**Определения 8.**

**Задача (повторение пройденного материала):**

У Вас есть таблица с числами от 1 до 9, как сгенерировать таблицу с числами от 1 до 999?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Запрос на SQL:**  DECLARE @T TABLE (val int)  INSERT INTO @T  VALUES (0), (1), (2), (3), (4), (5), (6), (7), (8), (9)  SELECT T1.val\*100 + T2.val\*10 + T3.val new  FROM @T T1, @T T2, @T T3  ORDER BY new asc | |  |  | | --- | --- | |  | Avg on 10 Trials, ms | | Client processing time | 216.8 | | Total execution time | 344.2 | | Wait time on server replies | 127.4 |   Таблицы для 100 000 (5 декартовых) |
| **Запрос на синтаксисе SQL, но без использования SQL (!!! ПЛОХОЙ ВАРИАНТ !!!)**  DECLARE @new TABLE (val int)  DECLARE @i int  SET @i = 0  SET NOCOUNT ON  WHILE @i < 1000  BEGIN  INSERT INTO @new VALUES (@i)  SET @i = @i + 1  END  SET NOCOUNT OFF  SELECT \*  FROM @new | |  |  | | --- | --- | |  | Avg on 10 Trials, ms | | Client processing time | 2116.7 | | Total execution time | 2147.4 | | Wait time on server replies | 30.7 |   Обратите внимание, что такой синтаксис в можно использовать только в том случае, если запрос нельзя написать средствами стандартного SQL.  В нашем курсе допустимо использование такого синтаксиса только в случае, если этот пункт не оценивается (например, Вы лично для себя можете заполнить таблицу, но решать пункт в контрольной таким способом запрещено!) |

**Задача (построение гистограммы):**

У Вас есть таблица: для каждой частицы скорость, с которой она пролетела мимо датчика. Необходимо при помощи SQL получить данные для построения гистограммы.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **N** | **V** | | 1 | 10 | | 2 | 12 | | 3 | 13 | | 4 | 12 | | 5 | 11 | | DECLARE @Data TABLE (N int, V float)  SELECT V, COUNT(\*) Cnt FROM @Data  GROUP BY V | |  |  | | --- | --- | | **V** | **Cnt** | | 10 | 2 | | 12 | 2 | | 13 | 1 | |
| |  |  | | --- | --- | | **N** | **V** | | 1 | 10 | | 2 | 12 | | 3 | 13 | | 4 | 12 | | 5 | 11 | | DECLARE @h float  SET @h = 2  SELECT FLOOR(V/@h)\*@h B, FLOOR(V/@h + 1)\*@h E, COUNT(\*)  FROM @Data  GROUP BY FLOOR(V/@h)\*@h, FLOOR(V/@h + 1)\*@h | |  |  |  | | --- | --- | --- | | **B** | **E** | **Cnt** | | 10 | 12 | 2 | | 12 | 14 | 3 | |

**Пример замены UNION на LEFT**

|  |  |
| --- | --- |
| SELECT Покупатель, SUM(Отпущено), SUM(Брак)  FROM  (  SELECT Покупатель, SUM(Колво) as Отпущено, 0 as Брак  FROM Продажи  GROUP BY Покупатель  UNION  SELECT Покупатель, 0, SUM(Колво)  FROM Брак  GROUP BY Покупатель  ) T  GROUP BY Покупатель | SELECT Покупатели.Покупатель,  isNULL(Отпущено, 0) Продажи, isNULL(Брак, 0) Брак  FROM Покупатели  LEFT JOIN  (  SELECT Покупатель, SUM(Колво) as Отпущено  FROM Продажи  GROUP BY Покупатель  ) as Продажи ON  Покупатели.Покупатель = Продажи.Покупатель  LEFT JOIN  (  SELECT Покупатель, SUM(Колво) as  Брак  FROM Брак  GROUP BY Покупатель  ) as Брак ON  Покупатели.Покупатель = Брак.Покупатель |

Полезные операторы: **INTERSECT и EXCEPT. Синтаксис.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DECLARE @A TABLE (id1 int, id2 int)  INSERT INTO @A VALUES (1, 2), (1, 2)  DECLARE @B TABLE (id1 int, id2 int)  INSERT INTO @B VALUES (1, 4), (1, 3), (2, 5) | | | DECLARE @A TABLE (id1 int, id2 int)  INSERT INTO @A VALUES (1, 2), (1, 3), (NULL, NULL)  DECLARE @B TABLE (id1 int, id2 int)  INSERT INTO @B VALUES (1, 4), (1, 3), (NULL, 1) | | |
| SELECT id1  FROM @A  INTERSECT  SELECT id1  FROM @B | SELECT id2  FROM @A  EXCEPT  SELECT id2  FROM @B | | SELECT id1  FROM @A  INTERSECT  SELECT id1  FROM @A | | SELECT id2  FROM @A  EXCEPT  SELECT id2  FROM @B |
| DECLARE @A TABLE (id1 int)  INSERT INTO @A VALUES (3)  DECLARE @B TABLE (id1 int)  INSERT INTO @B VALUES (3), (2)  DECLARE @C TABLE (id1 int)  INSERT INTO @C VALUES (3), (1) | | SELECT id1  FROM @A  EXCEPT  SELECT id1  FROM @B  EXCEPT  SELECT id1  FROM @C | | SELECT id1  FROM @B  EXCEPT  SELECT id1  FROM @A  INTERSECT  SELECT id1  FROM @C | |

**Задача**: как получить разницу двух таблиц (с одинаковым колвом полей) ?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DECLARE @A TABLE (id1 int, id2 int, id3 int)  INSERT INTO @A  VALUES  (1, 2, 3), (4, 5, 6),  (1, 2, 3), (NULL, 1, 2),  (NULL, NULL, NULL)  DECLARE @B TABLE (id1 int, id2 int, id3 int)  INSERT INTO @B  VALUES (4, 5, 6), (7, 8, 9), (NULL, 1, 2) | SELECT A.\*  FROM @A A LEFT JOIN @B B ON  A.id1 = B.id1 AND  A.id2 = B.id2 AND  A.id3 = B.id3  WHERE B.id1 is NULL   |  |  |  | | --- | --- | --- | | id1 | id2 | id3 | | 1 | 2 | 3 | | 1 | 2 | 3 | | NULL | 1 | 2 | | NULL | NULL | NULL | | SELECT \*  FROM @A  EXCEPT  SELECT \*  FROM @B   |  |  |  | | --- | --- | --- | | id1 | id2 | id3 | | NULL | NULL | NULL | | 1 | 2 | 3 | |

Что изменилось бы, если в таблице В тоже была бы строка с NULL значениями?

**Задача**: вывести все товары, которые покупатель с идентификатором = 1 не покупал после 01.11.16 (включительно).

|  |  |
| --- | --- |
| DECLARE @Товары TABLE (good\_id int primary key)  INSERT INTO @Товары  VALUES (1), (2), (3)  DECLARE @Продажи TABLE (doc\_id int, Data datetime, IDC int, good\_id int, Qty int, primary key (doc\_id, good\_id))  INSERT INTO @Продажи VALUES  (1, '20160901', 1, 2, 1),  (2, '20160901', 1, 3, 1),  (3, '20161101', 1, 3, 1),  (4, '20161101', 2, 1, 1) | SELECT Товары.good\_id  FROM @Товары Товары LEFT JOIN @Продажи Продажи ON  Товары.good\_id = Продажи.good\_id AND  Продажи.IDC = 1 AND  Продажи.Data >= '20161101'  WHERE Продажи.doc\_id is NULL |
| SELECT good\_id  FROM @Товары  EXCEPT  SELECT good\_id  FROM @Продажи  WHERE IDC = 1 AND Data >= '20161101' |
| SELECT Товары.good\_id  FROM @Товары Товары LEFT JOIN  (  SELECT good\_id  FROM @Продажи  WHERE IDC = 1 AND Data >= '20161101'  ) ТоварыПокупателя ON  Товары.good\_id = ТоварыПокупателя.good\_id  WHERE ТоварыПокупателя.good\_id is NULL |

Что будет, если в первом запросе условие из LEFT JOIN перенести в WHERE?