Understanding Transactions



Gerald Britton
IT SOLUTIONS DESIGNER, SQL SERVER SPECIALIST

@GeraldBritton www.linkedin.com/in/geraldbritton

Overview



Understanding responsibilities
Controlling transactions
Types of transactions
Saving your work, just in case

Locking basics

Understanding Responsibilities

SQL Programmers

Starting and ending

Follow business rules

Logical units of work

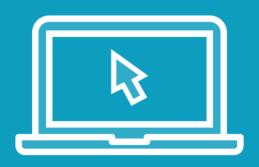
SQL Server Database Engine

Locking: Isolation

Logging: Durability

Management: Atomicity and Consistency

Demo



First Transactions

Transaction Types

Implicit Autocommit **Explicit**

Autocommit mode

BEGIN {TRANSACTION | TRAN}

COMMIT | ROLLBACK

SET IMPLICIT_TRANSACTIONS ON

COMMIT | ROLLBACK

MARS: Multiple Active Result Sets

Distributed transactions (MS DTC)

◄ Default mode

◄ Explicit Transaction mode

◄ Implicit Transaction mode

■ Batched-scoped

■ Distributed

Naming and Marking Transactions

```
BEGIN TRAN name or @name WITH MARK ['description']
```

```
COMMIT TRAN name or @name
WITH (DELAYED_DURABILITY =
{ OFF | ON } )
```

ROLLBACK TRAN name or @name

COMMIT WORK | ROLLBACK WORK

Using Transaction Savepoints

SAVE {TRANSACTION | TRAN}

name or @name

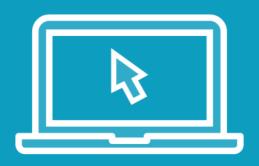
Location to return to

ROLLBACK TRAN[SACTION]

name or @name

COMMIT or ROLLBACK transaction

Demo



Savepoints

Summary



T-SQL transaction framework

Autocommit, explicit and implicit transactions

MARS, distributed transactions

Savepoints

Challenge!