

TCL Commands in SQL

* In SQL, TCL stands for **Transaction control language**.
* **COMMIT. ROLLBACK** and **SAVEPOINT** are the most commonly used TCL commands in SQL.

Now let us take a deeper dive into the TCL commands of SQL with the help of examples. All the queries in the examples will be written using the MySQL database.

1. COMMIT

COMMIT command in SQL is used to save all the transaction-related changes permanently to the disk. Whenever DDL commands such as INSERT, UPDATE and DELETE are used, the changes made by these commands are permanent only after closing the current session. So before closing the session, one can easily roll back the changes made by the DDL commands. Hence, if we want the changes to be saved permanently to the disk without closing the session, we will use the commit command.

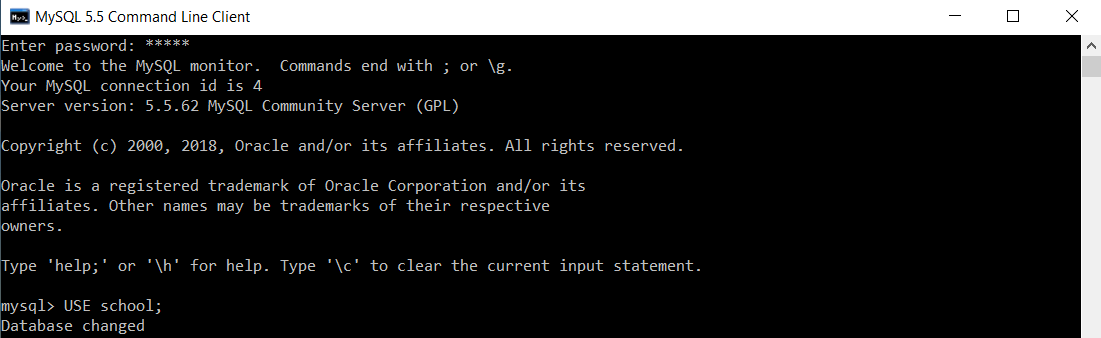
**Syntax:**

1. **COMMIT**;

**Example:**

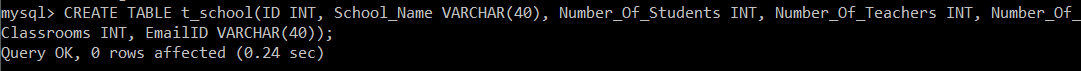
We will select an existing database, i.e., school.

1. mysql> USE school;



To create a table named t\_school, we will execute the following query:

1. mysql> **CREATE** **TABLE** t\_school(ID **INT**, School\_Name **VARCHAR**(40), Number\_Of\_Students **INT**, Number\_Of\_Teachers **INT**, Number\_Of\_Classrooms **INT**, EmailID **VARCHAR**(40));



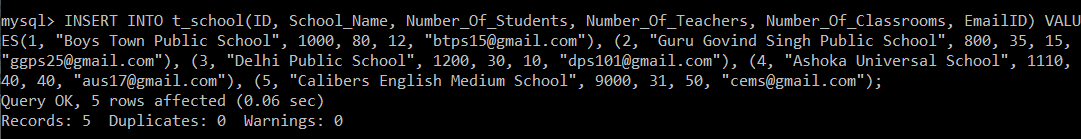
BEGIN / START TRANSACTION command is used to start the transaction.

1. mysql> START **TRANSACTION**;

TCL Commands in SQL

Now, we will execute the following query to insert multiple records at the same time in the t\_school table.

1. mysql> **INSERT** **INTO** t\_school(ID, School\_Name, Number\_Of\_Students, Number\_Of\_Teachers, Number\_Of\_Classrooms, EmailID) **VALUES**(1, "Boys Town Public School", 1000, 80, 12, "btps15@gmail.com"), (2, "Guru Govind Singh Public School", 800, 35, 15, "ggps25@gmail.com"), (3, "Delhi Public School", 1200, 30, 10, "dps101@gmail.com"), (4, "Ashoka Universal School", 1110, 40, 40, "aus17@gmail.com"), (5, "Calibers English Medium School", 9000, 31, 50, "cems@gmail.com");



We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

1. mysql> **SELECT** \***FROM** t\_school;

After executing the SELECT query on the t\_school table, you will get the following output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **School\_Name** | **Number\_Of\_Students** | **Number\_Of\_Teachers** | **Number\_Of\_Classrooms** | **EmailID** |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps15@gmail.com |
| 2 | Guru Govind Singh Public School | 800 | 35 | 15 | ggps25@gmail.com |
| 3 | Delhi Public School | 1200 | 30 | 10 | dps101@gmail.com |
| 4 | Ashoka Universal School | 1110 | 40 | 40 | aus17@gmail.com |
| 5 | Calibers English Medium School | 9000 | 31 | 50 | cems@gmail.com |

The output of the SELECT query shows that all the records are inserted successfully.

We will execute the COMMIT command to save the results of the operations carried on the t\_school table.

1. mysql> **COMMIT**;

TCL Commands in SQL

Autocommit is by default enabled in MySQL. To turn it off, we will set the value of autocommit as 0.

1. mysql> **SET** autocommit = 0;

TCL Commands in SQL

MySQL, by default, commits every query the user executes. But if the user wishes to commit only the specific queries instead of committing every query, then turning off the autocommit is useful.

2. SAVEPOINT

We can divide the database operations into parts. For example, we can consider all the insert related queries that we will execute consecutively as one part of the transaction and the delete command as the other part of the transaction. Using the SAVEPOINT command in SQL, we can save these different parts of the same transaction using different names. ***For example***, we can save all the insert related queries with the savepoint named INS. To save all the insert related queries in one savepoint, we have to execute the SAVEPOINT query followed by the savepoint name after finishing the insert command execution.

**Syntax:**

1. SAVEPOINT savepoint\_name;

3. ROLLBACK

While carrying a transaction, we must create savepoints to save different parts of the transaction. According to the user's changing requirements, he/she can roll back the transaction to different savepoints. *Consider a scenario*: We have initiated a transaction followed by the table creation and record insertion into the table. After inserting records, we have created a savepoint INS. Then we executed a delete query, but later we thought that mistakenly we had removed the useful record. Therefore in such situations, we have an option of rolling back our transaction. In this case, we have to roll back our transaction using the *ROLLBACK* command to the savepoint INS, which we have created before executing the DELETE query.

AD

**Syntax:**

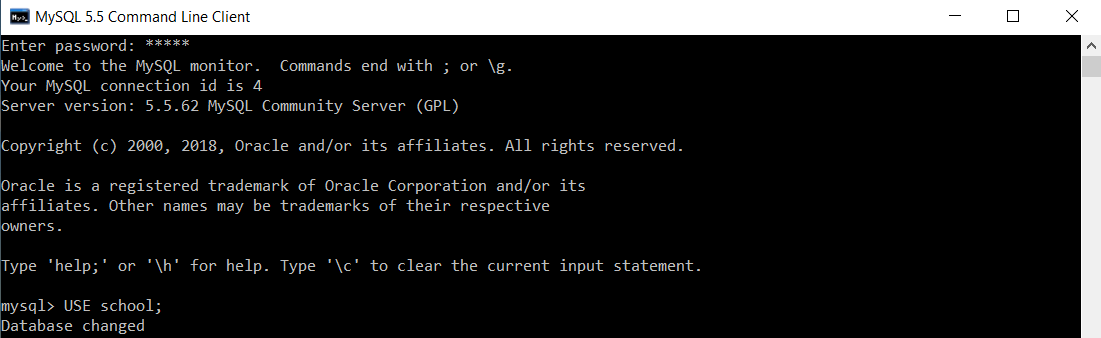
1. **ROLLBACK** **TO** savepoint\_name;

**Examples to understand the SAVEPOINT and ROLLBACK commands:**

**Example 1:**

We will select an existing database, i.e., school.

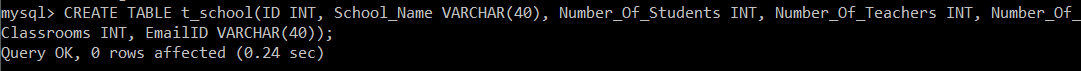
1. mysql> USE school;



To create a table named t\_school, we will execute the following query:

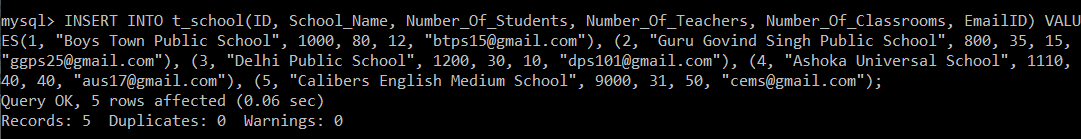
AD

1. mysql> **CREATE** **TABLE** t\_school(ID **INT**, School\_Name **VARCHAR**(40), Number\_Of\_Students **INT**, Number\_Of\_Teachers **INT**, Number\_Of\_Classrooms **INT**, EmailID **VARCHAR**(40));



Now, we will execute the following query to insert multiple records at the same time in the t\_school table.

1. mysql> **INSERT** **INTO** t\_school(ID, School\_Name, Number\_Of\_Students, Number\_Of\_Teachers, Number\_Of\_Classrooms, EmailID) **VALUES**(1, "Boys Town Public School", 1000, 80, 12, "btps15@gmail.com"), (2, "Guru Govind Singh Public School", 800, 35, 15, "ggps25@gmail.com"), (3, "Delhi Public School", 1200, 30, 10, "dps101@gmail.com"), (4, "Ashoka Universal School", 1110, 40, 40, "aus17@gmail.com"), (5, "Calibers English Medium School", 9000, 31, 50, "cems@gmail.com");



We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

1. mysql> **SELECT** \***FROM** t\_school;

After executing the SELECT query on the t\_school table, you will get the following output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **School\_Name** | **Number\_Of\_Students** | **Number\_Of\_Teachers** | **Number\_Of\_Classrooms** | **EmailID** |
| 1 | Boys Town Public School 1000 | 80 | 12 | btps15@gmail.com |  |
| 2 | Guru Govind Singh Public School | 800 | 35 | 15 | ggps25@gmail.com |
| 3 | Delhi Public School | 1200 | 30 | 10 | dps101@gmail.com |
| 4 | Ashoka Universal School | 1110 | 40 | 40 | aus17@gmail.com |
| 5 | Calibers English Medium School | 9000 | 31 | 50 | cems@gmail.com |

The output of the SELECT query shows that all the records are inserted successfully.

BEGIN / START TRANSACTION command is used to start the transaction.

1. mysql> START **TRANSACTION**;

TCL Commands in SQL

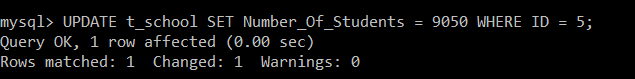
As we know, the SAVEPOINT command in SQL is used to save the different parts of the same transaction using different names. Consider till this point as one part of our transaction. We will save this part using a savepoint named Insertion.

1. mysql> SAVEPOINT Insertion;

TCL Commands in SQL

Now, we will execute the update command on the t\_school table to set the Number\_Of\_Students as 9050 for the record with ID 5.

1. mysql> **UPDATE** t\_school **SET** Number\_Of\_Students = 9050 **WHERE** ID = 5;



To verify that the record with ID 5 now has the Number\_Of\_Students as 9050, we will execute the SELECT query.

1. mysql> **SELECT** \***FROM** t\_school;

After executing the SELECT query on the t\_school table, you will get the following output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **School\_Name** | **Number\_Of\_Students** | **Number\_Of\_Teachers** | **Number\_Of\_Classrooms** | **EmailID** |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps15@gmail.com |
| 2 | Guru Govind Singh Public School | 800 | 35 | 15 | ggps25@gmail.com |
| 3 | Delhi Public School | 1200 | 30 | 10 | dps101@gmail.com |
| 4 | Ashoka Universal School | 1110 | 40 | 40 | aus17@gmail.com |
| 5 | Calibers English Medium School | 9050 | 31 | 50 | cems@gmail.com |

The output of the SELECT query shows that the record with ID 5 is updated successfully.

Consider the update operation as one part of our transaction. We will save this part using a savepoint named Updation.

1. mysql> SAVEPOINT Updation;

TCL Commands in SQL

Suddenly, our requirement changed, and we realized that we had updated a record that was not supposed to be. In such a scenario, we need to roll back our transaction to the savepoint, which was created prior to the execution of the UPDATE command.

1. mysql> **ROLLBACK** **TO** Insertion;

TCL Commands in SQL

We didn't need the updation carried on the record. Hence, we have rolled back to the savepoint named Insertion.

For confirming that we have got the same t\_school table that we had before carrying out the updation operation, we will again execute the SELECT query.

1. mysql> **SELECT** \***FROM** t\_school;

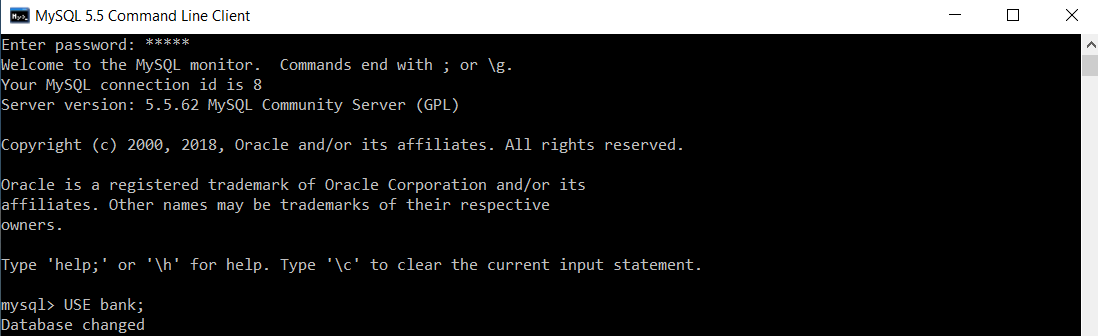
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **School\_Name** | **Number\_Of\_Students** | **Number\_Of\_Teachers** | **Number\_Of\_Classrooms** | **EmailID** |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps15@gmail.comm |
| 2 | Guru Govind Singh Public School | 800 | 35 | 15 | ggps25@gmail.comm |
| 3 | Delhi Public School | 1200 | 30 | 10 | dps101@gmail.comm |
| 4 | Ashoka Universal School | 1110 | 40 | 40 | aus17@gmail.comm |
| 5 | Calibers English Medium School | 9000 | 31 | 50 | cems@gmail.com |

The SELECT query output confirms that the transaction is now successfully rolled back to the savepoint 'Insertion'.

**Example 2:**

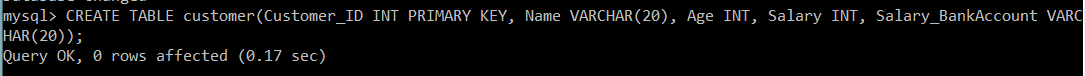
We will select an existing database, i.e., bank.

1. mysql> USE bank;



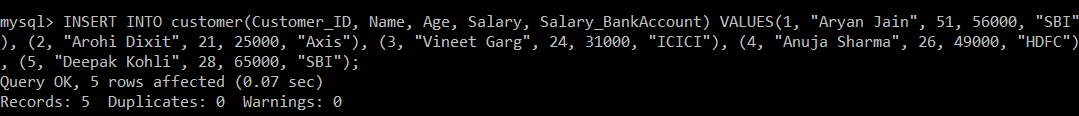
To create a table named customer, we will execute the following query:

1. mysql> **CREATE** **TABLE** customer(Customer\_ID **INT** **PRIMARY** **KEY**, **Name** **VARCHAR**(20), Age **INT**, Salary **INT**, Salary\_BankAccount **VARCHAR**(20));



Now, we will execute the following query to insert multiple records at the same time in the customer table.

1. mysql> **INSERT** **INTO** customer(Customer\_ID, **Name**, Age, Salary, Salary\_BankAccount) **VALUES**(1, "Aryan Jain", 51, 56000, "SBI"), (2, "Arohi Dixit", 21, 25000, "Axis"), (3, "Vineet Garg", 24, 31000, "ICICI"), (4, "Anuja Sharma", 26, 49000, "HDFC"), (5, "Deepak Kohli", 28, 65000, "SBI");



We will now execute the SELECT query to verify the execution of the INSERT INTO query executed above.

1. mysql> **SELECT** \***FROM** customer;

After executing the SELECT query on the t\_school table, you will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer\_ID** | **Name** | **Age** | **Salary** | **Salary\_BankAccount** |
| 1 | Aryan Jain | 51 | 56000 | SBI |
| 2 | Arohi Dixit | 21 | 25000 | Axis |
| 3 | Vineet Garg | 24 | 31000 | ICICI |
| 4 | Anuja Sharma | 26 | 49000 | HDFC |
| 5 | Deepak Kohli | 28 | 65000 | SBI |

The output of the SELECT query shows that all the records are inserted successfully.

*BEGIN / START TRANSACTION* command is used to start the transaction.

1. mysql> START **TRANSACTION**;

TCL Commands in SQL

As we know, the SAVEPOINT command in SQL is used to save the different parts of the same transaction using different names. Consider till this point as one part of our transaction. We will save this part using a savepoint named Insertion.

1. mysql> SAVEPOINT Insertion;

TCL Commands in SQL

We will execute the delete command on the customer table to remove the record with ID 5.

1. mysql> **DELETE** **FROM** customer **WHERE** Customer\_ID = 5;

TCL Commands in SQL

We will execute the SELECT query to verify that the record with ID 5 has been removed.

1. mysql> **SELECT** \***FROM** customer;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer\_ID** | **Name** | **Age** | **Salary** | **Salary\_BankAccount** |
| 1 | Aryan Jain | 51 | 56000 | SBI |
| 2 | Arohi Dixit | 21 | 25000 | Axis |
| 3 | Vineet Garg | 24 | 31000 | ICICI |
| 4 | Anuja Sharma | 26 | 49000 | HDFC |

The output of the SELECT query shows that the record with ID 5 is removed successfully.

Consider the delete operation as one part of our transaction. We will save this part using a savepoint named Deletion.

1. mysql> SAVEPOINT Deletion;

TCL Commands in SQL

Suddenly, our requirement changed, and we realized that we had deleted a record that was not supposed to be. In such a scenario, we need to roll back our transaction to the savepoint, which was created prior to the execution of the DELETE command.

1. mysql> **ROLLBACK** **TO** Insertion;

We didn't need the deletion carried on the record. Hence, we have rolled back to the savepoint named Insertion.

For confirming that we have got the same customer table that we had before carrying out the deletion operation, we will again execute the SELECT query.

1. mysql> **SELECT** \***FROM** customer;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer\_ID** | **Name** | **Age** | **Salary** | **Salary\_BankAccount** |
| 1 | Aryan Jain | 51 | 56000 | SBI |
| 2 | Arohi Dixit | 21 | 25000 | Axis |
| 3 | Vineet Garg | 24 | 31000 | ICICI |
| 4 | Anuja Sharma | 26 | 49000 | HDFC |
| 5 | Deepak Kohli | 28 | 65000 | SBI |

The SELECT query output confirms that the transaction is now successfully rolled back to the savepoint 'Insertion'.

**TCL**

1. **CREATE TABLE customer(Customer\_ID INT PRIMARY KEY, Name VARCHAR(20), Age INT, Salary INT, Salary\_BankAccount VARCHAR(20));**
2. **INSERT INTO customer(Customer\_ID, Name, Age, Salary, Salary\_BankAccount)**

**VALUES(1, "Aryan Jain", 51, 56000, "SBI"), (2, "Arohi Dixit", 21, 25000, "Axis"),**

**(3, "Vineet Garg", 24, 31000, "ICICI"), (4, "Anuja Sharma", 26, 49000, "HDFC"), (5, "Deepak Kohli", 28, 65000, "SBI");**

1. **select \* from Customer;**
2. **UPDATE CUSTOMER SET AGE =40 WHERE NAME LIKE 'DEEPAK%';**
3. **ROLLBACK; /\* NO EFFECT Because of Set autocommit is 1 enable\*/**
4. **INSERT INTO customer(Customer\_ID, Name, Age, Salary, Salary\_BankAccount) VALUES(11, "RAM", 51, 56000, "SBI");**
5. **INSERT INTO customer(Customer\_ID, Name, Age, Salary, Salary\_BankAccount) VALUES(12, "JAI", 51, 56000, "SBI");**
6. **select \* from Customer;**
7. **SET autocommit = 0; /\*Disable the Autocommit transanction set autocomit =0 \*/**
8. **START transaction;**
9. **DELETE FROM CUSTOMER WHERE NAME='JAI';**
10. **select \* from Customer;**
11. **ROLLBACK ;**
12. **select \* from Customer;**
13. **SAVEPOINT S1;**
14. **INSERT INTO customer VALUES(111, "KRISHNA", 53, 56000, "UBI");**
15. **SAVEPOINT S2;**
16. **INSERT INTO customer VALUES(222, "HARIA", 55, 76000, "PNB");**
17. **SELECT \* FROM CUSTOMER;**
18. **ROLLBACK TO S2;**
19. **SELECT \* FROM CUSTOMER;**