

```

-- Assignment 8
-- Due 4/11/18
-- Team: Kaylee Thomas, Madeline Cowgill, Aziz Arrak, Shivani Patel

-- Test Code below (before TRIGGER A8)

-- See what is originally in the DETAILRENTAL table BEFORE the Trigger
SELECT      *
FROM        DETAILRENTAL

-- Find data for testing (RENT_NUM: 1006, 1007)
SELECT      *
FROM        DETAILRENTAL
WHERE       RENT_NUM IN (1006, 1007)

-- reset the DETAIL_DAILYLATEFEE to $5 for testing purposes
UPDATE      DETAILRENTAL
SET DETAIL_DAILYLATEFEE = 5
WHERE       RENT_NUM IN (1006, 1007)

-- reset the due date for testing purposes
UPDATE      DETAILRENTAL
SET DETAIL_DUEDATE = '3-8-2013'
WHERE       RENT_NUM IN (1006, 1007)

-- reset the return date for testing purposes
UPDATE      DETAILRENTAL
SET DETAIL_RETURNDATE = '3-8-2013'
WHERE       RENT_NUM IN (1006, 1007)

-- Show what customers are affected by RENT_NUM 1006 and 1007 and show how many videos they currently have out
under that RENT_NUM
SELECT      R.RENT_NUM, M.MEM_NUM, M.MEM_FNAME, M.MEM_LNAME, M.MEM_BALANCE,
COUNT(M.MEM_NUM) AS NUMOFMOVIES
FROM        MEMBERSHIP M INNER JOIN RENTAL R ON M.MEM_NUM = R.MEM_NUM
INNER JOIN DETAILRENTAL D ON R.RENT_NUM = D.RENT_NUM
WHERE       R.RENT_NUM IN (1006, 1007)
GROUP BY   R.RENT_NUM, M.MEM_NUM, M.MEM_FNAME, M.MEM_LNAME, M.MEM_BALANCE

-- reset balances for MEM_NUM 104 and 107 to 0 for testing purposes
UPDATE      MEMBERSHIP
SET MEM_BALANCE = 0
WHERE       MEM_NUM IN (104, 107)

--TRIGGER STATEMENT
-- PURPOSE: we want to make changes (insert, delete, update) in DETAILRENTAL so it automatically updates
MEM_BALANCE value of the affected items

--75. Create a trigger named trg_mem_balance that will maintain the correct value in the membership balance in
the MEMBERSHIP table when videos are returned late.
--The trigger should execute as an AFTER trigger when the due date or return date attributes are updated in
the DETAILRENTAL table.

```

```

Create      TRIGGER A8
      ON DETAILRENTAL
      AFTER UPDATE
AS
BEGIN

DECLARE @RENT_NUM CHAR (4)
DECLARE @PRIOR_DAYS_LATE INT
DECLARE @NEW_DAYS_LATE INT
DECLARE @DAILY_LATE_FEE DECIMAL (5,2)
DECLARE @PRIOR_LATE_FEE DECIMAL (5,2)
DECLARE @NEW_LATE_FEE DECIMAL (5,2)
DECLARE @UPDATED_LATE_FEE DECIMAL (5,2)

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

    DECLARE          UPDATE_CURSOR CURSOR FOR
    SELECT I.RENT_NUM, DATEDIFF(DAY, D.DETAIL_DUEDATE, D.DETAIL_RETURNDATE) AS PRIOR_DAYS_LATE,
           DATEDIFF(DAY, I.DETAIL_DUEDATE, I.DETAIL_RETURNDATE) AS NEW_DAYS_LATE,
           I.DETAIL_DAILYLATEFEE
    FROM    INSERTED I INNER JOIN DELETED D ON I.RENT_NUM = D.RENT_NUM
           AND I.VID_NUM = D.VID_NUM

    OPEN    UPDATE_CURSOR
    FETCH   NEXT FROM UPDATE_CURSOR INTO @RENT_NUM, @PRIOR_DAYS_LATE, @NEW_DAYS_LATE, @DAILY_LATE_FEE
    WHILE   (@@FETCH_STATUS = 0)

--The trigger should satisfy the following conditions:
-- A. Calculate the value of the late fee prior to the update that triggered this execution of the trigger.
--The value of the late fee is the days late multiplied by the daily late fee.
--If the previous value of the late fee was null, then treat it as zero (0).

-- you can only use named variable here in IF statements
    BEGIN
        IF @PRIOR_DAYS_LATE IS NULL
            BEGIN
                SET @PRIOR_LATE_FEE = 0
            END
        ELSE
            BEGIN
                SET @PRIOR_LATE_FEE = (@PRIOR_DAYS_LATE * @DAILY_LATE_FEE)
            END
    END

--B. Calculate the value of the late fee after the update that triggered this execution of the trigger.
--If the value of the late fee is now null, then treat it as zero (0).

    IF @NEW_DAYS_LATE IS NULL
        BEGIN
            SET @NEW_LATE_FEE = 0
        END
    ELSE
        BEGIN
            SET @NEW_LATE_FEE = (@NEW_DAYS_LATE * @DAILY_LATE_FEE)
        END
    END

-- C. Subtract the prior value of the late fee from the current value of the late fee to determine the change
in late fee for this video rental.
-- &

```

-- D. If the amount calculated in Part c is not zero (0), then update the membership balance by the amount calculated for the membership associated with this rental.

```
        IF ((@NEW_LATE_FEE - @PRIOR_LATE_FEE) <> 0)
            BEGIN
                SET @UPDATED_LATE_FEE = (@NEW_LATE_FEE - @PRIOR_LATE_FEE)
            END
        ELSE
            BEGIN
                SET @UPDATED_LATE_FEE = 0
            END
        BEGIN
            UPDATE MEMBERSHIP
            SET     MEM_BALANCE = MEM_BALANCE + @UPDATED_LATE_FEE
            WHERE  MEM_NUM = (SELECT MEM_NUM
                                FROM RENTAL
                                WHERE RENT_NUM = @RENT_NUM
                            )
        END
        FETCH NEXT FROM UPDATE_CURSOR INTO @RENT_NUM, @PRIOR_DAYS_LATE, @NEW_DAYS_LATE,
        @DAILY_LATE_FEE
    END
    CLOSE UPDATE_CURSOR
    DEALLOCATE UPDATE_CURSOR
END
```

END

-- Test Code below (after TRIGGER A8)

```
SELECT *
FROM   DETAILRENTAL
WHERE  RENT_NUM IN (1006, 1007)
```

```
SELECT *
FROM   MEMBERSHIP
WHERE  MEM_NUM IN (104, 107)
```

-- Statements that will fire the TRIGGER statement

```
UPDATE DETAILRENTAL
SET DETAIL_RETURNDATE = '3-10-2013'
WHERE RENT_NUM IN (1006, 1007)
```

-- Reverse the statement above to make the table have original values

```
UPDATE DETAILRENTAL
SET DETAIL_RETURNDATE = '3-8-2013'
WHERE RENT_NUM IN (1006, 1007)
```