T-SQL Transactions

www.tsql.info

A transaction is a single logical unit of work and it is composed of several sql server statements. The transaction begins with the first sql server statement executed and ends when the transaction is saved or rolled back.

T-SQL Transaction Statements

- BEGIN DISTRIBUTED TRANSACTION
- BEGIN TRANSACTION
- COMMIT TRANSACTION
- COMMIT WORK
- ROLLBACK TRANSACTION
- ROLLBACK WORK
- SAVE TRANSACTION

Begin Distributed Transaction

A distributed transaction is an operations transaction that can be run on multiple different servers.

Begin distributed transaction syntax:

```
BEGIN DISTRIBUTED { TRAN | TRANSACTION }
[ transaction_name | @transaction_name_variable ] ;
```

Begin distributed transaction example:

```
USE model;
GO
BEGIN DISTRIBUTED TRANSACTION;
--> Delete students from local instance.
DELETE students WHERE id = 7;
--> Delete students from remote instance.
DELETE RemoteServer.students WHERE id = 7;
COMMIT TRANSACTION;
GO
```

Begin Transaction

The Begin Transaction is the start point of an explicit transaction and increments @@TRANCOUNT by 1.

Begin transaction syntax:

```
BEGIN { TRAN | TRANSACTION }
[ { transaction_name | @transaction_name_variable }
[ WITH MARK [ 'transaction description' ] ] ];
```

Begin transaction example:

```
BEGIN TRANSACTION DelStudent WITH MARK N'Delete student';
GO
USE model;
GO
DELETE FROM students WHERE id = 7 and section = 'History';
GO
COMMIT TRANSACTION DelStudent;
GO
```

COMMIT Transaction

The COMMIT Transaction is the end point of a successful implicit or explicit transaction.

When @@TRANCOUNT is 1, all data modifications performed are commit and the resources held by the transaction are released, and decrements @@TRANCOUNT to 0. When @@TRANCOUNT is greater than 1, COMMIT TRANSACTION decrements @@TRANCOUNT only by 1 and the transaction stays active.

Commit transaction syntax:

```
COMMIT [ { TRAN | TRANSACTION } [ transaction_name |
@transaction_name_variable ] ]
[ WITH ( DELAYED_DURABILITY = { OFF | ON } ) ] ;
```

Commit transaction example:

```
USE model;
GO
BEGIN TRANSACTION;
GO
DELETE FROM students WHERE id = 7 and section = 'History';
GO
insert into students(id,first_name, last_name, gender,city, country, section)
values(7,'Ashley','THOMPSON','F','London','Liverpool', 'History');
GO
COMMIT TRANSACTION;
GO
```

COMMIT WORK

The COMMIT WORK transaction is the end point of a successful transaction.

Commit work syntax:

```
COMMIT [ WORK ] ;
```

Commit work example:

```
USE model;
GO
BEGIN TRANSACTION;
GO
DELETE FROM students WHERE id = 7 ;
GO
insert into students(id,first_name, last_name, gender,city, country, section)
values(7,'Ashley','THOMPSON','F','Liverpool','England', 'History');
GO
COMMIT WORK;
GO
```

ROLLBACK Transaction

The ROLLBACK Transaction is an operation that rolls back an unsuccessful explicit or implicit transaction to the beginning of the transaction or to a save point inside the transaction.

Rollback transaction syntax:

```
ROLLBACK { TRAN | TRANSACTION }
[ transaction_name | @transaction_name_variable
| savepoint_name | @savepoint_variable ] ;
```

Rollback transaction example:

```
USE model;
GO
BEGIN TRANSACTION;
GO
insert into students(id,first_name, last_name, gender,city, country, section)
values(8,'Alysia','MARTIN','F','Toronto','Canada', 'Biology');
GO
ROLLBACK TRANSACTION;
GO
```

ROLLBACK WORK

The ROLLBACK WORK is an transaction operation that rolls back a user specified transaction.

Rollback work syntax:

```
ROLLBACK [ WORK ] ;
```

Rollback work example:

```
USE model;
GO
BEGIN TRANSACTION;
GO
```

```
insert into students(id,first_name, last_name, gender,city, country, section)
values(8,'Alysia','MARTIN','F','Toronto','Canada', 'Biology');
GO
ROLLBACK WORK;
GO
```

SAVE Transaction

The SAVE transaction sets a save point within a transaction.

Save transaction syntax:

```
SAVE { TRAN | TRANSACTION } { savepoint_name | @savepoint_variable } ;
```

Save transaction example:

```
USE model;
GO
DECLARE @Counter INT;
SET @Counter = @@TRANCOUNT;
IF @Counter > 0
SAVE TRANSACTION my_savepoint;
ELSE
BEGIN TRANSACTION;
GO
insert into students(id,first_name, last_name, gender,city, country, section)
values(8,'Alysia','MARTIN','F','Toronto','Canada', 'Biology');
GO
COMMIT TRANSACTION;
GO
```

Resources:

www.tsql.info/transactions/transactions.php