**10.1 - MERGE STATEMENT, RANK FUNCTIONS**

CREATE TABLE tblOrders (

ID INT NOT NULL PRIMARY KEY, TitleTag NVARCHAR(30) NOT NULL, Measure INT NOT NULL DEFAULT 0 )

CREATE TABLE tblInventory (

ID INT NOT NULL PRIMARY KEY, TitleTag NVARCHAR(30) NOT NULL, Measure INT NOT NULL DEFAULT 0 )

INSERT tblOrders VALUES (1, 'Comedy', 100), (2, 'NEGOT', 100), (3, 'BUSINESS', 100)

INSERT tblInventory VALUES (3, 'BUSN PROFILES', 200), (4, 'PB CHAPTER BKS', 200),

(5, 'BEG READER HC', 200), (6, 'Literature', 200)

SELECT \* FROM tblOrders; SELECT \* FROM tblInventory

**MERGE** : A MECHANISM TO COMPARE TABLES. IDENTIFY MATCHING DATA AND MISSING DATA. PERFORM DML OPERATIONS ON THIS MATCHING & / MISSING DATA.

**REQUIREMENT**: COMPARE ABOVE TABLES BASED ON ID COLUMN.

FOR MATCHING IDs : ADD MEASURE COLUMN VALUES, STORE UPDATED VALUE IN INVENTORY TABLE

FOR UNMATCHED IDs : INSERT SUCH MISSING ROWS FROM ORDERS TABLE INTO INVENTORY TABLE

**-- SOLUTION:**

MERGE tblInventory -- this is the target table

USING tblOrders -- this is the source table

ON tblInventory.ID = tblOrders.ID

WHEN MATCHED THEN UPDATE SET tblInventory.Measure = tblInventory.Measure + tblOrders.Measure

WHEN NOT MATCHED THEN INSERT VALUES (tblOrders.ID, tblOrders.TitleTag, tblOrders.Measure);

SELECT \*, **IIF**(MEASURE = 100, 'row from orders', 'ROW from inventory') AS RSTATUS FROM tblInventory

SELECT \*, **CASE** WHEN MEASURE = 100 THEN 'ROW FROM ORDERS'

WHEN MEASURE = 200 THEN 'ROW FROM INVENTORY'

WHEN MEASURE = 300 THEN 'ROW FROM ORDERS & INVENTORY'

ELSE 'NO GRADE AVAILABLE'

END AS ROW\_STATUS FROM tblInventory

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**-- RANK FUNCTIONS:**

SELECT \*, ROW\_NUMBER() OVER (ORDER BY COUNTRY) FROM ServiceSales

SELECT \*, RANK() OVER (ORDER BY COUNTRY) FROM ServiceSales

SELECT \*, DENSE\_RANK() OVER (ORDER BY COUNTRY) FROM ServiceSales

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY COUNTRY ORDER BY COUNTRY) FROM ServiceSales

-- **ROW\_NUMBER** : A UNIQUE NUMBER FOR EVERY ROW

-- **RANK** : SAME SEQUENCE FOR REPEATED VALUE.

NEW SEQUENCE DEPENDS ON ROW NUMBER

-- **DENSE\_RANK** : SAME SEQUENCE FOR REPEATED VALUE.

NEW SEQUENCE DEPENDS ON PREVIOUS SEQUENCE

-- **PARTITION BY** : A UNIQUE SEQUENCE FOR REPEATED VALUES.

SEQUENCE REPEATS FOR EVERY REPEATED VALUE