**Lesson 4**

**--Exercise 3**

*При каких значениях оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат?*

Поскольку ранжирующие функции строятся на основе определенного столбца, то в данном столбце должны быть уникальные значения (не повторяющиеся), тогда оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат.

**--Exercise 4**

/\*

a) Изучите данные в таблице Production.UnitMeasure. Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. Сколько всего различных кодов здесь есть?

Вставьте следующий набор данных в таблицу:

 TT1, Test 1, 9 сентября 2020

 TT2, Test 2, getdate()

Проверьте теперь, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’.

\*/

SELECT [UnitMeasureCode]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'



SELECT COUNT([UnitMeasureCode]) AS count\_code

FROM [Production].[UnitMeasure]



INSERT INTO [Production].[UnitMeasure] ([UnitMeasureCode], [Name], [ModifiedDate])

VALUES ('TT1', 'Test 1', '2020-09-09'),

('TT2', 'Test 2', GETDATE())



SELECT \*

FROM [Production].[UnitMeasure]



SELECT [UnitMeasureCode]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'



/\*

b) Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’. Посмотрите результат в отсортированном виде по коду.

\*/

SELECT [UnitMeasureCode], [Name], [ModifiedDate]

INTO Production.UnitMeasureTest

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] IN ('TT1', 'TT2')



INSERT INTO Production.UnitMeasureTest ([UnitMeasureCode], [Name], [ModifiedDate])

SELECT [UnitMeasureCode], [Name], [ModifiedDate]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] = 'CAN'



SELECT \*

FROM Production.UnitMeasureTest

ORDER BY [UnitMeasureCode]



/\*

c) Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’.

\*/

UPDATE Production.UnitMeasureTest

SET [UnitMeasureCode] = 'TTT'



SELECT \*

FROM Production.UnitMeasureTest



/\*

d) Удалите все строки из Production.UnitMeasureTest.

\*/

DELETE

FROM Production.UnitMeasureTest



SELECT \*

FROM Production.UnitMeasureTest



/\*

e) Найдите информацию из Sales.SalesOrderDetail по заказам 43659,43664. С помощью оконных функций MAX, MIN, AVG найдем агрегаты по LineTotal для каждого SalesOrderID.

\*/

SELECT [SalesOrderID]

, MAX([LineTotal]) OVER(PARTITION BY [SalesOrderID]) AS max

, MIN([LineTotal]) OVER(PARTITION BY [SalesOrderID]) AS min

, AVG([LineTotal]) OVER(PARTITION BY [SalesOrderID]) AS avg

FROM [Sales].[SalesOrderDetail]

WHERE [SalesOrderID] IN ('43659', '43664')



/\*

f) Изучите данные в объекте Sales.vSalesPerson. Создайте рейтинг cреди продавцов на основе годовых продаж SalesYTD, используя ранжирующую оконную функцию.

Добавьте поле Login, состоящий из 3 первых букв фамилии в верхнем регистре + ‘login’ + TerritoryGroup (Null заменить на пустое значение).

Кто возглавляет рейтинг? А кто возглавлял рейтинг в прошлом году (SalesLastYear).

\*/

SELECT s.[BusinessEntityID]

,p.FirstName

,p.LastName

,s.[SalesYTD]

,DENSE\_RANK() OVER (ORDER BY s.[SalesYTD] DESC) AS Rating\_now

,CONCAT(UPPER(LEFT(LastName,3)), 'login', [Group]) AS Login

FROM [Sales].[SalesPerson] s

JOIN [Person].[Person] p ON s.BusinessEntityID = p.BusinessEntityID

LEFT JOIN [Sales].[SalesTerritory] st ON s.TerritoryID = st.TerritoryID



SELECT s.[BusinessEntityID]

,p.FirstName

,p.LastName

,s.[SalesLastYear]

,DENSE\_RANK() OVER (ORDER BY s.[SalesLastYear] DESC) AS Rating\_last\_year

FROM [Sales].[SalesPerson] s

JOIN [Person].[Person] p ON s.BusinessEntityID = p.BusinessEntityID



*В этом году рейтинг возглавляет Linda Mitchell, в прошлом - Ranjit Varkley Chudukatil.*

/\*

g) Найдите первый будний день месяца (FROM не используем). Нужен стандартный код на все времена.

\*/

SELECT DATEADD(day, -DAY(GETDATE())+2, GETDATE())



**--Exercise 5**

Найдите значения count(1), count(name), count(id), count(\*) для следующей таблицы:

Id(PK) Name DepName

1 null A

2 null null

3 A C

4 B C

***ОТВЕТ:***

COUNT(1) = 4

COUNT(name) = 2

COUNT(id) = 4

COUNT(\*) = 4