**TCL**

SQL, or Structured Query Language, is a database language that allows you to create a database and perform various operations. This is done using various types of SQL commands, such as DDL, DQL, DML, TCL, and DCL. But, this article will mainly focus on TCL commands in SQL.

2023\_02\_MicrosoftTeams-image-273.jpg

Before we dive deeper into the article (TCL commands in SQL), let’s quickly go over the list of topics listed under the table of contents (TOC) that we will cover in this article. Here’s the table of contents:

Table of Contents

What are TCL Commands in SQL

Types of TCL Commands in SQL

BEGIN TRANSACTION Command

COMMIT Command

ROLLBACK Command

SAVEPOINT Command

Advantages of TCL Commands

Disadvantages of TCL Commands

What are TCL Commands in SQL?

TCL Command Definition: The TCL (Transaction Control Language) commands in SQL help users manage and control a database’s transactions (changes) to maintain consistency.

In layman’s terms, we use TCL commands in order to manage transactions within a database. The transaction is a set of SQL statements executed on the data stored in the Database Management System. In simple terms, transactions are a set of one or more operations that are executed as a single unit of work. Either all or none of the operations should be completed.

Whenever any transaction happens, it is not permanent but rather temporary. TCL commands allow you to make those transactions (changes) permanent. TCL commands ensure that either all the changes are completed or none of them are.

**Types of TCL Commands in SQL**

There are mainly four types of TCL commands in SQL that help users to manage transactions. These four types of Transaction Control Language commands are:

BEGIN TRANSACTION Command

COMMIT Command

ROLLBACK Command

SAVEPOINT Command

In the next section, let’s explore these three commands in detail.

You can also explore: What is the Difference Between SQL and MySQL?

Must Explore: Understanding the Differences Between DDL and DML

BEGIN TRANSACTION Command

BEGIN TRANSACTION Command Definition: The BEGIN TRANSACTION command allows users to start a new transaction.

In order to understand it, let’s go through an example. Suppose you want to transfer money from one bank account to another. Hence, to accomplish this, both the debit and credit operations should happen together or not at all.

To ensure this, you can wrap both operations in a transaction by starting it with the BEGIN TRANSACTION command.

Here’s the code:

BEGIN TRANSACTION;

UPDATE customers SET balance = balance + 100 WHERE customer\_id = 1001;

COMMIT;

**COMMIT Command**

The COMMIT command allows users to save the changes made in the current transaction.

In order to understand it, let’s go through an example. Suppose you successfully transfer money from one bank account to another as part of a transaction. Once done, you can issue a COMMIT command to make the changes permanent.

Here’s the code:

BEGIN TRANSACTION;

UPDATE customers SET balance = balance + 100 WHERE customer\_id = 1001;

COMMIT;

**ROLLBACK Command**

The ROLLBACK command allows users to undo the changes made in the current transaction.

In order to understand it, let’s go through an example. Suppose, for certain reasons, the debit operation in the bank transfer failed. Hence, you can use the ROLLBACK command to undo the changes and return the system to its previous state.

Here’s the code:

BEGIN TRANSACTION;

UPDATE customers SET balance = balance + 100 WHERE customer\_id = 1001;

ROLLBACK;

**SAVEPOINT Command**

The SAVEPOINT command allows a user to create a point within a transaction to which you can later roll back.

In order to understand it, let’s go through an example. Suppose you are transferring money from one bank account to another. And you also want to be able to undo the debit operation separately from the credit operation. Hence, in order to accomplish this, you can use a savepoint to mark the point after the debit operation and later roll back to that point if needed.

Here’s the code:

BEGIN TRANSACTION;

SAVEPOINT savepoint\_1;

UPDATE customers SET balance = balance + 100 WHERE customer\_id = 1001;

ROLLBACK TO savepoint\_1;

COMMIT;

**Advantages of TCL Commands**

There are various advantages to using transaction control language commands. Let’s explore some of the most common and popular advantages of TCL commands, such as:

The TCL commands are easy to remember and handy.

Database users can very easily access, edit, and save changes to the database.

The coding required is not very complex. Hence, even new users can use these commands feasibly.

TCL commands ensure the consistency of the data in a database. Hence, if any operation within a transaction fails, the transaction is rolled back.

TCL commands ensure the durability of the data in a database. Hence, when a transaction is committed, its changes are made permanent, even if the system fails or restarts.

In TCL commands, transactions provide atomicity, i.e., all the operations within a transaction are executed as a single, indivisible unit. Hence, either all the operations are completed or none of them are.

You can also explore: ACID Properties in SQL

Disadvantages of TCL commands

The disadvantages of using transaction control language commands in SQL include the following:

For new users, identifying savepoints can be a lot confusing.

The ROLLBACK command is of no use when a user commits using the COMMIT command.

Transactions add overhead to the database system, requiring additional processing and storage resources. Thus, resulting in slower performance and increased resource consumption.

Transactions can add complexity to database systems if the database is large or complex. Hence, it will require careful management and monitoring to ensure the system remains stable and functional.

Conclusion

In this article, we have explored what transaction control language commands are. We have also explored its types with examples, advantages, and disadvantages in detail. If you have any queries related to the topic, please feel free to send your queries to us in the form of a comment. We will be happy to help.