# Example 4

/*

In this scenario, we will use T-SQL variables to calculate and update loyalty points for customers based on their purchase history.
The calculation will be based on the total amount spent by the customer in a given year,
with a simple point system where 1 point is awarded for every $10 spent.

This example demonstrates how T-SQL variables can be used for more complex calculations
involving data from multiple tables, and how these results can be used to update records in a database,
showcasing the power and versatility of SQL in handling real-world business scenarios.

```
*/
-- Declare variables
DECLARE @CustomerID INT;
DECLARE @TotalSpent DECIMAL(10, 2);
DECLARE @PointsEarned INT;
DECLARE @CurrentYear INT = YEAR(GETDATE());

-- Initialize CustomerID
SET @CustomerID = 1; -- Example: Calculate points for CustomerID 1

-- Calculate total amount spent by the customer in the current year
SELECT @TotalSpent = SUM(Amount)
FROM Purchases
WHERE CustomerID = @CustomerID AND YEAR(PurchaseDate) =
@CurrentYear;

-- Calculate loyalty points (1 point for every $10 spent)
SET @PointsEarned = CAST(@TotalSpent / 10 AS INT);
```

```sql
-- Update loyalty points in Customers table
UPDATE Customers
SET LoyaltyPoints = LoyaltyPoints + @PointsEarned
WHERE CustomerID = @CustomerID;

-- Print the results
PRINT 'Loyalty Points Update for Customer ID: ' + CAST(@CustomerID AS VARCHAR);
PRINT 'Total Amount Spent in ' + CAST(@CurrentYear AS VARCHAR) + ': $' + CAST(@TotalSpent AS VARCHAR);
PRINT 'Loyalty Points Earned: ' + CAST(@PointsEarned AS VARCHAR);

-- This script calculates and updates the loyalty points for a customer based on their total spending in the current year.
```