/\*

What are Ranking functions?

Row\_number: This function simply assigns sequential numbers to each row

RANK(): This function will return the rank (first, second, third, etc..) of each row in the result set

DENSE\_RANK(): This is same as RANK() function. Only difference is returns rank without gaps

The purpose of each is to either rank records in a result-set or rank records within groups that are partitioned

The PARTITION BY clause groups the data

If the Rank function views two or more records that have the same value, then these records get the same rank

\*/

Use TSQL

Go

--Create a table

Create Table SportsProduct

(ProductCategory varchar (30),

Product varchar (30),

Price Money,

DateSold date)

--Insert records

Insert Into SportsProduct

Values

('Ball','Tennisball', 4.95, '2015-08-02'),

('Rackets','Wilson', 49.00, '2015-02-02'),

('Shirts','Nike', 49.95, '2014-08-05'),

('Ball','Basketball', 64.95, '2015-04-07'),

('Rackets','Babolot', 19.95, '2015-08-02'),

('Shirts','Addidas', 54.95, '2012-10-02'),

('Ball','Football', 14.95, '2015-05-02'),

('Rackets','Prince', 95.95, '2012-08-02'),

('Shirts','Polo', 94.95, '2013-08-02'),

('Ball','Soccorball', 4.95, '2015-08-02')

--View records

Select \* From SportsProduct

--Order By can sort multiple columns, but cannot rank them, so use Rank function.

Select \* From SportsProduct

Order By ProductCategory,Product,Price Desc

--When you need to sort the data by grouping or finding maximum cost WITHIN A GROUP and rank them, use these 3 Rank FUNCTIONS.

--First thing to do is add a Row Number for sorting using the Row\_number() function

--ROW\_NUMBER() This function returns a sequential number and sorts the data by the Order By clause

Select

ROW\_NUMBER() Over (Order By Product desc) AS TempRowNumber, ProductCategory, Product, Price --<< this (Order By is mandatory)

from SportsProduct

--RANK() This function will returns the rank (first, second, third, etc..) of each row in the result set of partitioned (grouped) column

--Note, when ranking encounters a value with the same amount, it ranks them the same (Soccorball and Tennisball)

Select

ROW\_NUMBER() Over(Order By ProductCategory desc) RowNumber,ProductCategory, Product, Price,

RANK() Over(Partition by ProductCategory Order By Price Desc) AS Rankorder --<<will rank the partitioned column (ProductCategory) ordered by column (Price) in descending order

from SportsProduct

--DENSE\_RANK() This is same as RANK() function. Only difference is returns rank without gaps.(there is no gap in the sequence of numbers)

Select

ROW\_NUMBER() Over(Order By Price) RowNumber,ProductCategory, Product, Price,

RANK() over(order by Price ) as RankNumber

From SportsProduct

Select

ROW\_NUMBER() Over(Order By Price) RowNumber,ProductCategory, Product, Price

,RANK() over(order by Price ) as RankNumber,

DENSE\_RANK() over(order by Price) AS DenseRankorder

From SportsProduct

--DROP TABLE SportsProduct

--TRUNCATE TABLE SportsProduct