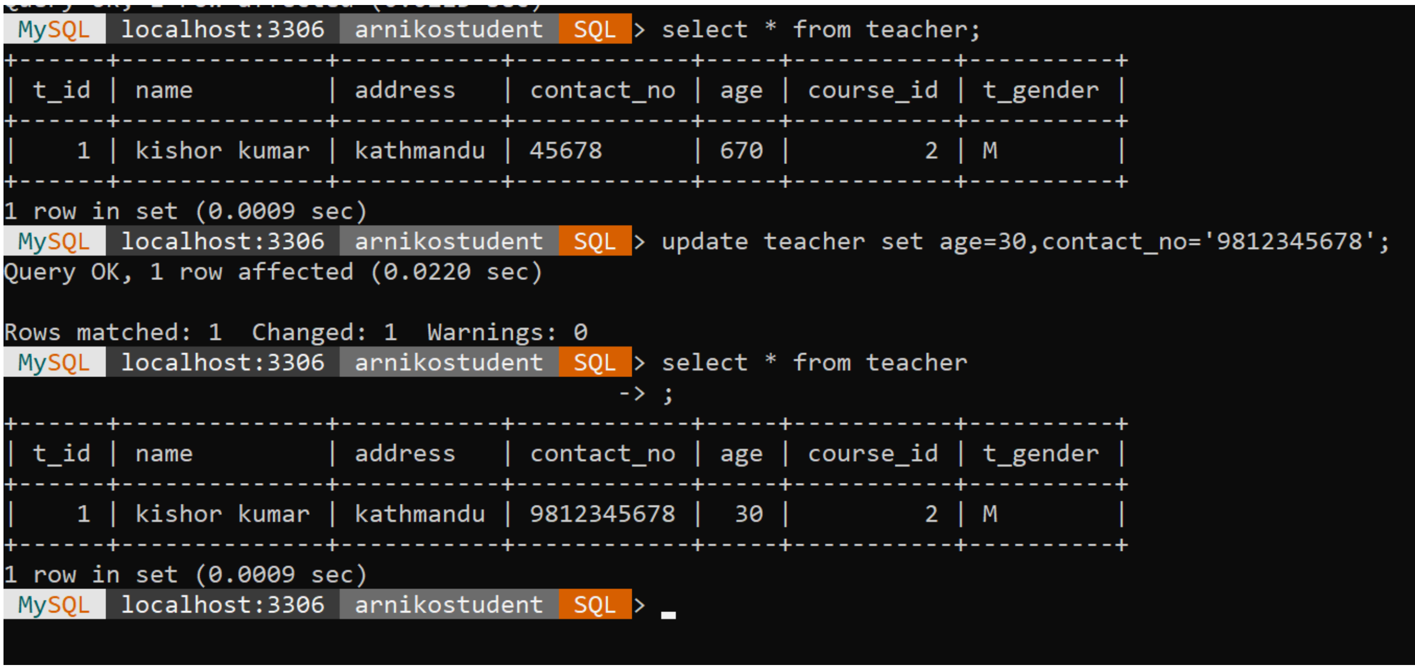
```
1 row in set (0.0010 sec)
 MySQL  localhost:3306  arnikostudent  SQL > select * from teacher;
Empty set (0.0007 sec)
 MySQL  localhost:3306  arnikostudent  SQL > rollback to xyz;
Query OK, 0 rows affected (0.0049 sec)
 MySQL  localhost:3306  arnikostudent  SQL > select * from teacher;
+------+--------------+----------+------------+-----+-----------+----------+
| t_id | name         | address  | contact_no | age | course_id | t_gender |
+------+--------------+----------+------------+-----+-----------+----------+
|    1 | kishor kumar | kathmandu | 9812345678 |  30 |         2 | M        |
+------+--------------+----------+------------+-----+-----------+----------+
1 row in set (0.0009 sec)
 MySQL  localhost:3306  arnikostudent  SQL >
```

13) Update
The UPDATE statement is used to modify the existing records in a table.

Syntax:

UPDATE table_name
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];

Example with output



```
 MySQL  localhost:3306  arnikostudent  SQL > select * from teacher;
+------+--------------+-----------+------------+-----+-----------+----------+
| t_id | name         | address   | contact_no | age | course_id | t_gender |
+------+--------------+-----------+------------+-----+-----------+----------+
|    1 | kishor kumar | kathmandu | 45678      | 670 |         2 | M        |
+------+--------------+-----------+------------+-----+-----------+----------+
1 row in set (0.0009 sec)
 MySQL  localhost:3306  arnikostudent  SQL > update teacher set age=30,contact_no='9812345678';
Query OK, 1 row affected (0.0220 sec)

Rows matched: 1  Changed: 1  Warnings: 0
 MySQL  localhost:3306  arnikostudent  SQL > select * from teacher
                                          -> ;
+------+--------------+-----------+------------+-----+-----------+----------+
| t_id | name         | address   | contact_no | age | course_id | t_gender |
+------+--------------+-----------+------------+-----+-----------+----------+
|    1 | kishor kumar | kathmandu | 9812345678 |  30 |         2 | M        |
+------+--------------+-----------+------------+-----+-----------+----------+
1 row in set (0.0009 sec)
 MySQL  localhost:3306  arnikostudent  SQL > _
```