Patrick Warren

Assignment: A9

Due: 11/15/2017

CIS310-01

-- Patrick Warren

-- A9

-- CIS310-01

-- Due 11/15/2017

--MAKE SURE THAT INVOICE TOTAL VALUE IS CORRECT BEFORE RUNNING TRIGGER.

--THIS RESETS INV\_TOTAL BACK TO THE VALUES ADDED IN LGLINE FOR THE TOTAL

UPDATE LGINVOICE

SET INV\_TOTAL = (SELECT SUM(LINE\_QTY\*LINE\_PRICE) FROM LGLINE)

DECLARE @INV\_NUM CHAR(4)

DECLARE @INV\_TOTAL MONEY

IF(EXISTS (SELECT \* FROM LGLINE))

BEGIN

DECLARE INSERTED\_CURSOR CURSOR FOR

SELECT LI.INV\_NUM, SUM(LL.LINE\_QTY\*LL.LINE\_PRICE) AS INV\_TOTAL

FROM LGLINE LL INNER JOIN LGINVOICE LI ON LI.INV\_NUM = LL.INV\_NUM

GROUP BY LI.INV\_NUM

OPEN INSERTED\_CURSOR

FETCH NEXT FROM INSERTED\_CURSOR

INTO @INV\_NUM, @INV\_TOTAL

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE LGINVOICE

SET INV\_TOTAL = @INV\_TOTAL

WHERE INV\_NUM = @INV\_NUM

FETCH NEXT FROM INSERTED\_CURSOR

INTO @INV\_NUM, @INV\_TOTAL

END

CLOSE INSERTED\_CURSOR

DEALLOCATE INSERTED\_CURSOR

END

--START OF TRIGGER

ALTER TRIGGER A9

ON LGLINE

AFTER INSERT, DELETE, UPDATE

AS

BEGIN

DECLARE @INV\_NUM CHAR(4)

DECLARE @TOTAL MONEY

--LOGIC TO HANDLES RECORDS IN INSERTED

IF(EXISTS (SELECT \* FROM INSERTED) AND NOT EXISTS(SELECT \* FROM DELETED))

BEGIN

DECLARE INSERTED\_CURSOR CURSOR FOR

SELECT INV\_NUM, SUM(LINE\_QTY\*LINE\_PRICE) AS TOTAL

FROM INSERTED

GROUP BY INV\_NUM, LINE\_QTY

OPEN INSERTED\_CURSOR

FETCH NEXT FROM INSERTED\_CURSOR

INTO @INV\_NUM, @TOTAL

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE LGINVOICE

SET INV\_TOTAL = INV\_TOTAL + @TOTAL

WHERE INV\_NUM = @INV\_NUM

FETCH NEXT FROM INSERTED\_CURSOR

INTO @INV\_NUM, @TOTAL

END

CLOSE INSERTED\_CURSOR

DEALLOCATE INSERTED\_CURSOR

END

--LOGIC TO HANDLE RECORDS IN DELETED

IF(EXISTS (SELECT \* FROM DELETED) AND NOT EXISTS (SELECT \* FROM INSERTED))

BEGIN

DECLARE DELETED\_CURSOR CURSOR FOR

SELECT INV\_NUM, SUM(LINE\_QTY\*LINE\_PRICE) AS TOTAL

FROM DELETED

GROUP BY INV\_NUM, LINE\_QTY

OPEN DELETED\_CURSOR

FETCH NEXT FROM DELETED\_CURSOR

INTO @INV\_NUM, @TOTAL

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE LGINVOICE

SET INV\_TOTAL = INV\_TOTAL - @TOTAL

WHERE INV\_NUM = @INV\_NUM

FETCH NEXT FROM DELETED\_CURSOR

INTO @INV\_NUM, @TOTAL

END

CLOSE DELETED\_CURSOR

DEALLOCATE DELETED\_CURSOR

END

--LOGIC FOR UPDATED. LOOKS AT BOTH INSERTED AND DELETED TABLES FOR CHANGES

IF(EXISTS (SELECT \* FROM INSERTED) AND EXISTS(SELECT \* FROM DELETED))

BEGIN

DECLARE UPDATED\_CURSOR CURSOR FOR

SELECT I.INV\_NUM, SUM((D.LINE\_QTY\*D.LINE\_PRICE) - (I.LINE\_QTY \* I.LINE\_PRICE)) AS INV\_TOTAL

FROM INSERTED I INNER JOIN DELETED D ON I.INV\_NUM = D.INV\_NUM

GROUP BY I.INV\_NUM

OPEN UPDATED\_CURSOR

FETCH NEXT FROM UPDATED\_CURSOR

INTO @INV\_NUM, @TOTAL

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE LGINVOICE

SET INV\_TOTAL = INV\_TOTAL - @TOTAL

WHERE INV\_NUM = @INV\_NUM

FETCH NEXT FROM UPDATED\_CURSOR

INTO @INV\_NUM, @TOTAL

END

CLOSE UPDATED\_CURSOR

DEALLOCATE UPDATED\_CURSOR

END

END

GO

--TESTS!!!

--Main query to see if INV\_TOTAL is updated

SELECT \*

FROM LGINVOICE

ORDER BY INV\_NUM

--Show important line items from LGLINE. TOTAL should match LGINVOICE from above

SELECT INV\_NUM, LINE\_NUM, LINE\_QTY, LINE\_PRICE, sum(LINE\_QTY\*LINE\_PRICE) AS TOTAL

FROM LGLINE

GROUP BY INV\_NUM, LINE\_NUM, LINE\_QTY, LINE\_PRICE

ORDER BY INV\_NUM

--Adds two pieces where the price is $7.49 and invoice is 104

UPDATE LGLINE

SET LINE\_QTY = LINE\_QTY + 2

WHERE LINE\_PRICE = '7.49' AND

INV\_NUM = 104

--Removes two items where INV\_NUM = 106

UPDATE LGLINE

SET LINE\_QTY = LINE\_QTY - 1

WHERE INV\_NUM = 106

--Adds two items from all line items in invoice 105

UPDATE LGLINE

SET LINE\_QTY = LINE\_QTY + 2

WHERE INV\_NUM = 105

--Adds 10 '3274-XJX' to invoice 104.

INSERT INTO LGLINE

VALUES (104, 8, '3274-XJX', 10, 3.99)

--Removes 10 '3274-XJX' from invoice 104.

DELETE FROM LGLINE

WHERE INV\_NUM = 104 AND PROD\_SKU = '3274-XJX'

