LESLIE DAWSON

CIS 310-01

DATABASE DESIGN

ASSIGNMENT 8B

DUE 4/18/2011

USE CIS31007

-- =============================================

-- Author: Leslie Dawson

-- Create date: 4/13/2011

-- Description: A8B - ADDING CUS\_TOTAL\_DISTANCE COLUMN TO CUSTOMER TABLE

-- =============================================

ALTER TABLE CUSTOMER

ADD CUS\_TOTAL\_DISTANCE REAL

-- =============================================

-- Author: Leslie Dawson

-- Create date: 4/13/2011

-- Description: A8B - STORED PROCEDURE TO POPULATE CUS\_TOTAL\_DISTANCE

-- IN CUSTOMER TABLE

-- =============================================

CREATE PROCEDURE A8B\_UPDATE\_CUS\_TOTAL\_DISTANCE

AS

BEGIN

DECLARE CUS\_TOTAL\_DISTANCE CURSOR FOR

SELECT CUS\_CODE, SUM(CHAR\_DISTANCE) AS [CUS\_TOTAL\_DISTANCE]

FROM CHARTER

GROUP BY CUS\_CODE

DECLARE @CUS\_CODE INT

DECLARE @CUS\_TOTAL\_DISTANCE REAL

OPEN CUS\_TOTAL\_DISTANCE

FETCH NEXT FROM CUS\_TOTAL\_DISTANCE INTO @CUS\_CODE, @CUS\_TOTAL\_DISTANCE

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE CUSTOMER

SET CUS\_TOTAL\_DISTANCE = @CUS\_TOTAL\_DISTANCE

WHERE CUS\_CODE = @CUS\_CODE

FETCH NEXT FROM CUS\_TOTAL\_DISTANCE INTO @CUS\_CODE, @CUS\_TOTAL\_DISTANCE

END

CLOSE CUS\_TOTAL\_DISTANCE

DEALLOCATE CUS\_TOTAL\_DISTANCE

END

-- =============================================

-- Author: Leslie Dawson

-- Create date: 4/13/2011

-- Description: A8B - TRIGGER TO UPDATE CUS\_TOTAL\_DISTANCE IN CUSTOMER TABLE

-- AFTER CHANGE MADE TO CHARTER TABLE

-- =============================================

CREATE TRIGGER A8B\_TRIGGER\_TO\_UPDATE\_CUS\_TOTAL\_DISTANCE

ON CHARTER

AFTER INSERT,DELETE,UPDATE

AS

BEGIN

DECLARE @CUS\_CODE INT

DECLARE @DISTANCE\_CHANGE REAL

--FOR INSERT CASE (INSERTED TABLE NOT EMPTY)

IF(EXISTS (SELECT \* FROM INSERTED))

BEGIN

DECLARE INSERT\_CURSOR CURSOR FOR

SELECT I.CUS\_CODE, SUM(I.CHAR\_DISTANCE) AS [DISTANCE\_CHANGE]

FROM INSERTED I

GROUP BY I.CUS\_CODE

OPEN INSERT\_CURSOR

FETCH NEXT FROM INSERT\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE CUSTOMER

SET CUS\_TOTAL\_DISTANCE = CUS\_TOTAL\_DISTANCE + @DISTANCE\_CHANGE

WHERE CUS\_CODE = @CUS\_CODE

FETCH NEXT FROM INSERT\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

END

CLOSE INSERT\_CURSOR

DEALLOCATE INSERT\_CURSOR

END

--FOR DELETE CASE (DELETED TABLE NOT EMPTY)

IF(EXISTS (SELECT \* FROM DELETED))

BEGIN

DECLARE DELETE\_CURSOR CURSOR FOR

SELECT D.CUS\_CODE, SUM(D.CHAR\_DISTANCE) AS [DISTANCE\_CHANGE]

FROM DELETED D

GROUP BY D.CUS\_CODE

OPEN DELETE\_CURSOR

FETCH NEXT FROM DELETE\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE CUSTOMER

SET CUS\_TOTAL\_DISTANCE = CUS\_TOTAL\_DISTANCE - @DISTANCE\_CHANGE

WHERE CUS\_CODE = @CUS\_CODE

FETCH NEXT FROM DELETE\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

END

CLOSE DELETE\_CURSOR

DEALLOCATE DELETE\_CURSOR

END

--FOR UPDATE CASE (INSERTED AND DELETED TABLES NOT EMPTY)

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

BEGIN

DECLARE UPDATE\_CURSOR CURSOR FOR

SELECT I.CUS\_CODE, SUM(I.CHAR\_DISTANCE - D.CHAR\_DISTANCE) AS [DISTANCE\_CHANGE]

FROM INSERTED I INNER JOIN DELETED D ON I.CUS\_CODE = D.CUS\_CODE

GROUP BY I.CUS\_CODE

OPEN UPDATE\_CURSOR

FETCH NEXT FROM UPDATE\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

WHILE(@@FETCH\_STATUS = 0)

BEGIN

UPDATE CUSTOMER

SET CUS\_TOTAL\_DISTANCE = CUS\_TOTAL\_DISTANCE + @DISTANCE\_CHANGE

WHERE CUS\_CODE = @CUS\_CODE

FETCH NEXT FROM UPDATE\_CURSOR INTO @CUS\_CODE, @DISTANCE\_CHANGE

END

CLOSE UPDATE\_CURSOR

DEALLOCATE UPDATE\_CURSOR

END

END