



Вложенные подзапросы

- ✓ Уровень вложенности
- ✓ Конструкция NOT EXISTS
- ✓ Конструкция ANY, ALL
- ✓ Запрос в виде таблицы
- ✓ Запрос в столбце
- ✓ Создание таблицы с помощью запроса

Проверка на принадлежность результатов вложенного запроса (ключевое слово IN) является видоизменной формой на членство в множестве.

```
SELECT login
FROM user
WHERE id_user IN (SELECT id_user FROM group
WHERE id_group in(1,3));
```

Конструкция NOT EXISTS, EXISTS

В результате проверки на существование (EXISTS) можно выяснить, содержится ли в таблице результатов вложенного запроса хотя бы одна строка.

```
SELECT login
  FROM user AS u
 WHERE u.dt_create BETWEEN '2017-01-01' AND '2017-07-01'
 NOT EXISTS (SELECT 1 FROM grp g
              WHERE g.id_user = u.id_user);
```

Конструкция ANY

В проверки ANY используется один из шести операторов сравнения (=, <>, <, <=, >, >=) для того чтобы сравнить одно проверяемое значение со столбцом данных, возвращенным вложенным запросом. Проверяемое значение поочередно сравнивается с каждым значением, содержащимся в столбце. Если любое из этих сравнений дает результат TRUE, то проверка ANY возвращает значение TRUE

```
SELECT login
  FROM user AS u
 WHERE u.dt_create BETWEEN '2017-01-01' AND '2017-
07-01'
    AND 10 < ANY (SELECT id_group
                  FROM group g
                  WHERE g.id_user = u.id_user );
```

Конструкция ALL

В проверки ALL как и в ANY используется один из шести операторов (=, <>, <, <=, >, >=) для сравнения одного проверяемое значения со столбцом данных, возвращенным вложенным запросом. Проверяемое значение поочередно сравнивается с каждым значением, содержащимся в столбце. Если все сравнения дают результат TRUE, то проверка ANY возвращает значение TRUE

```
SELECT login
  FROM user AS u
 WHERE u.dt_create BETWEEN '2017-01-01' AND '2017-07-01'
        10 < ALL (SELECT id_group
                  FROM group g
                  WHERE g.id_user = u.id_user);
```

Результат выполнения запроса можно оформить в виде таблицы задав ее в конструкции FROM.

```
SELECT login
  FROM user a, (SELECT id_user    FROM group
WHERE id_group in(1,3))b
  WHERE a.id_user = b.id_user
```

Результат выполнения запроса можно оформить в виде таблицы.

```
SELECT login
  FROM user a, (SELECT id_user    FROM group
WHERE id_group in(1,3))b
WHERE a.id_user = b.id_user
```

Результат выполнения запроса можно вставить в качестве столбца.

```
SELECT id_user, (SELECT max(id_user) FROM group)  
max_id_user  
FROM user
```


Новую таблицу можно создать на базе подзапроса из существующей таблице или множества таблиц.

```
CREATE TABLE new_table  
  AS (SELECT * FROM old_table)
```

```
CREATE TABLE new_table  
  AS (SELECT column1, column2, ... column_n  
FROM old_table1, old_table2, ... old_table_n);
```