**Определения 10.**

|  |  |
| --- | --- |
| **Пример 1.** Использование подзапроса в WHERE:  Номенклатура(**Товар\_ID**, Наименование, Масса)  SELECT Товар\_ID, Наименование, Масса  FROM Номенклатура  WHERE Масса >  (SELECT Масса FROM Номенклатура WHERE Товар\_ID =@Товар) | **Пример 2**. Подумайте, будет ли такой запрос работать?  SELECT Товар\_ID, Наименование, Масса  FROM Номенклатура  WHERE Товар\_ID = (SELECT Товар\_ID FROM Номенклатура WHERE Масса >@Масса) |

**Структура данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Документы** |  | **Документы\_данные** |  | **Остатки** |
| **ндок** |  | **ндок** |  | **Дата** |
| Дата |  | **Товар\_ID** |  | **Товар\_ID** |
| Покупатель |  | Колво |  | Остаток |
| Сумма |  | Цена |  |  |

|  |  |
| --- | --- |
| **Пример 3:** Использование distinct  SELECT DISTINCT Документы\_данные.Товар\_ID  FROM Документы  INNER JOIN Документы\_данные ON  Документы.ндок = Документы\_данные.ндок  WHERE CAST([Документы].[Дата] as date)=@День  AND Документы\_данные.Цена <  (  SELECT DISTINCT Цена  FROM Документы INNER JOIN Документы\_данные ON  Документы.ндок = Документы\_данные.ндок  WHERE Товар\_ID = @Товар AND CAST(Документы.Дата as date) = @День  ) | **Пример 4:** Использование агрегатных функций:  SELECT Покупатель, Сумма, ндок  FROM Документы as Продажи  WHERE Сумма =  (  SELECT MAX(Сумма)  FROM Документы  )  **Пример 5**: Использование в HAVING:  SELECT Масса, COUNT (Товар\_ID)  FROM Номенклатура  GROUP BY Масса  HAVING Масса > (SELECT Avg(Масса) FROM Номенклатура) |

**Задача:** Для **каждого** покупателя вывести документ, по которому покупатель потратил максимальное количество денег. Будут ли работать следующие запросы и почему?

|  |  |
| --- | --- |
| SELECT Покупатель, Сумма, ндок  FROM Документы as Продажи  WHERE Сумма =  (  SELECT MAX(Сумма)  FROM Документы  GROUP BY Покупатель  ) | SELECT Покупатель, Сумма, ндок  FROM Документы as Продажи  WHERE Сумма IN  (  SELECT MAX(Сумма)  FROM Документы  GROUP BY Покупатель  ) |

Возможные варианты замены предыдущих запросов:

|  |  |
| --- | --- |
| SELECT Покупатель, Сумма, ндок  FROM Документы as Продажи  WHERE Сумма =  (  SELECT MAX(Сумма)  FROM Документы  WHERE Документы.Покупатель = Продажи.Покупатель  ) | SELECT Продажи.Покупатель, Продажи.Сумма, ндок  FROM Документы as Продажи INNER JOIN  (  SELECT MAX(Сумма) as MaxSumm, Покупатель as Пок  FROM Документы  GROUP BY Покупатель  ) as MaxT ON  Продажи.Покупатель = MaxT.Пок AND  Продажи.Сумма = MaxT.MaxSumm |

**Задача:** на каждый факт продажи проставить остаток товара

SELECT Док.ндок, Док.Дата, Док.Покупатель, Док.Сумма, Днн.Товар\_ID, Днн.Колво, Днн.Цена,

(SELECT TOP 1 Остаток FROM Остатки WHERE Остатки.Товар\_ID = Днн.Товар\_ID AND Остатки.Дата <= Док.Дата ORDER BY Остатки.Дата desc) Остаток

FROM Документы Док INNER JOIN Документы\_данные Днн ON

Док.ндок = Днн.ндок

**Задача**: как для каждого акта продажи товара вывести статистику, сколько товара купили за месяц до?

**Задача**: как вывести рейтинг покупателя, используя подзапрос в SELECT?

**Пример** (осторожно обращайтесь с названиями полей!!!):

|  |  |
| --- | --- |
| DECLARE @Покупатели TABLE (Пок\_ID int primary key, ФИО nvarchar(max))  INSERT INTO @Покупатели VALUES (1, 'test1'), (2, 'test2') | DECLARE @Документы TABLE (ндок int primary key, ПокID int)  INSERT INTO @Документы VALUES (1, 1) |

SELECT \*

FROM @Покупатели

WHERE Пок\_ID in (SELECT distinct Пок\_ID FROM @Документы)

**EXISTS:**  SELECT \* FROM Таблица WHERE EXISTS (подзапрос)

Пример использования **EXISTS**: (выбор товаров, которые не продавались после 01.11.17)

|  |  |  |  |
| --- | --- | --- | --- |
| SELECT Товар\_ID  FROM Номенклатура  WHERE not exists  (  SELECT \*  FROM Документы INNER JOIN Документы\_данные ON  Документы.ндок = Документы\_данные.ндок  WHERE Документы\_данные.Товар\_ID = Номенклатура.Товар\_ID  AND Документы.Дата > ‘20171101’  ) | | SELECT Товар\_ID  FROM Номенклатура  WHERE Товар\_ID not in  (  SELECT Товар\_ID  FROM Документы INNER JOIN Документы\_данные ON  Документы.ндок = Документы\_данные.ндок  WHERE Документы.Дата > ‘20171101’  )  **PS:** не забывайте, как работает IN в случае, если в подзапросе есть NULL значения.  Как написать запрос при помощи LEFT JOIN ? | |
| DECLARE @Empl TABLE (empl\_id int)  INSERT INTO @Empl VALUES (1), (2), (3)  DECLARE @Deps TABLE (empl\_id int, boss\_id int)  INSERT INTO @Deps VALUES  (1, 2), (2, 3), (3, NULL) | SELECT \*  FROM @Empl E  WHERE empl\_id not in  (SELECT boss\_ID FROM @deps) | | SELECT \*  FROM @Empl E  WHERE NOT Exists (  SELECT \*  FROM @deps  WHERE boss\_id = E.empl\_id  ) |

**Задача (поиск минимального пропущенного числа):**

|  |  |
| --- | --- |
| CREATE TABLE #T (num INT NOT NULL PRIMARY KEY CHECK(num > 0));  INSERT INTO #T VALUES (3), (4), (6), (7)  SELECT CASE  WHEN EXISTS  (SELECT \* FROM #T WHERE num = 1) THEN missing ELSE 1 END  FROM  (  SELECT MIN(A.num) + 1 as missing  FROM #T AS A  WHERE NOT EXISTS  (  SELECT \* FROM #T AS B  WHERE B.num = A.num + 1  )  ) T | SELECT CASE WHEN (SELECT MIN(num) FROM #T) = 1 THEN missing ELSE 1 END  FROM  (  SELECT MIN(A.num + 1) missing  FROM #T A LEFT JOIN #T B ON  A.num + 1 = B.num  WHERE B.num is NULL  ) T |
| DECLARE @T TABLE (num int)  INSERT INTO @T SELECT num FROM #T  INSERT INTO @T VALUES (0)  SELECT MIN(A.num + 1) missing  FROM @T A LEFT JOIN @T B ON  A.num + 1 = B.num  WHERE B.num is NULL  DROP TABLE #T |

**Примеры использования ANY (какой-то) и ALL (все)**

|  |  |  |
| --- | --- | --- |
| DECLARE @T1 TABLE (num int)  INSERT INTO @T1 VALUES (2), (4), (6)  DECLARE @T2 TABLE (num int)  INSERT INTO @T2 VALUES (1), (3), (5) | SELECT \*  FROM @T1  WHERE num > any (SELECT num FROM @T2) | SELECT \*  FROM @T1  WHERE num > all (SELECT num FROM @T2) |