Transactions – Exercises

[1. Basic Transaction (INSERT and ROLLBACK) 1](#_Toc188515230)

[2. Transfer Funds Between Accounts 1](#_Toc188515231)

[3. Transaction with Error Handling 1](#_Toc188515232)

[4. Savepoint Example 2](#_Toc188515233)

[5. AFTER Trigger (Audit Logging) 2](#_Toc188515234)

[6. INSTEAD OF Trigger (Prevent Deletion) 2](#_Toc188515235)

[7. Trigger for Data Validation 3](#_Toc188515236)

[8. Cascade Update with Trigger 3](#_Toc188515237)

[9. Conditional Trigger 3](#_Toc188515238)

# Basic Transaction (INSERT and ROLLBACK)

Create a transaction to insert a new record into an Accounts table. If the balance is less than zero, rollback the transaction.

|  |
| --- |
| CREATE TABLE Accounts (  AccountID INT PRIMARY KEY,  AccountName NVARCHAR(50),  Balance DECIMAL(10, 2)  ); |

# Transfer Funds Between Accounts

Write a transaction to transfer $500 from Account A to Account B. Ensure the transaction is committed only if both updates are successful.

# Transaction with Error Handling

Create a transaction that updates the stock of a Products table. If any error occurs during the update, rollback the transaction and print an error message.

# Savepoint Example

Create a transaction that:

* Updates the quantity of multiple products.
* Creates a savepoint after the first product update.
* Rolls back to the savepoint if the second product update fails.

# AFTER Trigger (Audit Logging)

Create an AFTER INSERT trigger for an Orders table to log new orders into an OrderLog table.

|  |
| --- |
| CREATE TABLE Orders (  OrderID INT PRIMARY KEY,  CustomerID INT,  OrderDate DATE  );  CREATE TABLE OrderLog (  LogID INT IDENTITY(1,1) PRIMARY KEY,  OrderID INT,  LogDate DATETIME DEFAULT GETDATE()  ); |

# INSTEAD OF Trigger (Prevent Deletion)

Write an INSTEAD OF DELETE trigger to prevent deletions from an Employees table. Instead of deleting, insert the record into a DeletedEmployees table.

|  |
| --- |
| CREATE TABLE Employees (  EmployeeID INT PRIMARY KEY,  Name NVARCHAR(50),  Position NVARCHAR(50)  );  CREATE TABLE DeletedEmployees (  EmployeeID INT PRIMARY KEY,  Name NVARCHAR(50),  Position NVARCHAR(50),  DeletedDate DATETIME DEFAULT GETDATE()  ); |

# Trigger for Data Validation

Create a trigger on the Products table to prevent inserting products with a negative price. If an attempt is made, raise an error.

# Cascade Update with Trigger

Write an AFTER UPDATE trigger on a Departments table. When a department’s name is updated, automatically update the DepartmentName in the Employees table.

|  |
| --- |
| CREATE TABLE Departments (  DepartmentID INT PRIMARY KEY,  DepartmentName NVARCHAR(50)  );  CREATE TABLE Employees (  EmployeeID INT PRIMARY KEY,  Name NVARCHAR(50),  DepartmentID INT,  DepartmentName NVARCHAR(50)  ); |

# Conditional Trigger

Write an AFTER INSERT trigger for an Orders table. If the order value exceeds $1000, insert the record into a HighValueOrders table.

|  |
| --- |
| CREATE TABLE Orders (  OrderID INT PRIMARY KEY,  CustomerID INT,  TotalAmount DECIMAL(10, 2)  );  CREATE TABLE HighValueOrders (  OrderID INT PRIMARY KEY,  CustomerID INT,  TotalAmount DECIMAL(10, 2),  RecordedDate DATETIME DEFAULT GETDATE()  ); |