# Задание (SQL)

**Важно!**

При проверке мы не запускаем запросы, а смотрим только на логику их построения.

Т.е. запрос может содержать незначительные синтаксические ошибки.

Нет необходимости добиваться фактической их работоспособности т.к. на это может не хватить времени выделенного на выполнение задания.

1. Что вернет запрос при последовательном выполнении команд?

**create** t**able** tab1

(

id int **not** **null**,

val int

)**;**

**select** 'abc' **from** tab1

**union**

**select** 'abc' **from** tab1

**union** **all**

**select** 'abcdefg' **from** tab1**;**

Ответ:

**UNION удаляет дубликат и возвращает только одну строку 'abc'**

**В результате будет две строки: 'abc' и 'abcdefg'.**

2. Что вернут запросы (пункты 2.1, 2.2, 2.3)?

**create** **table** tab1

(

id int **not** **null**,

val int

)**;**

**insert** **into** tab1 (id, val) **values** (1, 1)**;**

**insert** **into** tab1 (id, val) **values** (2, 2)**;**

**insert** **into** tab1 (id, val) **values** (3, 3)**;**

**insert** **into** tab1 (id, val) **values** (4, **null**)**;**

2.1

**select** count(\*), count(id), count(val), max(val) **from** tab1**;**

Ответ:

**4 , 4, 3, 3**

2.2

**select** \* **from** tab1 **where** val **in** (**select** val **from** tab1)**;**

Ответ:

**Id val**

**1 1**

**2 2**

**3 3**

2.3

**select** \* **from** tab1 **where** val **not** **in** (**select** val **from** tab1)**;**

Ответ:

**Результат будет пустым**

3. Напишите запрос, который выведет значения всех полей таблицы в соответсвии с уловиями:

- для каждого id\_parent выбирать только первые 3 id, отсортированные по убыванию

- если для id\_parent не набирается 3х id, то такие случаи не выводить

**table** tab2

(

id int **not** **null**,

id\_parent int

)**;**

Ответ:

**SELECT tt.\***

**FROM (**

**SELECT**

tab2.**\*,**

**ROW\_NUMBER() OVER (****PARTITON BY** id\_parent ORDER BY id DESC) AS rn,

**COUNT(‘**id**’) OVER (PARTITON BY** id\_parent**) AS cnt**

**FROM** tab2

**) AS tt**

**WHERE rn <=3 AND cnt >= 3**

4. Написать запрос, который выводит список id, у которых кол-во значений val равных 3 больше 10 штук или равных 4 больше 5 штук.

**table** tab1

(

id int **not** **null**,

val int

)**;**

Ответ:

**SELECT**

**Id**

**FROM**

**(**

**SELECT**

**id,**

**val,**

**count(‘val’) AS cnt**

**FROM tab1**

**GROUP BY id, val**

**) AS tt**

**GROUP BY id**

**HAVING**

**SUM(CASE WHEN val = 3 THEN 1 ELSE 0 END) > 10**

**OR**

**SUM(CASE WHEN val = 4 THEN 1 ELSE 0 END) > 5**

5. Написать запрос, который вернет список регионов с иформацией о количестве сотрудников у которых нет руководителя или грeйд их руководителя выше 10.

Исходные таблицы:

-- Справочник регионов

**table** tab\_region

(

id\_reg int **not** **null**,

reg\_name varchar(100) **not** **null**

)**;**

– Иерархический справочник сотрудников

**table** tab\_empl

(

id\_emp int **not** **null**,

fio varchar(100) **not** **null**,

id\_reg int, -- ИД региона

id\_emp\_manager int, -- ИД руководителя в tab\_empl

grade int **not** **null --** грейд сотрудника

)**;**

Ответ:

**SELECT   
 r.id\_reg, r.reg\_name, COUNT(e.id\_emp) AS num\_empl  
FROM   
 tab\_region r  
LEFT JOIN   
 (SELECT \* FROM tab\_empl WHERE id\_emp\_manager IS NULL OR grade > 10) e   
ON   
 r.id\_reg = e.id\_reg  
GROUP BY   
 r.id\_reg, r.reg\_name**

6. В таблицу tab1 в поле val необходимо загрузить значения val\_src из таблиц tab2 и tab3.

Написать скрипт, который вставит только те уникальные значения val\_src, которых еще не было в таблице tab1.

**table** tab1

(

id int **GENERATED** **BY** **DEFAULT** **AS** **IDENTITY** **not** **null**,

val int **null**

)**;**

**table** tab2

(

id int **not** **null**,

val\_src int

)**;**

**table** tab3

(

id int **not** **null**,

val\_src int

)**;**

Ответ:

**INSERT INTO tab1 (val)**

**SELECT val\_src**

**FROM (**

**SELECT val\_src FROM tab2**

**UNION**

**SELECT val\_src FROM tab3**

**) AS unique\_vals**

**WHERE val\_src NOT IN (SELECT val FROM tab1)**