Hello! ☺☺☺

**Task\_3**

|  |
| --- |
| При каких значениях оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат? |
| Одинаковый результат будет, если все значения сортируемого поля будут уникальными |

**Task\_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(DISTINCT UnitMeasureCode) AS Number\_of\_Codes  FROM Production.UnitMeasure    INSERT INTO Production.UnitMeasure (UnitMeasureCode, Name, ModifiedDate)  VALUES ('TT1', 'Test1', CAST('2020-09-09' as datetime)),  INSERT INTO Production.UnitMeasure (UnitMeasureCode, Name, ModifiedDate)  VALUES ('TT2', 'Test2', GETDATE())    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 LIKE 'CAN' OR Name LIKE 'Test1' OR Name LIKE 'Test2'  ORDER BY UnitMeasureCode |

|  |
| --- |
| c) Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’. |
| UPDATE Production.UnitMeasureTest  SET UnitMeasureCode = 'TTT' |

|  |
| --- |
| d) Удалите все строки из Production.UnitMeasureTest. |
| DELETE  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 \*  FROM [Sales].[SalesOrderDetail]  WHERE SalesOrderID IN (43659, 43664)    SELECT SalesOrderID, LineTotal, MAX(LineTotal)  OVER (PARTITION BY SalesOrderID) as max\_LineTotal,  MIN(LineTotal)  OVER (PARTITION BY SalesOrderID) as min\_LineTotal,  AVG(LineTotal)  OVER (PARTITION BY SalesOrderID) as avg\_LineTotal  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 BusinessEntityID, SalesYTD,  RANK() OVER (ORDER BY SalesYTD DESC) as Rank  FROM [Sales].[SalesPerson]    SELECT BusinessEntityID, SalesYTD,  CONCAT(UPPER(LEFT(FirstMame, 3)), 'login', ISNULL(TerritoryGroup, '')) as login,  RANK() OVER (ORDER BY SalesLastYear DESC) as Rank  FROM [Sales].[SalesPerson] |

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

**Task\_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 |
| count(1) = 4  count(name) = 2  count(id) = 4  count(\*) = 4 |