

Exploring Transactions in SQL Server



Jared Westover

SQL ARCHITECT

@WestoverJared



Module Overview



Transactions are important to SQL Server

- Methods to start and finish a transaction
- Review the ACID principle
- Isolation levels

Exploring transactions in detail

- Three modes

Nested transactions

- Misconceptions
- Save points



ACID Principle

Guarantees a stable database

Atomic

- All or nothing

Consistent

- Keeping integrity in place

Isolation

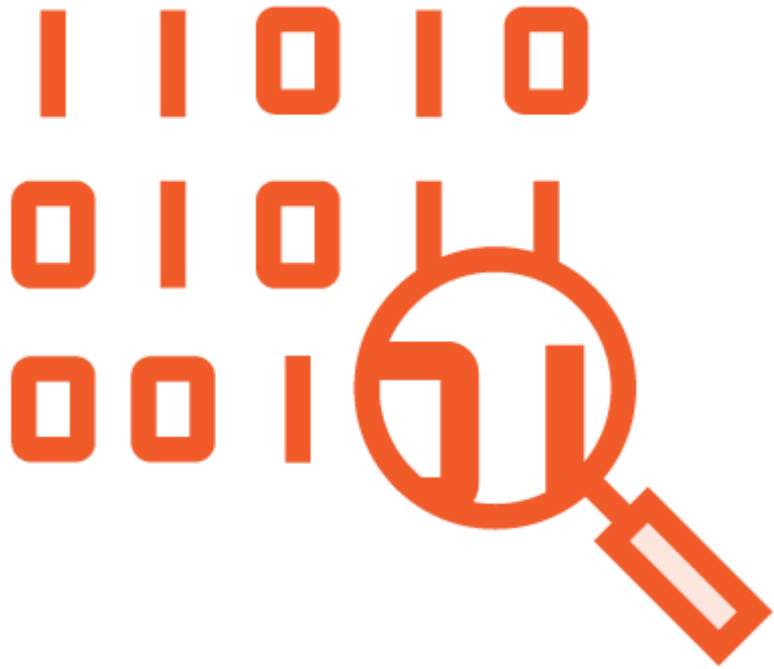
- Play well with others

Durable

- Failure recovery



Transaction Modes in SQL Server



Autocommit transactions

- Don't forget the where clause

Implicit transactions

- Only need to COMMIT/ROLLBACK

Explicit transactions

- Need BEGIN to start

```
UPDATE MyTable SET MyColumn = 'Special Value';
```

```
UPDATE MyOtherTable SET MyColumn = 'Other Special Value';
```

```
DELETE FROM MyTable;
```

Autocommit Transactions

Default mode for SQL Server unless changed



```
SET IMPLICIT_TRANSACTIONS ON;  
    UPDATE MyTable SET MyColumn = 'Special Value';  
    UPDATE MyOtherTable SET MyColumn = 'Special Value';  
COMMIT TRANSACTION;
```

Implicit Transactions

When enabled we don't need to specify a beginning



```
BEGIN TRANSACTION;
```

```
    UPDATE MyTable SET MyColumn = 'Special Value';
```

```
    UPDATE MyOtherTable SET MyColumn = 'Special Value';
```

```
COMMIT TRANSACTION;
```

Explicit Transactions

We specify the transaction beginning and ending



Demo



Review three modes of transactions

- The behavior of each
- Checking if implicit transactions are turned on
- Transactions with DDL statements



Isolation Levels

Keeps transactions separate

- Read Committed
- Can be changed at database or session level

Read Uncommitted

- Reading dirty pages
- NOLOCK hint
- Table locks

Great Pluralsight course on isolation levels

- SQL Server 2012 Querying (70-461)
Part 2



Demo



Multiple users updating the same data

- What happens if we change the isolation level
- What does NOLOCK allow us to do



Nested Transactions

Using nested transactions in SQL Server

- Paul Randal has written about this
- @@TRANCOUNT

A better approach to saving our work

- Save points
- Naming transactions



Demo



Exploring nested transactions

- Are they performing how we expect
- @@TRANCOUNT



Demo



Using save points in transactions

- Naming transactions



What We Covered



Explored why transactions are important

- Three types of transactions
- The ACID principle

Explored common isolation levels

- Prevents reading dirty pages

Nested transactions

- What really happens
- Save points

Next Module: Examining Error Handling Methods

