

Лекция 13
Генерация последовательности.
Рекурсивные СТЕ

1. Объединение 100 запросов, выводящих по одному числу

```
SELECT 1 AS num
```

```
UNION ALL
```

```
SELECT 2
```

```
...
```

```
UNION ALL
```

```
SELECT 100;
```

2. Способ использования декартова произведения двух таблиц

```
SELECT 10*(a-1)+b num
```

```
FROM
```

```
(SELECT 1 a UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4  
UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7  
UNION ALL SELECT 8 UNION ALL SELECT 9 UNION ALL SELECT 10
```

```
) x CROSS JOIN
```

```
(SELECT 1 b UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4  
UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7  
UNION ALL SELECT 8 UNION ALL SELECT 9 UNION ALL SELECT 10
```

```
) y;
```

num
1
10
100
11
12
13
14
15
16
17
18
19
2
20
21
22
23
24
25
26
27
28
29
3

30
31
32
33
34
35
36
37
38
39
4
40
41
42
43
44
45
46
47
48
49
5
50
51

52
53
54
55
56
57
58
59
6
60
61
62
63
64
65
66
67
68
69
7
70
71
72
73

74
75
76
77
78
79
8
80
81
82
83
84
85
86
87
88
89
9
90
91
92
93
94
95

96
97
98
99

3. Способ использования декартова произведения трех и более таблиц

```
SELECT 5*5*(a-1)+5*(b-1) + c AS num
FROM (SELECT 1 a UNION ALL SELECT 2 UNION ALL SELECT 3
      UNION ALL SELECT 4 UNION ALL SELECT 5
      ) x CROSS JOIN
      (SELECT 1 b UNION ALL SELECT 2 UNION ALL SELECT 3
      UNION ALL SELECT 4 UNION ALL SELECT 5
      ) y CROSS JOIN
      (SELECT 1 c UNION ALL SELECT 2 UNION ALL SELECT 3
      UNION ALL SELECT 4 UNION ALL SELECT 5
      ) z
WHERE 5*5*(a-1)+5*(b-1) + c <= 100
ORDER BY 1
```

4. Способ использования рекурсивных СТЕ

```
WITH <имя>[(<список столбцов>)]  
AS (  
< SELECT... > -- анкорная часть  
UNION ALL -- рекурсивная часть  
< SELECT...FROM <имя>... >  
WHERE <условие продолжения итераций>  
)
```

```
Select * from <имя>
```

```
WITH nums  
AS (  
SELECT 1 num  
UNION ALL  
SELECT num+1 FROM nums  
WHERE num+1<=100  
)
```

```
Select num from nums
```

По умолчанию количество итераций не больше 100!!!

Задача 1

Получить 100 последовательных незанятых номеров моделей, идущих за последним номером модели в таблице Product.

Идея для решения (1 способ): находим максимальный номер модели и далее, используя генерацию последовательности, 100 последующих значений с шагом 1.

```
SELECT (SELECT MAX(model) FROM Product) + 5*5*(a-1)+5*(b-1) + c AS num
FROM
    (SELECT 1 a UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5
    ) x CROSS JOIN
    (SELECT 1 b UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5
    ) y CROSS JOIN
    (SELECT 1 c UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5
    ) z
WHERE 5*5*(a-1)+5*(b-1) + c <= 100
ORDER BY 1;
```

Задача 1

Получить 100 последовательных незанятых номеров моделей, идущих за последним номером модели в таблице Product.

Идея для решения (2 способ):

Создаем CTE для нахождения максимального номера модели и далее, используем рекурсивный CTE для 100 последующих значений.

```
With  
max_model as (SELECT cast (MAX(model) as int) model FROM Product),  
num as (select model from max_model  
    Union all  
    Select model+1 from num  
    WHERE model+1<= 100+(select model from max_model)  
    )  
Select model from num
```

Нельзя использовать агрегирующие функции, группировку в рекурсивной части CTE

«GROUP BY, HAVING, or aggregate functions are not allowed in the recursive part of a recursive common table ...»

Задача 2

Вывести алфавитные символы - прописные латинские буквы

1 способ (с использованием декартова произведения таблиц с числами)

```
SELECT CHAR(ASCII('A')+5*(a-1) + b-1) AS letter
FROM    (SELECT 1 a UNION ALL SELECT 2 UNION ALL SELECT 3
        UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6
        ) x CROSS JOIN
        (SELECT 1 b UNION ALL SELECT 2 UNION ALL SELECT 3
        UNION ALL SELECT 4 UNION ALL SELECT 5
        ) y
WHERE 5*(a-1) + b <= 26
ORDER BY 1
```

letter
A
B
C
D
E
F
G
H
I
J
K
L

M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

Задача 2

Вывести алфавитные символы - прописные латинские буквы

2 способ (с использованием рекурсивных CTE)

```
WITH Letters AS  
(  
SELECT ASCII('A') code, CHAR(ASCII('A')) letter  
UNION ALL  
SELECT code+1, CHAR(code+1) FROM Letters  
WHERE code+1 <= ASCII('Z')  
)
```

```
SELECT letter FROM Letters
```

В запросе анкорной части определяем ASCII-код первой буквы алфавита и соответствующий ему символ.

В запросе рекурсивной части увеличивается ASCII-код на единицу, обращаясь к CTE в предложении FROM.

В результате к строке с первым символом будут последовательно добавляться (UNION ALL) строки со следующими буквами в порядке их ASCII-кодов.

Итерации будут продолжаться до тех пор, пока условие `code + 1 <= ascii('Z')` будет истинным, т.е. пока не будет добавлена буква "Z".

Оператор

```
SELECT letter FROM Letters
```

служит для обращения к CTE, запуска рекурсии и вывода результата.

Задача 3

Вывести последовательность дат между 20.02.2016 и 05.03.2016.

WITH

dates as

(SELECT CONVERT (date,'20160220') d

UNION ALL

SELECT dateadd(dd,1,d) FROM dates

WHERE dateadd(dd,1,d) <= (SELECT CONVERT (date,'20160305') a)
)

SELECT d FROM dates

d
2016-02-26
2016-02-27
2016-02-28
2016-02-29
2016-03-01
2016-03-02
2016-03-03
2016-03-04
2016-03-05

Задача 4

Вывести последовательность дат между датами первого и последнего полета пассажира с id_psg=5.

```
WITH  
maxdate as  
(SELECT max(date) max_date FROM pass_in_trip where id_psg = 5),  
dates AS  
(  
SELECT MIN(date) d FROM pass_in_trip where id_psg = 5  
UNION ALL  
SELECT d+1 FROM dates  
WHERE d+1 <= (SELECT max_date FROM maxdate)  
)  
  
SELECT d FROM dates
```

Нельзя использовать агрегирующие функции, группировку в рекурсивной части CTE «GROUP BY, HAVING, or aggregate functions are not allowed in the recursive part of a recursive common table ...»

d

2003-04-13 00:00:00.000

2003-04-14 00:00:00.000

2003-04-15 00:00:00.000

2003-04-16 00:00:00.000

2003-04-17 00:00:00.000

2003-04-18 00:00:00.000

2003-04-19 00:00:00.000

2003-04-20 00:00:00.000

2003-04-21 00:00:00.000

2003-04-22 00:00:00.000

2003-04-23 00:00:00.000

2003-04-24 00:00:00.000

2003-04-25 00:00:00.000

Задача 66

Для всех дней в интервале с 01/04/2003 по 07/04/2003
определить число рейсов из Rostov. Вывод: дата, количество
рейсов

Trip(**trip_no**, ID_comp, plane, **town_from**, town_to, time_out, time_in)
Pass_in_trip(**trip_no**, **date**, ID_psg, place)

Заменим задачу:

Для всех дней определить число рейсов из Rostov.
Вывод: дата, количество рейсов

Задача 66

Для всех дней в интервале с 01/04/2003 по 07/04/2003
определить число рейсов из Rostov. Вывод: дата, количество
рейсов

Для всех дней определить число рейсов из Rostov.

Вывод: дата, количество рейсов

```
SELECT date, COUNT(DISTINCT trip_no) AS tripnum
FROM Pass_in_trip AS PiT
where PiT.trip_no IN (SELECT trip_no FROM Trip WHERE
town_from='Rostov')
GROUP BY date
```


date	tripnum
2003-04-01 00:00:00.000	1
2003-04-05 00:00:00.000	1
2003-04-08 00:00:00.000	1
2003-04-13 00:00:00.000	1
2003-04-14 00:00:00.000	1
2003-04-29 00:00:00.000	1

Создадим дополнительную таблицу с датами между 01/04/2003 и 07/04/2003

- В запросе анкорной части CTE используйте преобразование типов данных:

```
CONVERT (datetime,'20030401') tripdate
```

- В запросе рекурсивной части CTE используйте функцию `Dateadd (dd,1,tripdate)` для увеличения даты на один день

Создадим дополнительную таблицу с датами между
01/04/2003 и 07/04/2003

WITH

Calend AS

(SELECT CONVERT(datetime,'20030401') AS tripdate

UNION ALL

SELECT dateadd (dd,1,tripdate)

FROM Calend

WHERE tripdate<'20030407')

К запросу

```
SELECT date, COUNT(DISTINCT trip_no) AS tripnum  
FROM Pass_in_trip AS PiT  
where PiT.trip_no IN (SELECT trip_no FROM Trip WHERE  
town_from='Rostov')  
GROUP BY date
```

добавим левое соединение с таблицей CALEND по полю trip_no , а дату будем выводить из таблицы CALEND.

Преобразуем запрос

добавим левое соединение с таблицей CALEND по полю trip_no , а дату будем выводить из таблицы CALEND.

```
SELECT Calend.tripdate, tripnum
```

```
FROM Calend LEFT JOIN
```

```
(select date, COUNT(DISTINCT trip_no) AS tripnum from Pass_in_trip AS PiT
```

```
where PiT.trip_no IN (SELECT trip_no FROM Trip WHERE  
town_from='Rostov')
```

```
group by date
```

```
) y ON Calend.tripdate=y.date
```

tripdate	tripnum
2003-04-01 00:00:00.000	1
2003-04-02 00:00:00.000	NULL
2003-04-03 00:00:00.000	NULL
2003-04-04 00:00:00.000	NULL
2003-04-05 00:00:00.000	1
2003-04-06 00:00:00.000	NULL
2003-04-07 00:00:00.000	NULL

Обработка пустые значения

```
SELECT Calend.tripdate, coalesce(tripnum, 0)
```

```
FROM Calend LEFT JOIN
```

```
(select date, COUNT(DISTINCT trip_no) AS tripnum from Pass_in_trip AS PiT
```

```
where PiT.trip_no IN (SELECT DISTINCT trip_no FROM Trip WHERE  
town_from='Rostov')
```

```
group by date
```

```
) y ON Calend.tripdate=y.date
```

tripdate	tripnum
2003-04-01 00:00:00.000	1
2003-04-02 00:00:00.000	0
2003-04-03 00:00:00.000	0
2003-04-04 00:00:00.000	0
2003-04-05 00:00:00.000	1
2003-04-06 00:00:00.000	0
2003-04-07 00:00:00.000	0

Другой вариант

WITH Calend AS

(SELECT CONVERT(datetime,'20030401') AS tripdate

UNION ALL

SELECT dateadd (dd,1,tripdate)

FROM Calend

WHERE tripdate<'20030407')

SELECT tripdate, COUNT(DISTINCT trip_no) AS tripnum

FROM Calend LEFT JOIN Pass_in_trip ON tripdate=date

and trip_no IN (SELECT trip_no FROM Trip WHERE town_from='Rostov')

GROUP BY tripdate

```
SELECT date, max(tripnum)
FROM (
SELECT date, COUNT(DISTINCT trip_no) AS tripnum
FROM Pass_in_trip
WHERE trip_no IN (SELECT trip_no FROM Trip WHERE town_from='Rostov') and
date>='2003-04-01' AND date<='2003-04-07'
GROUP BY date
```

```
UNION ALL
SELECT '2003-04-01',0
UNION ALL
..... -- по всем датам
SELECT '2003-04-06',0
UNION ALL
SELECT '2003-04-07',0
) AS t
GROUP BY date
```

Разбор упражнения 121 (3 балла) из ЛР 6.

Разбор упражнения 151 (4 балла) из ЛР 11.

Упражнение №151 (4 балла) из ЛР 11

Для каждого корабля из таблицы Ships указать название **первого по времени сражения** из таблицы Battles, в котором корабль мог бы участвовать после спуска на воду.

Если год спуска на воду неизвестен, взять **последнее по времени сражение**.

Если нет сражения, произошедшего после спуска на воду корабля, вывести NULL вместо названия сражения.

Считать, что корабль может участвовать во всех сражениях, которые произошли в год спуска на воду корабля.

Вывод: имя корабля, год спуска на воду, название сражения

Замечание: считать, что не существует двух битв, произошедших в один и тот же день.

WITH SIB AS (

```
SELECT Ships.name as s_name, launched, Battles.name as b_name, date  
FROM Ships LEFT JOIN Battles ON launched<=DATEPART(yyyy, date)  
OR launched IS NULL)
```

s_name	launched	b_name	date
California	1921	#Cuba62a	1962-10-20 00:00:00.000
California	1921	#Cuba62b	1962-10-25 00:00:00.000
California	1921	Guadalcanal	1942-11-15 00:00:00.000
California	1921	North Atlantic	1941-05-25 00:00:00.000
California	1921	North Cape	1943-12-26 00:00:00.000
California	1921	Surigao Strait	1944-10-25 00:00:00.000
New Jersey	1943	#Cuba62a	1962-10-20 00:00:00.000
New Jersey	1943	#Cuba62b	1962-10-25 00:00:00.000
New Jersey	1943	North Cape	1943-12-26 00:00:00.000
New Jersey	1943	Surigao Strait	1944-10-25 00:00:00.000
Название корабля1	Null	Null
Название корабля2	Null		

```
*) SELECT b_name
   FROM SIB as SIB1
   WHERE SIB1.s_name=Sib.s_name AND
date IN (SELECT MIN(date)
         FROM SIB as SIB1
         WHERE SIB1.s_name=SIB.s_name)
```

Коррелированные подзапросы,
для каждой строки основного
запроса формируется подзапрос,
выбирающий минимум/ максимум
среди части строк всей таблицы
(для текущего корабля)

```
***) SELECT b_name
   FROM SIB as SIB1
   WHERE SIB1.s_name=Sib.s_name AND
date IN (SELECT MAX(date)
         FROM SIB as SIB1
         WHERE SIB1.s_name=SIB.s_name)
```

```
WITH SIB AS (  
SELECT Ships.name as s_name, launched, Battles.name as b_name,  
date  
FROM Ships LEFT JOIN Battles ON launched<=DATEPART(yyyy, date)  
OR launched IS NULL)
```

```
Select distinct s_name, launched,  
case  
when launched<=DATEPART(yyyy, date)  
then *  
when launched IS NULL  
then **  
end as b_name
```

```
From SIB
```

Другое решение

```
with t as
(select name, launched,
 (select case
      when s.launched is null
      then max(date)
      else min(date)
      end
 from Battles
 where datepart(year,date) > =coalesce(s.launched,0)
 ) as date
from Ships s)

select t.name, t.launched, b.name
from t left join Battles b on t.date=b.date
```