Transaction in SQL Server

Definitions:

A **transaction** can be defined in many different ways and I've always had this question come up in interviews. Basically, a **transaction is a unit of work that is performed against a database.** This work can be performed **manually**, such as an UPDATE statement you issue in SQL Server Management Studio or an application that INSERTS data into the database. These are all transactions. SQL Server supports transaction control. **Transaction processing follows these steps:**

1. Begin transaction.
2. Process database commands.
3. Check for errors. If the error occurred then rollback the transaction else commit the transactions

Note that transaction controls are only used with DML commands.

- ✓ **BEGIN TRANSACTION** : the starting point of a transaction
- ✓ **ROLLBACK TRANSACTION** : roll back a transaction either because of a mistake or a failure
- ✓ **COMMIT TRANSACTION** - save changes to the database

Transactional Control Commands:

Transactional control commands are only used with the **DML Commands** such as INSERT, UPDATE and DELETE only. They cannot be used while creating tables or dropping them because these operations are automatically committed in the database.

COMMIT Command

- ✓ It is the transactional command used to save changes invoked by a transaction to the database.
- ✓ It used to save changes invoked by a transaction to the database.
- ✓ It saves all the transactions to the database since the last COMMIT or ROLLBACK command

ROLLBACK Command

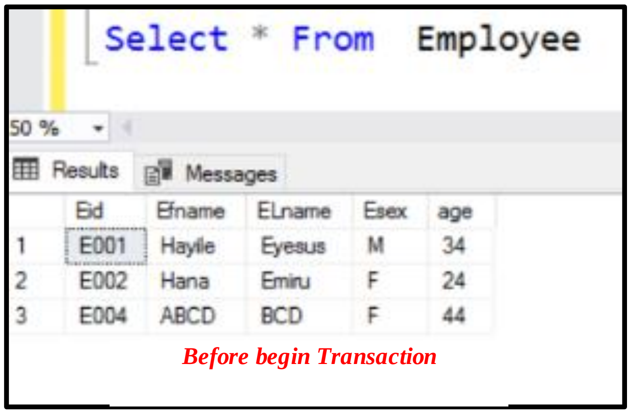
- ✓ It the transactional command used to undo transactions that have not already been saved to the database.
- ✓ It can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

Examples:

When creating a SQL Statement by default, the SQL Server will run this statement and immediately return the results. For Example, **Select * From Employee**

If you were to add BEGIN TRANSACTION (or BEGIN TRAN) before the statement it automatically makes the transaction explicit and holds a lock on the table until the transaction is either committed or rolled back.

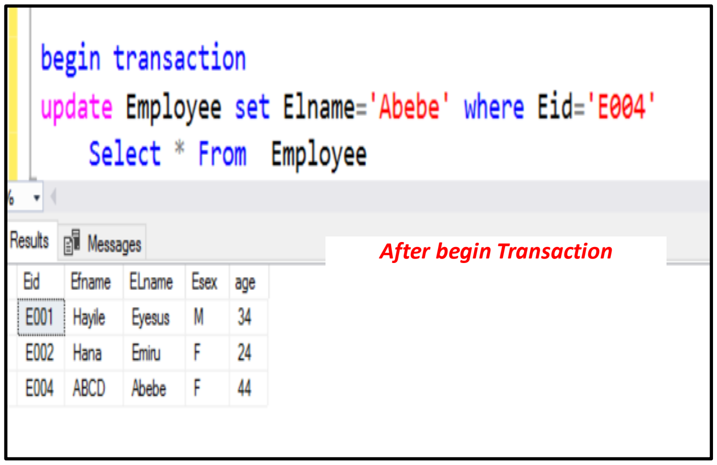
For example, when we UPDATE statement always explicitly use BEGIN TRAN to make sure my statement is correct and I get the correct number of results returned.



```
Select * From  Employee
```

	Eid	Efname	ELname	Esex	age
1	E001	Haylle	Eyesus	M	34
2	E002	Hana	Emiru	F	24
3	E004	ABCD	BCD	F	44

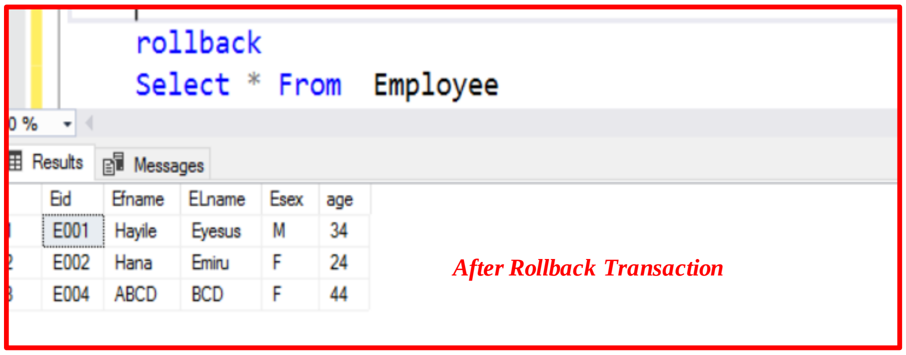
Before begin Transaction



```
begin transaction
update Employee set Elname='Abebe' where Eid='E004'
   Select * From  Employee
```

After begin Transaction

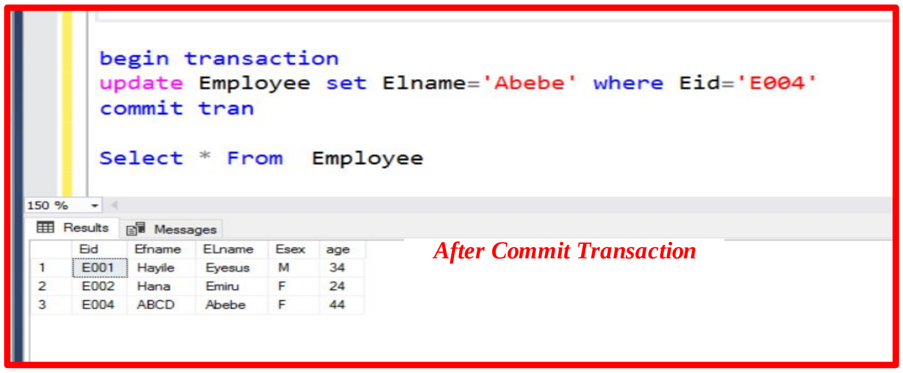
Eid	Efname	ELname	Esex	age
E001	Hayile	Eyesus	M	34
E002	Hana	Emiru	F	24
E004	ABCD	Abebe	F	44



```sql
rollback
Select * From  Employee
```

	Eid	Efname	ELname	Esex	age
1	E001	Hayile	Eyesus	M	34
2	E002	Hana	Emiru	F	24
3	E004	ABCD	BCD	F	44

After Rollback Transaction

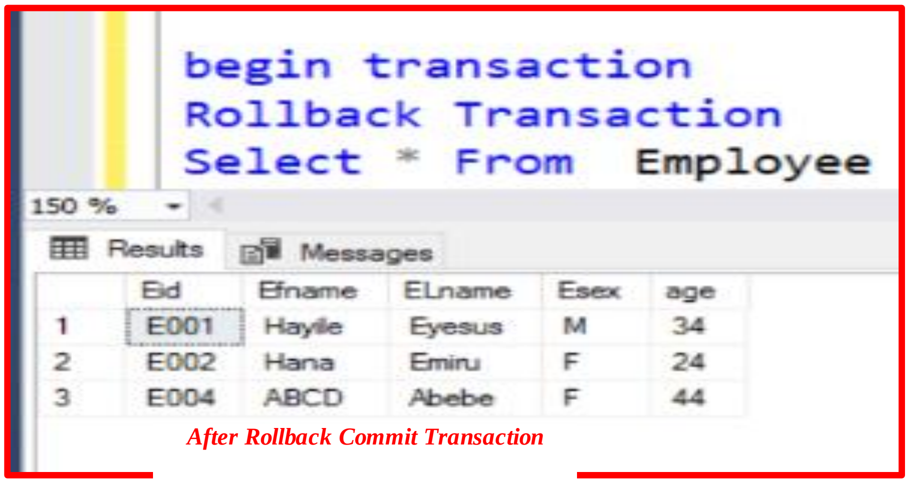


```sql
begin transaction
update Employee set Elname='Abebe' where Eid='E004'
commit tran

Select * From  Employee
```

	Eid	Efname	ELname	Esex	age
1	E001	Hayile	Eyesus	M	34
2	E002	Hana	Emiru	F	24
3	E004	ABCD	Abebe	F	44

After Commit Transaction



```sql
begin transaction
Rollback Transaction
Select * From  Employee
```

	Eid	Efname	ELname	Esex	age
1	E001	Hayile	Eyesus	M	34
2	E002	Hana	Emiru	F	24
3	E004	ABCD	Abebe	F	44

After Rollback Commit Transaction