Instructor Notes:

Add instructor notes here.

Instructor Notes:

Explain the lesson coverage

Lesson Objectives

- In this lesson, you will learn:
  - Introduction to Triggers
  - Defining Triggers
  - How Triggers Work

Instructor Notes:

4.2: Triggers

Introduction

- A trigger is a mechanism that is invoked when a particular action occurs on a particular table
- Each trigger has three general parts:
  - A name
  - The action
  - The execution
- The action of a trigger can be either a DML statement (INSERT, UPDATE, or DELETE) or a DDL statement
- Therefore, there are two trigger forms: DML triggers and DDL triggers
- The execution part of a trigger usually contains a stored procedure or a batch

What is trigger?
A trigger is a special type of stored procedure that is fired on an event-driven basis rather than by a direct call.
You can set up a trigger to fire when a data modification statement is issued—that is, an INSERT, UPDATE, or DELETE statement. SQL Server 2008 provides two types of triggers: *after triggers* and *instead-of triggers*

• By default triggers are after trigger i.e. executed after the event and NOT
before
• Views have a special type of Triggers called instead of triggers
• You can define multiple after triggers on a table for each event, and each
trigger can invoke many stored procedures as well as carry out many different actions based on the values of a given column of data.

However, you have only a minimum amount of control over the order in which triggers are fired on the same table.
• No trigger can be written for select statement because select does not
modify table data

Instructor Notes:

Explain uses with examples similar to one given in note pages

4.2: Triggers

Advantages of triggers

- Cascade Changes Through Related Tables in a Database
- Enforce More Complex Data Integrity check than constraints
- Implement complex business rules
- Triggers can be used to create business rules for an application
- Procedural integrity constraints are handled by triggers

Here are some common uses for triggers:
1. To maintain data integrity rules that extend beyond simple referential integrity
    - Validity for minimum balance in the bank account before withdrawal.
    - In ATM limit on amount one can withdarw /day
2. To implement a referential action, such as cascading deletes
    If we have employee and department tables and deptno is forignkey in employee table. Then one can write a after update trigger on department table to update employee table if we update any of the deptno in department table.

3.To maintain an audit record of changes
    One can write update trigger on item table to add history of rate in history_item table when we are updating rates in item tables
4. To invoke an external action, such as beginning a reorder process if inventory falls below a certain level or sending e-mail or a pager notification to someone who needs to perform an action because of data changes

Instructor Notes:

4.2: Triggers

Features

- Triggers Are Reactive; Constraints Are Proactive
- Constraints Are Checked First
- Tables Can Have Multiple Triggers for Any Action
- Table owners only can create triggers
- Triggers can be written for DDL and DML operations on the table
- SQL Triggers are AFTER triggers

Instructor Notes:

8.2: Triggers

After Trigger

- AFTER triggers are executed after the action of the INSERT,UPDATE, or DELETE statement is performed
- Specifying AFTER is the same as specifying FOR, which is the only option available in earlier versions of SQL Server
- You can define AFTER triggers only on tables
- When a trigger is defined as After, the trigger fires after the modification has passed all constraints
- Multiple After Triggers can be defined for one action like INSERT, UPDATE or DELETE
- If you have multiple trigger created for the same action, you can specify the order in which they can get fired
- You can achieve this with use of sp_settriggerorder system stored procedure
- In SQL Server triggers are by default After Trigger

Types of Triggers – After Trigger

AFTER triggers are executed after the action of the INSERT,UPDATE, or DELETE statement is performed. Specifying AFTER is the same as specifying FOR, which is the only option available in earlier versions of SQL Server. You can define AFTER triggers only on tables. When a trigger is defined as After, the trigger fires after the modification has passed all constraints. Multiple After Triggers can be defined for one action like INSERT, UPDATE or DELETE. If you have multiple trigger created for the same action, you can specify the order in which they can get fired. You can achieve this with use of sp_settriggerorder system stored procedure.

Example

These triggers run after an insert, update or delete on a table. They are not supported for views.

AFTER TRIGGERS can be classified further into three types as:

a. AFTER INSERT Trigger.
b. AFTER UPDATE Trigger.
c. AFTER DELETE Trigger.

Instructor Notes:

Give introduction about inserted and deleted table

8.2: Triggers

Tables used by DML trigger

- Tables used by trigger
  - inserted table: It always stores new values while execution of trigger
  - deleted table: Stores old values while execution of trigger
- In after trigger the statement which fires trigger gets logged which will give old and new values through inserted and deleted tables.

Trigger Name	Inserted	Deleted
INSERT	Newly inserted record	None
DELETE	None	Deleted record
UPDATE	Record with New Values	Record with Old Values

Inserted and Deleted tables

A trigger has access to the before image and after image of the data via the special pseudotables inserted and deleted. These two tables have the same set of columns as the underlying table being changed. You can check the before and after values of specific columns and take action depending on what you encounter. These tables are not physical structures—SQL Server constructs them from the transaction log.

Instructor Notes:

8.2: Triggers

Creating Triggers

```
CREATE TRIGGER Empl_Delete ON Employees
FOR DELETE
AS
IF exists (select 'a' from loan,deleted  where
   loan.EmpCode = deleted.EmpCOde)
BEGIN
  RAISERROR(
   'You cannot delete   employee having loan', 16, 1)
  ROLLBACK TRANSACTION
END
```

In this scenario the delete trigger will check if the employee is having a loan if so will rollback the transaction.

Instructor Notes:

4.2: Triggers

Instead of Trigger

- A trigger with an INSTEAD OF clause replaces the corresponding triggering action
- It is executed after the corresponding inserted and deleted tables are created, but before any integrity constraint or any other action is performed
- INSTEAD OF triggers can be created on tables as well as on views
- There are certain requirements on column values that are supplied by an INSTEAD OF trigger:
  - Values cannot be specified for computed columns
  - Values cannot be specified for columns with the TIMESTAMP data type
  - Values cannot be specified for columns with an IDENTITY property, unless the IDENTITY_INSERT option is set to ON

Instead Of Trigger

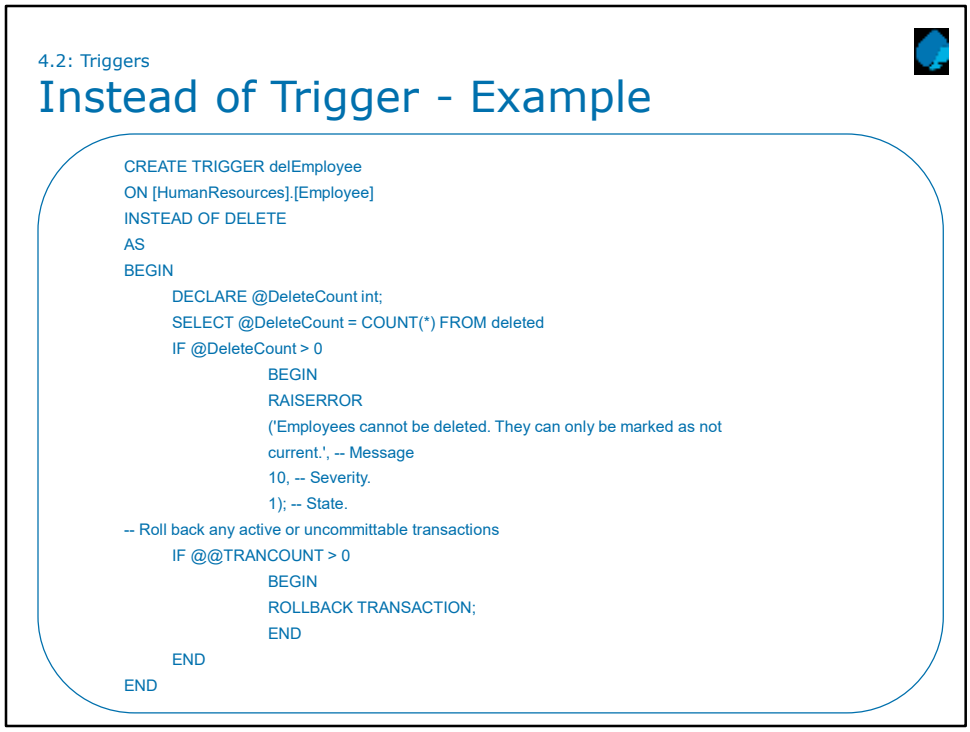
A trigger with an INSTEAD OF clause replaces the corresponding triggering action.
It is executed after the corresponding inserted and deleted tables are created, but
before any integrity constraint or any other action is performed. INSTEAD OF triggers can be created on tables as well as on non-updatable views. When a Transact-SQL statement references a view that has an INSTEAD OF trigger, the
database system executes the trigger instead of taking any action against any table.
The trigger always uses the information in the inserted and deleted tables built for
the view to create any statements needed to build the requested event.

There are certain requirements on column values that are supplied by an INSTEAD
OF trigger:

1.Values cannot be specified for computed columns.
2.Values cannot be specified for columns with the TIMESTAMP data type.
3.Values cannot be specified for columns with an IDENTITY property, unless the IDENTITY_INSERT option is set to ON.

These requirements are valid only for INSERT and UPDATE statements that reference a base table. An INSERT statement that references a view that has an
INSTEAD OF trigger must supply values for all non-nullable columns of that view.
The same is true for an UPDATE statement: an UPDATE statement that references
a view that has an INSTEAD OF trigger must supply values for each view column
that does not allow nulls and that is referenced in the SET clause.

Instructor Notes:



Instructor Notes:

4.2: Triggers

Altering and Dropping a trigger

- Altering a Trigger
  - Changes the definition without dropping the trigger
  - Can disable or enable a trigger
  - ALTER Trigger <trggername>
- Dropping a Trigger

  DROP Trigger <trggername>
- When a table is dropped Triggers are also automatically dropped

ALTERING trigger

ALTER TRIGGER trigname

Disable the trigger
Disables a DML or DDL trigger.

Syntax
DISABLE TRIGGER { [*schema* .] *trigger_name* [,*...n*] | ALL } ON { *object_name* | DATABASE | ALL SERVER } [;]

Arguments
schema_name
Is the name of the schema to which the trigger belongs. *schema_name* cannot be specified for DDL triggers.
trigger_name
Is the name of the trigger to be disabled.
ALL
Indicates that all triggers defined at the scope of the ON clause are disabled.

Specifying Trigger Firing Order

As I mentioned earlier, in most cases you shouldn't manipulate the trigger firing order. However, if you do want a particular trigger to fire first or fire last, you can use the *sp_settriggerorder* procedure to indicate this ordering. This procedure takes a trigger name, an order value (FIRST, LAST, or NONE), and an action (INSERT, UPDATE, or DELETE) as arguments. Setting a trigger's order value to NONE removes the ordering from a particular trigger.

Instructor Notes:

Demo

- Working with Triggers

Instructor Notes:

None

Summary

- In this lesson, you have learnt:
  - How to define Triggers
  - How the Triggers works

Instructor Notes:

Answers for the Review Questions:

Answer 1: True
Answer 2: Transaction
Answer 3: Explicit transaction

Review Question

- Question 1: A trigger with an INSTEAD OF clause replaces the corresponding triggering action
  - True
  - False