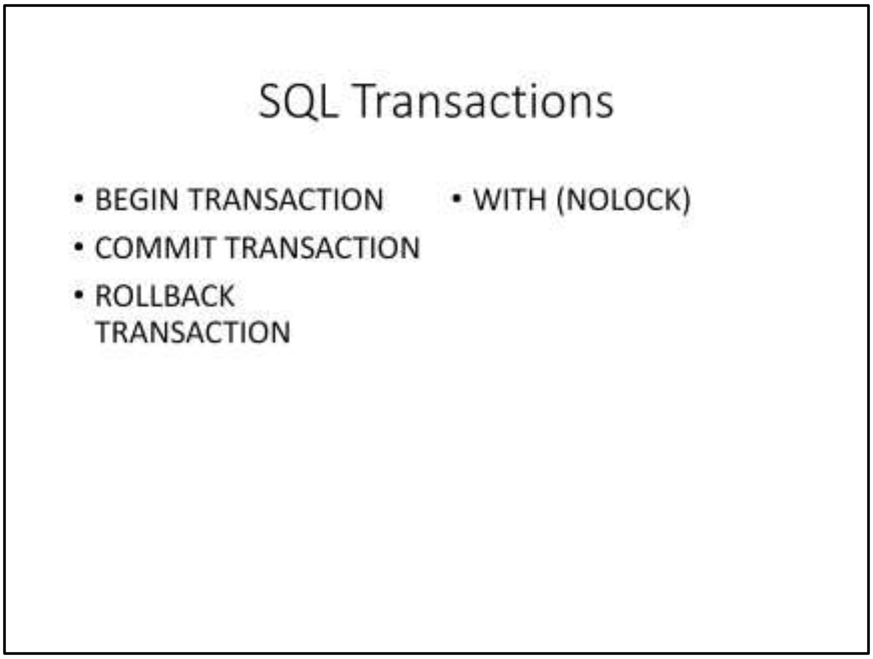
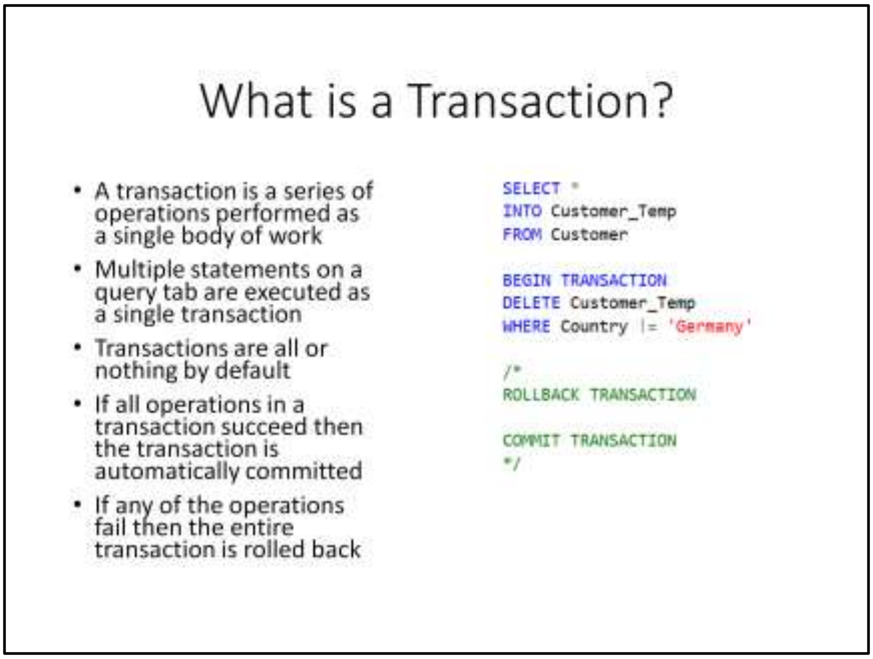


SQL Transactions

- BEGIN TRANSACTION
- COMMIT TRANSACTION
- ROLLBACK TRANSACTION
- WITH (NOLOCK)

This presentation will cover transactions. I will talk about how to manually start a transaction, commit and rollback transactions, and how to view the progress of a transaction.

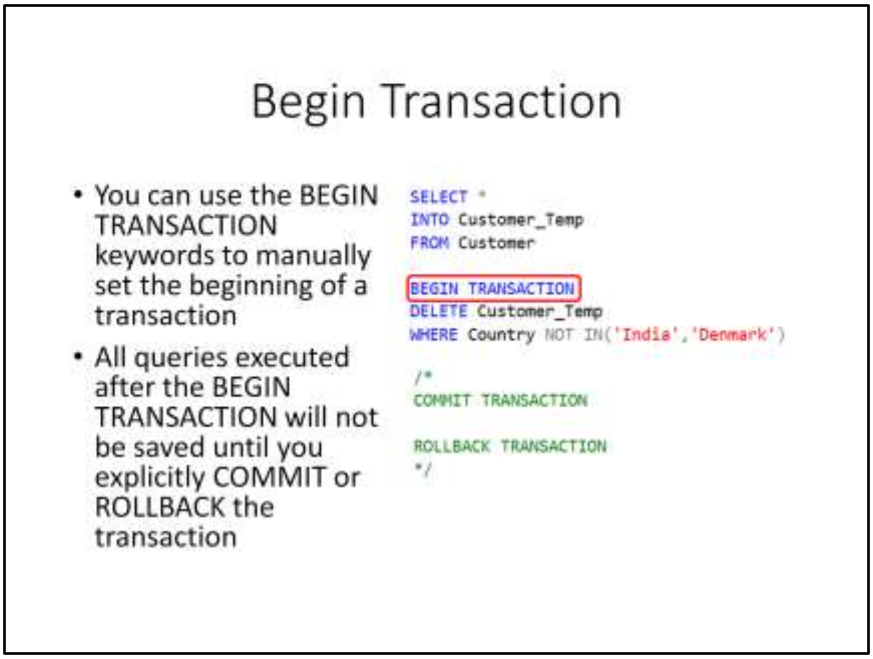


What is a Transaction?

- A transaction is a series of operations performed as a single body of work
- Multiple statements on a query tab are executed as a single transaction
- Transactions are all or nothing by default
- If all operations in a transaction succeed then the transaction is automatically committed
- If any of the operations fail then the entire transaction is rolled back

```
SELECT *
INTO Customer_Temp
FROM Customer

BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country != 'Germany'

/*
ROLLBACK TRANSACTION

COMMIT TRANSACTION
*/
```

A transaction is a series of operations performed as a single body of work. When you execute multiple SQL statements in a query execution tab, all of those statements are executed as a transaction. Statement execution is an all or nothing affair. If you have 5 statements and all 5 execute successfully, then the results of those statements will be automatically saved – or committed – to the database. On the other hand if any one of the 5 statements fails due to an error, then none of the statement results will be saved. Their results are rolled back.



Begin Transaction

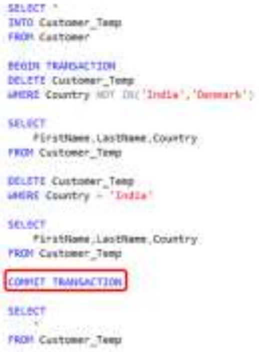
- You can use the BEGIN TRANSACTION keywords to manually set the beginning of a transaction
- All queries executed after the BEGIN TRANSACTION will not be saved until you explicitly COMMIT or ROLLBACK the transaction

```
SELECT *
INTO Customer_Temp
FROM Customer

BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country NOT IN('India','Denmark')

/*
COMMIT TRANSACTION

ROLLBACK TRANSACTION
*/
```

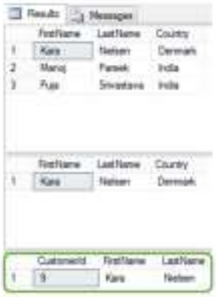
You can manually control when a transaction will be committed or rolled back by using the BEGIN TRANSACTION keywords before executing any of your statements. This puts the results of your statements into a holding pattern. They have not been committed nor have they been rolled back. Until you close your transaction with a COMMIT or ROLLBACK, you can continue to add statements to your transaction.

COMMIT TRANSACTION

- When the COMMIT TRANSACTION statement is executed, the operations in the transaction are saved to the database
- The operations cannot be undone



```
SELECT *
INTO Customer_Temp
FROM Customer

BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country NOT IN('India','Denmark')

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

DELETE Customer_Temp
WHERE Country = 'India'

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

COMMIT TRANSACTION

SELECT
    *
FROM Customer_Temp
```

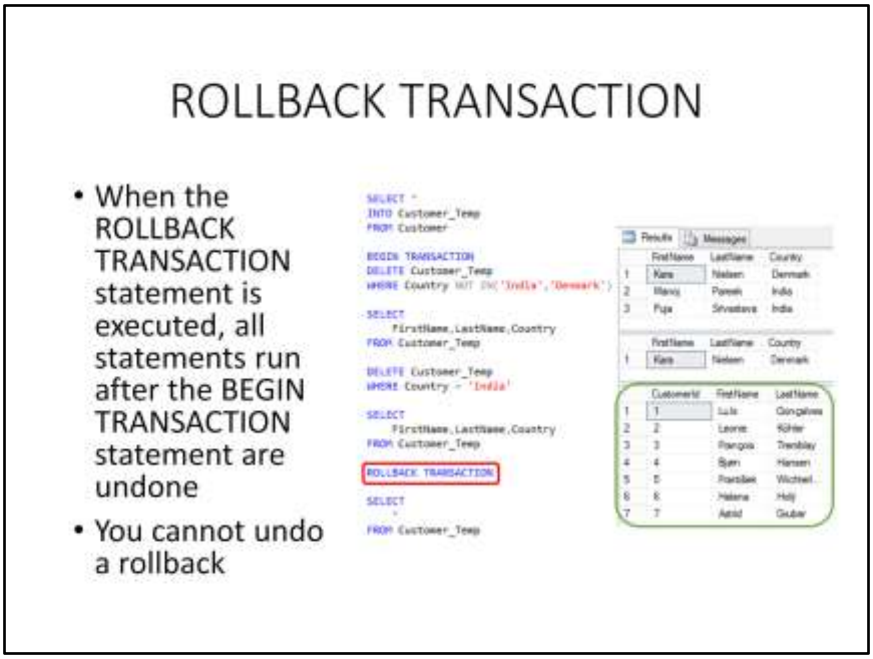


	FirstName	LastName	Country
1	Kara	Nielsen	Denmark
2	Manoj	Pareek	India
3	Puja	Srivastava	India

	FirstName	LastName	Country
1	Kara	Nielsen	Denmark

	CustomerId	FirstName	LastName
1	9	Kara	Nielsen

When you COMMIT a transaction, all of the changes you've made are saved to your database. Once committed, these changes cannot be undone.



ROLLBACK TRANSACTION

- When the ROLLBACK TRANSACTION statement is executed, all statements run after the BEGIN TRANSACTION statement are undone
- You cannot undo a rollback

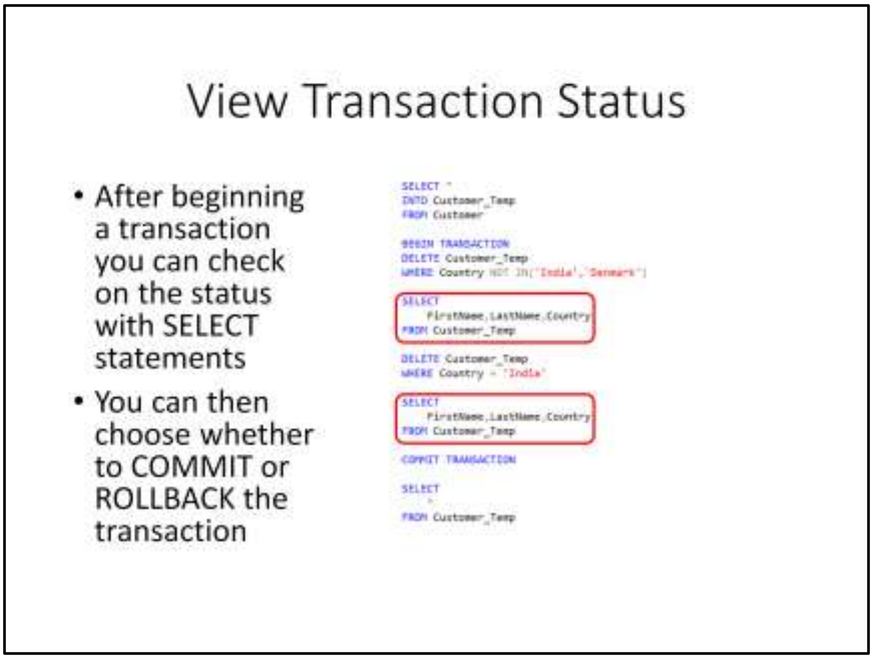
```
SELECT *
INTO Customer_Temp
FROM Customer

BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country NOT IN('India','Denmark')

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

DELETE Customer_Temp
WHERE Country = 'India'

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

ROLLBACK TRANSACTION

SELECT
    *
FROM Customer_Temp
```

	FirstName	LastName	Country
1	Kara	Nielsen	Denmark
2	Manoj	Pareek	India
3	Puja	Srivastava	India

	FirstName	LastName	Country
1	Kara	Nielsen	Denmark

	CustomerId	FirstName	LastName
1	1	Luís	Gonçalves
2	2	Leonie	Köhler
3	3	François	Tremblay
4	4	Bjørn	Hansen
5	5	František	Wichterl...
6	6	Helena	Holý
7	7	Astrid	Gruber

When you ROLLBACK a transaction, all of the changes you've made are undone. You cannot recover any work from a rollback.

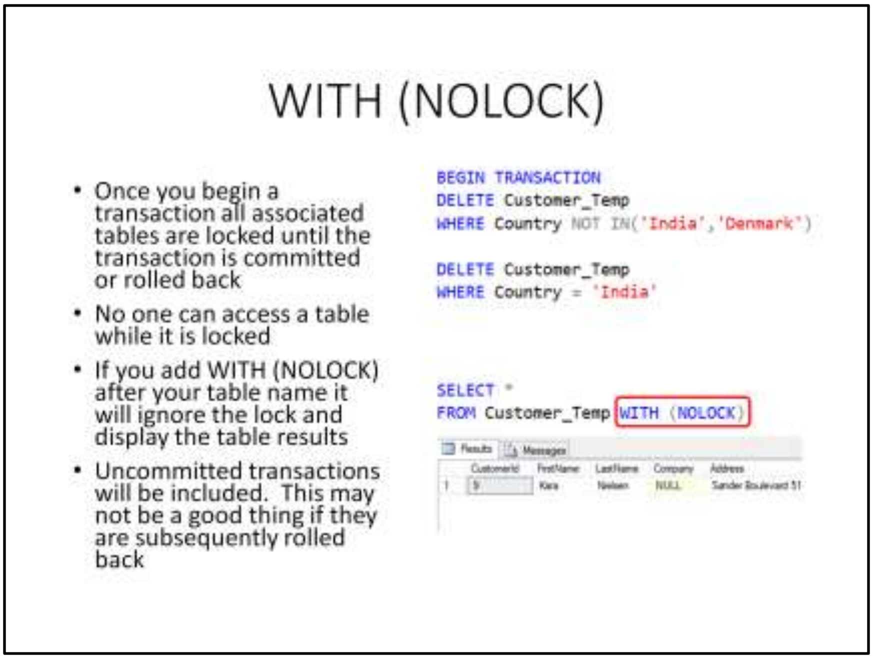


View Transaction Status

- After beginning a transaction you can check on the status with SELECT statements
- You can then choose whether to COMMIT or ROLLBACK the transaction

```
SELECT *
INTO Customer_Temp
FROM Customer

BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country NOT IN('India','Denmark')

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

DELETE Customer_Temp
WHERE Country = 'India'

SELECT
    FirstName,LastName,Country
FROM Customer_Temp

COMMIT TRANSACTION

SELECT
    *
FROM Customer_Temp
```

Once you have begun a transaction, you can run SELECT statements to view your progress. The SELECT statements will only work in the same session that the BEGIN TRANSACTION was run in. SELECT statements will not work in separate sessions. Each tab you open in Management Studio is a separate session.



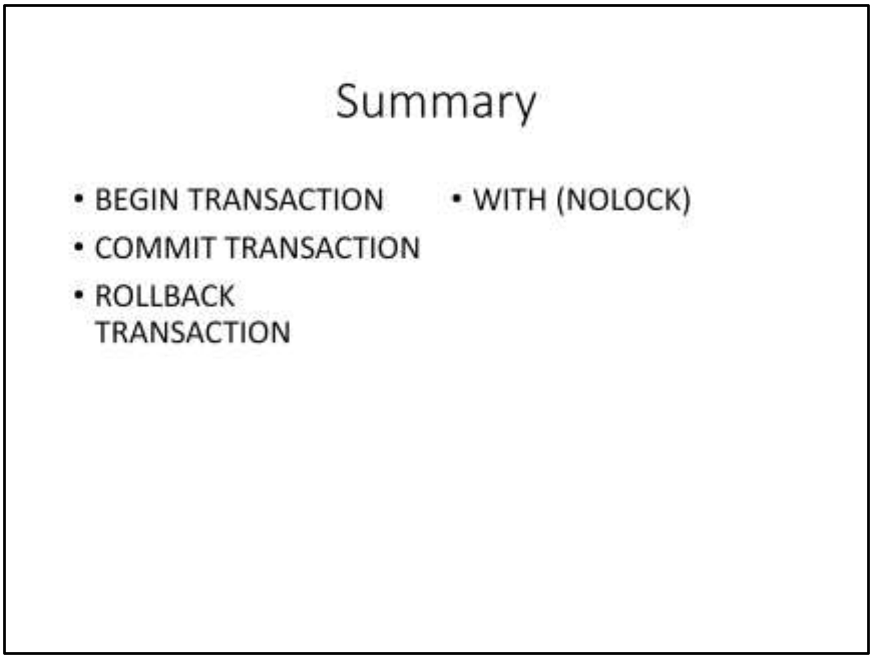
WITH (NOLOCK)

- Once you begin a transaction all associated tables are locked until the transaction is committed or rolled back
- No one can access a table while it is locked
- If you add WITH (NOLOCK) after your table name it will ignore the lock and display the table results
- Uncommitted transactions will be included. This may not be a good thing if they are subsequently rolled back

```sql
BEGIN TRANSACTION
DELETE Customer_Temp
WHERE Country NOT IN('India','Denmark')

DELETE Customer_Temp
WHERE Country = 'India'


SELECT *
FROM Customer_Temp WITH (NOLOCK)
```

```
   CustomerId  FirstName  LastName  Company  Address
1  5           Kara       Nielsen   NULL     Sander Boulevard 51
```

While a transaction is open, all of the objects it is interacting will are locked. This means that it is not possible for any other sessions to query against the objects until the transaction closes. In the example, if I do not commit the transaction or roll it back, then no one will be able to access the Customer_Temp table. There is an exception to this. If you include the table hint WITH (NOLOCK) after the table name, then your query will ignore the lock and return all committed and uncommitted data. You must be careful when using NOLOCK because the data you return may include records that were subsequently altered or deleted.



Summary

- BEGIN TRANSACTION
- COMMIT TRANSACTION
- ROLLBACK TRANSACTION
- WITH (NOLOCK)

This concludes the presentation on transactions.