Batch Examples

These examples are scripts that use SQL Server Management Studio Code editor and the **sqlcmd** utility GO command to define batch boundaries.

The following example creates a view. Because CREATE VIEW must be the only statement in a batch, the GO commands are required to isolate the CREATE VIEW statement from the USE and SELECT statements around it.

USE AdventureWorks2008R2;

GO

CREATE VIEW dbo.vProduct

AS

SELECT ProductNumber, Name

FROM Production.Product;

GO

SELECT \*

FROM dbo.vProduct;

GO

The following example shows several batches combined into one transaction. The BEGIN TRANSACTION and COMMIT statements delimit the transaction boundaries. The BEGIN TRANSACTION, USE, CREATE TABLE, SELECT, and COMMIT statements are all in their own single-statement batches. All of the INSERT statements are included in one batch.

BEGIN TRANSACTION

GO

USE AdventureWorks2008R2;

GO

CREATE TABLE dbo.mycompanies

(

id\_num int IDENTITY(100, 5),

company\_name nvarchar(100)

)

GO

INSERT mycompanies (company\_name)

VALUES (N'A Bike Store');

INSERT mycompanies (company\_name)

VALUES (N'Progressive Sports');

INSERT mycompanies (company\_name)

VALUES (N'Modular Cycle Systems');

INSERT mycompanies (company\_name)

VALUES (N'Advanced Bike Components');

INSERT mycompanies (company\_name)

VALUES (N'Metropolitan Sports Supply');

INSERT mycompanies (company\_name)

VALUES (N'Aerobic Exercise Company');

INSERT mycompanies (company\_name)

VALUES (N'Associated Bikes');

INSERT mycompanies (company\_name)

VALUES (N'Exemplary Cycles');

GO

SELECT id\_num, company\_name

FROM dbo.mycompanies

ORDER BY company\_name ASC;

GO

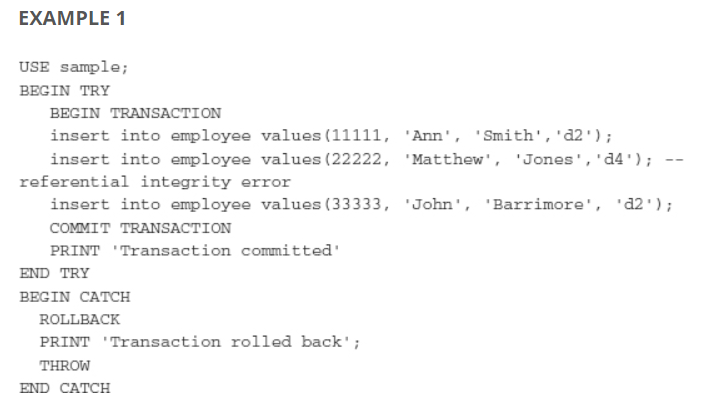
COMMIT;

GO

# SQL Server Exception Handling with TRY, CATCH and THROW

SQL Server 2012 introduces the third statement in relation to handling errors: THROW. This statement allows you to throw an exception caught in the exception handling block. Simply stated, the THROW statement is another return mechanism, which behaves similarly to the already described RAISEERROR statement.

Example 1 shows how exception handling with the TRY/CATCH/THROW works. It shows how you can use exception handling to insert all statements in a batch or to roll back the entire statement group if an error occurs. The example is based on the referential integrity between the **department** and **employee**tables. For this reason, you have to create both tables using the PRIMARY KEY and FOREIGN KEY constraints.



After the execution of the batch in Example 1, all three statements in the batch won’t be executed at all, and the output of this example is:

The execution of Example 1 works as follows. The first INSERT statement is executed successfully. Then, the second statement causes the referential integrity error. Because all three statements are written inside the TRY block, the exception is “thrown” and the exception handler starts the CATCH block. CATCH rolls back all statements and prints the corresponding message. After that the THROW statement returns the execution of the batch to the caller. For this reason, the content of the employee table won’t change.