

Lab 5

Joshua Evans | CS 276 | Due 2024-05-26

Part 1 - Textbook Questions:

The SQL script files associated with the following code snippets can be found here:

https://github.com/TheJoshuaEvans/LCC-CS276/blob/main/coursework/queries/module_07/Lab5Pt1Q16-30.sql

https://github.com/TheJoshuaEvans/LCC-CS276/blob/main/coursework/queries/module_07/Lab5Pt1Q31-34.sql

Question 16

```
CREATE TABLE [CUSTOMER](
  CUST_NUM INT IDENTITY(1000,1) NOT NULL,
  CUST_LNAME VARCHAR(30) NOT NULL,
  CUST_FNAME VARCHAR(30) NOT NULL,
  CUST_BALANCE DECIMAL(8,2) NOT NULL
);
```

Question 17

```
CREATE TABLE [INVOICE](
  INV_NUM INT IDENTITY(8000,1) NOT NULL,
  CUST_NUM INT,
  INV_DATE DATETIME NOT NULL,
  INV_AMOUNT DECIMAL(10,2) NOT NULL
);
```

Question 18

```
INSERT INTO [CUSTOMER] (CUST_LNAME, CUST_FNAME, CUST_BALANCE)
VALUES ('Smith', 'Jeanne', 1050.11);
INSERT INTO [CUSTOMER] (CUST_LNAME, CUST_FNAME, CUST_BALANCE)
VALUES ('Ortega', 'Juan', 840.92);
```

Question 19

```
INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1000, '2022-03-23', 235.89);
INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1001, '2022-03-23', 312.82);
INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1001, '2022-03-30', 528.10);
INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1000, '2022-04-12', 194.78);
INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1000, '2022-04-23', 619.44);
```

Question 22

```
INSERT INTO [CUSTOMER] (CUST_LNAME, CUST_FNAME, CUST_BALANCE)
```

```
VALUES ('Powers', 'Ruth', 500);
```

Question 23

```
ALTER TABLE [CUSTOMER] ADD CUST_DOB DATETIME NULL;
```

Question 24

```
UPDATE [CUSTOMER] SET CUST_DOB = '1989-03-05' WHERE CUST_NUM = 1000;
```

Question 25

```
UPDATE [CUSTOMER] SET CUST_DOB = '1988-12-22' WHERE CUST_NUM = 1001;
```

Question 26

```
CREATE TRIGGER [trg_updatecustbalance] ON [INVOICE] FOR INSERT AS
    DECLARE @INV_NUM INT, @INV_CUST_NUM INT, @NEW_BALANCE DECIMAL(8,2), @OLD_BALANCE DECIMAL(8,2);

    SELECT
        @INV_NUM = [inserted].INV_NUM,
        @INV_CUST_NUM = [inserted].CUST_NUM,
        @OLD_BALANCE = [CUSTOMER].CUST_BALANCE,
        @NEW_BALANCE = [inserted].[INV_AMOUNT] + [CUSTOMER].CUST_BALANCE
    FROM [inserted], [CUSTOMER]
    WHERE [inserted].CUST_NUM = [CUSTOMER].CUST_NUM;

    UPDATE [CUSTOMER] SET CUST_BALANCE = @NEW_BALANCE WHERE CUST_NUM = @INV_CUST_NUM;

    PRINT
        'Creation of invoice "' + CAST(@INV_NUM AS CHAR) + '"' +
        ' triggered update to customer number "' + CAST(@INV_CUST_NUM AS CHAR) + '". ' +
        'Balance changed from "' + CAST(@OLD_BALANCE AS CHAR) +
        '"' to '"' + CAST(@NEW_BALANCE AS CHAR) + '".';

GO

INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (1001, '2022-04-27', 225.40);
```

Question 27

```
CREATE PROCEDURE [prc_cust_add] AS
    INSERT INTO [CUSTOMER] (CUST_LNAME, CUST_FNAME, CUST_BALANCE)
    VALUES ('Rauthor', 'Peter', 0);
```

Question 28

```
CREATE PROCEDURE [prc_invoice_add] AS
    INSERT INTO [INVOICE] (CUST_NUM, INV_DATE, INV_AMOUNT)
    VALUES (1000, '2022-04-30', 301.72);
```

Question 29

```
CREATE TRIGGER [trg_updatecustbalance2] ON [INVOICE] FOR DELETE AS
    DECLARE @INV_NUM INT, @INV_CUST_NUM INT, @NEW_BALANCE DECIMAL(8,2), @OLD_BALANCE
DECIMAL(8,2);

    SELECT
        @INV_NUM = [deleted].INV_NUM,
        @INV_CUST_NUM = [deleted].CUST_NUM,
        @OLD_BALANCE = [CUSTOMER].CUST_BALANCE,
        @NEW_BALANCE = [CUSTOMER].CUST_BALANCE - [deleted].[INV_AMOUNT]
    FROM [deleted], [CUSTOMER]
    WHERE [deleted].CUST_NUM = [CUSTOMER].CUST_NUM;

    UPDATE [CUSTOMER] SET CUST_BALANCE = @NEW_BALANCE WHERE CUST_NUM = @INV_CUST_NUM;

    PRINT
        'Deletion of invoice ' + CAST(@INV_NUM AS CHAR) + ' ' +
        ' triggered update to customer number ' + CAST(@INV_CUST_NUM AS CHAR) + ' '. ' +
        'Balance decreased from ' + CAST(@OLD_BALANCE AS CHAR) +
        ' to ' + CAST(@NEW_BALANCE AS CHAR) + ' ';
```

Question 30

```
CREATE PROCEDURE [prc_inv_delete] @INV_NUM INT AS
    DELETE [INVOICE] WHERE INV_NUM = @INV_NUM;
GO

EXEC [prc_inv_delete] @INV_NUM = 8005;
EXEC [prc_inv_delete] @INV_NUM = 8006;
```

Question 31

```
CREATE TRIGGER [trg_line_total] ON [LINE] FOR INSERT AS
    UPDATE [LINE] SET
        LINE_TOTAL = [inserted].LINE_UNITS * [inserted].LINE_PRICE
    FROM [LINE] JOIN [inserted]
    ON [LINE].LINE_NUMBER = [inserted].LINE_NUMBER;
```

Question 32

```
CREATE TRIGGER [trg_line_prod] ON [LINE] FOR INSERT AS
    UPDATE [PRODUCT] SET
        P_QOH = P_QOH - 1
    FROM [PRODUCT] JOIN [inserted]
    ON [PRODUCT].P_CODE = [inserted].P_CODE
```

Question 33

```
CREATE PROCEDURE [prc_cus_balance_update] @INV_NUMBER INT AS
DECLARE @INV_SUBTOTAL DECIMAL(9,2), @INV_TAX DECIMAL (9,2)
SELECT
    @INV_SUBTOTAL = SUM(LINE_TOTAL),
    @INV_TAX = SUM(LINE_TOTAL) * 0.08
FROM [LINE]
WHERE INV_NUMBER = @INV_NUMBER

UPDATE [INVOICE] SET
    INV_SUBTOTAL = @INV_SUBTOTAL,
    INV_TAX = @INV_TAX,
    INV_TOTAL = @INV_SUBTOTAL + @INV_TAX
WHERE INV_NUMBER = @INV_NUMBER;
```

Question 34

```
CREATE PROCEDURE [prc_cus_balance_update] @INV_NUMBER INT AS
DECLARE @CUS_CODE INT = (
    SELECT CUS_CODE FROM [INVOICE] WHERE INV_NUMBER = @INV_NUMBER
);
DECLARE @ALL_INV_TOTAL DECIMAL(9,2) = (
    SELECT SUM(INV_TOTAL) FROM [INVOICE] WHERE CUS_CODE = @CUS_CODE
);

UPDATE [CUSTOMER] SET
    CUS_BALANCE = @ALL_INV_TOTAL
WHERE CUS_CODE = @CUS_CODE
```

Part 2 – SQL Practice

The SQL script file associated with the following screenshots can be found here:

https://github.com/TheJoshuaEvans/LCC-CS276/blob/main/coursework/queries/module_07/Lab5Pt2.sql

Section 1

```
DROP PROCEDURE IF EXISTS [spInsertCategory];

GO

CREATE PROCEDURE [spInsertCategory] @CategoryName VARCHAR(255) AS
    DECLARE @ItemsWithCategoryNameCount INT;
    SET @ItemsWithCategoryNameCount = (
        SELECT COUNT(*) FROM [Categories]
        WHERE CategoryName = @CategoryName
    );

    -- Ensure uniqueness of category names
    IF @ItemsWithCategoryNameCount > 0 BEGIN
        DECLARE @ErrorMessage VARCHAR(255);
        SET @ErrorMessage = 'Category "' + @CategoryName + '" already exists';
        THROW 51000, @ErrorMessage, 1;
    END

    INSERT INTO [Categories] (CategoryName) VALUES (@CategoryName);

GO

SELECT * FROM [Categories];
EXEC [spInsertCategory] @CategoryName = 'Cowbells';
EXEC [spInsertCategory] @CategoryName = 'Theremins';
SELECT * FROM [Categories];
```

RESULTS		
	CategoryID	CategoryName
1	2	Basses
2	3	Drums
3	1	Guitars
4	4	Keyboards
	CategoryID	CategoryName
1	2	Basses
2	1011	Cowbells
3	3	Drums
4	1	Guitars
5	4	Keyboards
6	1012	Theremins

MESSAGES	
Timestamp	Message
[11:41:28 PM]	Started executing query at Line 9 Commands completed successfully.
[11:41:28 PM]	Started executing query at Line 12 Commands completed successfully.
[11:41:28 PM]	Started executing query at Line 28 (4 rows affected) (1 row affected) (1 row affected) (6 rows affected) Total execution time: 00:00:00.021

Section 2

```
DROP FUNCTION IF EXISTS [fnDiscountPrice]

GO

CREATE FUNCTION [fnDiscountPrice] (@ItemID INT) RETURNS MONEY AS
BEGIN
    DECLARE @DiscountPrice INT;
    SELECT @DiscountPrice = ItemPrice - DiscountAmount
        FROM OrderItems WHERE ItemID = @ItemID;

    RETURN @DiscountPrice
END;
GO

DECLARE @DiscountPriceResult1 MONEY, @DiscountPriceResult2 MONEY;
EXEC @DiscountPriceResult1 = [fnDiscountPrice] @ItemID = 1;
EXEC @DiscountPriceResult2 = [fnDiscountPrice] @ItemID = 2;
SELECT
    'Discount Price 1' = @DiscountPriceResult1,
    'Discount Price 2' = @DiscountPriceResult2;
```

RESULTS		
	Discount Price 1	Discount Price 2
1	839.00	304.00

MESSAGES	
Timestamp	Message
[11:48:38 PM]	Started executing query at Line 39 Commands completed successfully.
[11:48:38 PM]	Started executing query at Line 42 Commands completed successfully.
[11:48:38 PM]	Started executing query at Line 51 (1 row affected) Total execution time: 00:00:00.013

Section 3

```
DROP FUNCTION IF EXISTS [fnItemTotal]

GO

CREATE FUNCTION [fnItemTotal] (@ItemID INT) RETURNS MONEY AS
BEGIN
    DECLARE @ItemTotal MONEY, @DiscountPrice MONEY;
    EXEC @DiscountPrice = [fnDiscountPrice] @ItemID = @ItemID;
    SELECT @ItemTotal = @DiscountPrice * Quantity
        FROM OrderItems WHERE ItemID = @ItemID;

    RETURN @ItemTotal;
END;
GO

DECLARE @ItemTotalResult1 MONEY, @ItemTotalResult33 MONEY;
EXEC @ItemTotalResult1 = [fnItemTotal] @ItemID = 1;
EXEC @ItemTotalResult33 = [fnItemTotal] @ItemID = 33;
SELECT
    'Item Total 1' = @ItemTotalResult1,
    'Item Total 33' = @ItemTotalResult33;
```

RESULTS		
	Item Total 1	Item Total 33
1	839.00	2800.00

MESSAGES	
Timestamp	Message
[11:53:42 PM]	Started executing query at Line 66 Commands completed successfully.
[11:53:42 PM]	Started executing query at Line 69 Commands completed successfully.
[11:53:42 PM]	Started executing query at Line 79 (1 row affected) Total execution time: 00:00:00.013

Section 4

```
DROP PROCEDURE IF EXISTS [spInsertProduct];

GO

CREATE PROCEDURE [spInsertProduct]
    @ProductCode VARCHAR(10),
    @ProductName VARCHAR(255),
    @ListPrice MONEY,
    @DiscountPercent MONEY,
    @CategoryID INT = NULL
AS
    -- Validation
    IF @ListPrice < 0 BEGIN
        DECLARE @ListPriceInvalidErrorMessage VARCHAR(255) =
            'List price cannot be less than 0. ' +
            'Provided list price: ' + CAST (@ListPrice AS VARCHAR);
        THROW 51001, @ListPriceInvalidErrorMessage, 1;
    END

    IF @DiscountPercent < 0 BEGIN
        DECLARE @DiscountPercentInvalidErrorMessage VARCHAR(255) =
            'Discount percent cannot be less than 0. ' +
            'Provided discount percent: ' + CAST (@DiscountPercent AS VARCHAR);
        THROW 51002, @DiscountPercentInvalidErrorMessage, 1;
    END

    -- Perform the insert!
    INSERT INTO [Products] (
        ProductCode, ProductName, ListPrice, DiscountPercent, CategoryID,
        Description, DateAdded
    ) VALUES (
        @ProductCode, @ProductName, @ListPrice, @DiscountPercent, @CategoryID,
        '', GETDATE()
    );

GO
```



```

-- Initial State
SELECT TOP 8 * FROM Products ORDER BY ProductID DESC;

-- Test Validation
BEGIN TRY
    EXEC [spInsertProduct]
        @ProductCode = 'prod_001', @ProductName = 'More Cowbell',
        @ListPrice = -10, @DiscountPercent = 0;
END TRY
BEGIN CATCH PRINT 'Error message: ' + ERROR_MESSAGE() END CATCH

BEGIN TRY
    EXEC [spInsertProduct]
        @ProductCode = 'prod_002', @ProductName = 'Even More Cowbell!!!',
        @ListPrice = 100, @DiscountPercent = -33;
END TRY
BEGIN CATCH PRINT 'Error message: ' + ERROR_MESSAGE() END CATCH

-- Test functionality
DELETE FROM [Products] WHERE ProductCode IN ('prod_001', 'prod_002');

EXEC [spInsertProduct]
    @ProductCode = 'prod_001', @ProductName = 'More Cowbell',
    @ListPrice = 10, @DiscountPercent = 0;

EXEC [spInsertProduct]
    @ProductCode = 'prod_002', @ProductName = 'Even More Cowbell!!!',
    @ListPrice = 100, @DiscountPercent = 33;

-- Final State
SELECT TOP 8 * FROM Products ORDER BY ProductID DESC;

```

RESULTS								CTRL +
	ProductID	CategoryID	ProductCode	ProductName	Description	ListPrice	DiscountPercent	DateAdded
1	10	3	tama	Tama 5-Piece D...	The Tama 5-pie...	799.99	15.00	2019-12-30 13:...
2	9	3	ludwig	Ludwig 5-piece...	This product in...	699.99	30.00	2019-12-30 12:...
3	8	2	hofner	Hofner Icon	With authentic ...	499.99	25.00	2019-12-30 14:...
4	7	2	precision	Fender Precision	The Fender Pre...	799.99	30.00	2019-12-30 11:...
5	6	1	rodriguez	Rodriguez Cab...	Featuring a car...	415.00	39.00	2019-12-30 14:...
6	5	1	washburn	Washburn D10S	The Washburn ...	299.00	0.00	2019-11-01 13:...
7	4	1	fg700s	Yamaha FG700S	The Yamaha FG...	489.99	38.00	2019-11-01 11:...
8	3	1	sg	Gibson SG	This Gibson SG...	2517.00	52.00	2019-07-04 11:...
	ProductID	CategoryID	ProductCode	ProductName	Description	ListPrice	DiscountPercent	DateAdded
1	22	NULL	prod_002	Even More Cow...		100.00	33.00	2024-05-24 06:...
2	21	NULL	prod_001	More Cowbell		10.00	0.00	2024-05-24 06:...
3	10	3	tama	Tama 5-Piece D...	The Tama 5-pie...	799.99	15.00	2019-12-30 13:...
4	9	3	ludwig	Ludwig 5-piece...	This product in...	699.99	30.00	2019-12-30 12:...
5	8	2	hofner	Hofner Icon	With authentic ...	499.99	25.00	2019-12-30 14:...
6	7	2	precision	Fender Precision	The Fender Pre...	799.99	30.00	2019-12-30 11:...
7	6	1	rodriguez	Rodriguez Cab...	Featuring a car...	415.00	39.00	2019-12-30 14:...
8	5	1	washburn	Washburn D10S	The Washburn ...	299.00	0.00	2019-11-01 13:...
MESSAGES								CTRL +
Timestamp	Message							
[11:56:52 PM]	Started executing query at Line 101 Commands completed successfully.							
[11:56:52 PM]	Started executing query at Line 104 Commands completed successfully.							
[11:56:52 PM]	Started executing query at Line 135 (8 rows affected) Error message: List price cannot be less than 0. Provided list price: -10.00 Error message: Discount percent cannot be less than 0. Provided discount percent: -33.00 (0 rows affected) (1 row affected) (1 row affected) (8 rows affected) (0 rows affected) (2 rows affected)							
Total execution time: 00:00:00.028								

Section 5

```
USE MyGuitarShop;
DROP PROCEDURE IF EXISTS [spUpdateProductDiscount];

GO

CREATE PROCEDURE [spUpdateProductDiscount]
    @ProductID INT,
    @NewDiscountPercent MONEY
AS
    -- Validation
    IF @NewDiscountPercent < 0 BEGIN
        DECLARE @ListPriceInvalidErrorMessage VARCHAR(255) =
            'New discount percent cannot be less than 0. ' +
            'Provided discount percent: ' + CAST (@NewDiscountPercent AS VARCHAR);
        THROW 51003, @ListPriceInvalidErrorMessage, 1;
    END

    -- Perform the update!
    UPDATE [Products]
        SET DiscountPercent = @NewDiscountPercent
        WHERE ProductID = @ProductID
GO

-- Initial State
SELECT ProductID, DiscountPercent FROM Products WHERE ProductID = 1;

-- Test Validation
BEGIN TRY
    EXEC [spUpdateProductDiscount]
        @ProductId = 1, @NewDiscountPercent = -50;
END TRY
BEGIN CATCH PRINT 'Error message: ' + ERROR_MESSAGE() END CATCH

-- Test Functionality
EXEC [spUpdateProductDiscount]
    @ProductId = 1, @NewDiscountPercent = 60;

-- End State
SELECT ProductID, DiscountPercent FROM Products WHERE ProductID = 1;
```

RESULTS

	ProductID	DiscountPercent
1	1	30.00

	ProductID	DiscountPercent
1	1	60.00

MESSAGES

Timestamp	Message
[11:36:07 PM]	Started executing query at Line 179 Commands completed successfully.
[11:36:07 PM]	Started executing query at Line 183 Commands completed successfully.
[11:36:07 PM]	Started executing query at Line 200 (1 row affected) Error message: New discount percent cannot be less than 0. Provided discount percent: -50.00 (1 row affected) (1 row affected) (0 rows affected) (0 rows affected) (1 row affected) Total execution time: 00:00:00.036

Section 6

```
USE MyGuitarShop;
DROP TRIGGER IF EXISTS [Products_UPDATE];

GO

CREATE TRIGGER [Products_UPDATE] ON [Products] FOR UPDATE
AS
    -- Acquire variables
    DECLARE @ProductIDChar CHAR = CAST((SELECT ProductId FROM [inserted]) AS CHAR);
    DECLARE @NewDiscountPercent MONEY = (SELECT DiscountPercent FROM [inserted]);

    -- Validate the new discount percentage
    IF @NewDiscountPercent < 0 OR @NewDiscountPercent > 100 BEGIN
        DECLARE @DiscountPercentInvalidErrorMessage VARCHAR(255) =
            'Discount percent must be between 0 and 100 (inclusive). ' +
            'Provided discount percent of ' + CAST(@NewDiscountPercent AS CHAR) +
            ' for product with ID ' + @ProductIDChar;
        THROW 51004, @DiscountPercentInvalidErrorMessage, 1;
    END

    -- Normalize the discount percentage
    IF @NewDiscountPercent <= 1 BEGIN
        DECLARE @NormalizedDiscountPercent MONEY = @NewDiscountPercent * 100
        PRINT
            'Normalizing discount scalar of ' + CAST(@NewDiscountPercent AS CHAR) +
            ' to percentage of ' + CAST(@NormalizedDiscountPercent AS CHAR) +
            ' for product with ID ' + @ProductIDChar

        UPDATE [Products] SET DiscountPercent = @NormalizedDiscountPercent
        FROM [Products] INNER JOIN [inserted]
        ON [Products].ProductID = [inserted].ProductID
    END
GO

BEGIN TRY
    UPDATE [Products] SET DiscountPercent = -22 WHERE ProductID = 1;
END TRY BEGIN CATCH PRINT 'Error message: ' + ERROR_MESSAGE() END CATCH

BEGIN TRY
    UPDATE [Products] SET DiscountPercent = 101 WHERE ProductID = 1;
END TRY BEGIN CATCH PRINT 'Error message: ' + ERROR_MESSAGE() END CATCH

UPDATE [Products] SET DiscountPercent = 0.22 WHERE ProductID = 1;

SELECT ProductID, DiscountPercent FROM Products WHERE ProductID = 1;
```

RESULTS			CTRL+ALT+
	ProductID	DiscountPercent	
1	1	22.00	
MESSAGES			CTRL+ALT+
Timestamp	Message		
[5:21:04 PM]	Started executing query at Line 224 Commands completed successfully.		
[5:21:04 PM]	Started executing query at Line 228 Commands completed successfully.		
[5:21:04 PM]	Started executing query at Line 256 (0 rows affected) Error message: Discount percent must be between 0 and 100 (inclusive). Provided discount percent of -22.00 for product with ID 1 (0 rows affected) Error message: Discount percent must be between 0 and 100 (inclusive). Provided discount percent of 101.00 for product with ID 1 Normalizing discount scalar of 0.22 to percentage of 22.00 for product with ID 1 (1 row affected) (1 row affected) (1 row affected) Total execution time: 00:00:00.060		

Section 7

```
USE MyGuitarShop;
DROP TRIGGER IF EXISTS [Products_INSERT];

GO

CREATE TRIGGER [Products_INSERT] ON [Products] AFTER INSERT AS
    DECLARE @ProductIDChar CHAR = CAST((SELECT ProductId FROM [inserted]) AS CHAR);
    DECLARE @DateAdded DATETIME = (SELECT DateAdded FROM [inserted]);

    IF (@DateAdded) IS NULL BEGIN
        PRINT
            'Null DateAdded detected inserting product with id ' + @ProductIDChar + '. ' +
            'DateAdded value will be set to ' + CAST(GETDATE() AS CHAR)
        UPDATE [Products] SET DateAdded = GETDATE()
        FROM [Products] JOIN [inserted]
        ON [Products].ProductID = [inserted].ProductID;
    END
GO

INSERT INTO [Products] (
    ProductCode, ProductName, ListPrice, DiscountPercent, CategoryID, Description
) VALUES (
    'test_003', 'unimaginative test name', 100, 10, 1, ''
);

SELECT ProductID, ProductCode, DateAdded FROM Products WHERE ProductCode = 'test_003';
```

RESULTS

	ProductID	Catego...	ProductCode	ProductName	Des...	ListPrice	Discou...	DateAdded
1	1035	1	test_003	unimaginative test na...		100.00	10.00	2024-05-26 01:35:26.370

MESSAGES

Timestamp	Message
[6:35:26 PM]	Started executing query at Line 273 Commands completed successfully.
[6:35:26 PM]	Started executing query at Line 277 Commands completed successfully.
[6:35:26 PM]	Started executing query at Line 290 Null DateAdded detected inserting product with id 1. DateAdded value will be set to May 26 2024 1:35AM (1 row affected) (1 row affected) (1 row affected) Total execution time: 00:00:00.029

Section 8

```
USE MyGuitarShop;

DROP TABLE IF EXISTS [ProductsAudit];
CREATE TABLE [ProductsAudit](
    AuditID INT IDENTITY(1,1) NOT NULL,
    ProductID INT NOT NULL,
    CategoryID INT NULL,
    ProductCode VARCHAR(10) NOT NULL,
    ProductName VARCHAR(255) NOT NULL,
    ListPrice MONEY NOT NULL,
    DiscountPercent MONEY NOT NULL,
    DateUpdated DATETIME NULL,
);

DROP TRIGGER IF EXISTS [Products_UPDATE_ProductAudit];

GO
CREATE TRIGGER [Products_UPDATE_ProductAudit] ON [Products] INSTEAD OF UPDATE AS
    DECLARE @ProductID CHAR = (SELECT ProductId FROM [inserted]);

    -- Secretly add the current values into the audit table 🕵️
    INSERT INTO [ProductsAudit] (
        ProductID, CategoryID, ProductCode, ProductName, ListPrice, DiscountPercent, DateUpdated
    ) SELECT
        ProductID, CategoryID, ProductCode, ProductName, ListPrice, DiscountPercent, GETDATE()
        FROM [Products] WHERE [Products].ProductID = @ProductID

    -- Update the actual product table
    UPDATE [Products] SET
        CategoryID = [inserted].CategoryID,
        ProductCode = [inserted].ProductCode,
        ProductName = [inserted].ProductName,
        Description = [inserted].Description,
        ListPrice = [inserted].ListPrice,
        DateAdded = [inserted].DateAdded,
        DiscountPercent = [inserted].DiscountPercent
    FROM [Products] JOIN [inserted]
    ON [Products].ProductID = [inserted].ProductID

GO

UPDATE [Products] SET DiscountPercent = 40 WHERE ProductID = 1;
UPDATE [Products] SET DiscountPercent = 60 WHERE ProductID = 1;

SELECT * FROM [ProductsAudit];
SELECT * FROM [Products] WHERE ProductID = 1;
```


RESULTS

	AuditID	ProductID	Cate...	ProductC...	ProductN...	ListPrice	DiscountPercent	DateUpdated
1	1	1	1	strat	Fender St...	699.00	30.00	2024-05-26 01:50:01.113
2	2	1	1	strat	Fender St...	699.00	40.00	2024-05-26 01:50:01.117

	ProductID	Cate...	ProductC...	ProductN...	D...	ListPrice	DiscountPercent	DateAdded
1	1	1	strat	Fender Str...	T..	699.00	60.00	2019-03-30 09:32:40.000

MESSAGES

Timestamp	Message
[6:50:01 PM]	Started executing query at Line 307 Commands completed successfully.
[6:50:01 PM]	Started executing query at Line 324 Commands completed successfully.
[6:50:01 PM]	Started executing query at Line 346 (1 row affected) (1 row affected) (1 row affected) (1 row affected) (1 row affected) (1 row affected) (2 rows affected) (1 row affected) Total execution time: 00:00:00.033