

## Задание 1.

Примените пожалуйста [скрипт](#) для базы данных PostgreSQL. Если есть необходимость, то вы можете применить его заново, предварительно удалив таблицы командой

**DROP TABLE** *имя\_таблицы*;

Применить скрипт — это значит выполнить его через ваш используемый IDE (DataGrip, DBeaver , pgAdmin)

### Результат работы:

Table name	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated	
college	5	0	0	0	!
course	5	0	0	0	!
student	5	0	0	0	!
student_on_course	13	0	0	0	!



## Задание 2.

- а) Напишите SQL запрос, который возвращает имена студентов и их аккаунт в Telegram у которых родной город “Казань” или “Москва”. Результат отсортируйте по имени студента в убывающем порядке

### Код запроса:

```
SELECT name AS Имя, telegram_contact AS Телеграм  
FROM student  
WHERE city = 'Казань' or city = 'Москва'  
ORDER BY name DESC;
```

### Результат работы кода:

	Имя character varying 	Телеграм character varying 
1	Сергей Петров	@spetrov
2	Ильяс Мухаметшин	@ilyas
3	Иван Иванов	@ivanov1991
4	Екатерина Андреева	@kate_


- б) Напишите SQL запрос, который возвращает данные по университетам в следующем виде (один столбец со всеми данными внутри) с сортировкой по полю “полная информация”

"полная информация"
университет: Иннополис; количество студентов: 1077
университет: КФУ; количество студентов: 50000
университет: МГУ; количество студентов: 38000
университет: МФТИ; количество студентов: 7000
университет: Сколково; количество студентов: 1070

**Код запроса:**

```
SELECT CONCAT('университет: ', name, '
               количество студентов: ', size) AS
               "полная информация"
FROM college
ORDER BY "полная информация";
```

**Результат работы:**



	полная информация text	
1	университет: Иннополис; количество студентов: 1077	
2	университет: КФУ; количество студентов: 50000	
3	университет: МГУ; количество студентов: 38000	
4	университет: МФТИ; количество студентов: 7000	
5	университет: Сколково; количество студентов: 1070	

- с) Напишите SQL запрос который возвращает список университетов и количество студентов, если идентификатор университета должен быть выбран из списка 10, 30, 50. Пожалуйста примените конструкцию IN. Результат запроса отсортируйте по количеству студентов **И** затем по наименованию университета.

**Код запроса:**

```
SELECT name AS университет, size AS количество_студентов
FROM college
WHERE id IN (10, 30, 50)
ORDER BY количество_студентов, университет;
```

**Результат работы:**

	университет character varying 	количество_студентов integer 
1	Сколково	1070
2	МФТИ	7000
3	КФУ	50000



- d) Напишите SQL запрос который возвращает список университетов и количество студентов, если идентификатор университета **НЕ** должен соответствовать значениям из списка 10, 30, 50. Пожалуйста в основе примените конструкцию IN.

Результат запроса отсортируйте по количеству студентов **И** затем по наименованию университета.

**Код запроса:**

```
SELECT name AS университет, size AS количество_студентов
FROM college
WHERE id NOT IN (10, 30, 50)
ORDER BY количество_студентов, университет;
```

**Результат работы:**

	университет character varying 	количество_студентов integer 
1	Иннополис	1077
2	МГУ	38000

- е) Напишите SQL запрос который возвращает название online курсов университетов и количество заявленных слушателей. Количество заявленных слушателей на курсе

должно быть в диапазоне от 27 до 310 студентов. Результат отсортируйте по названию курса и по количеству заявленных слушателей в убывающем порядке для двух полей.

**Код запроса:**

```
SELECT name AS название_курса,  
       amount_of_students AS количество_студентов  
FROM course  
WHERE is_online = true AND amount_of_students BETWEEN 27 AND 310  
ORDER BY name, amount_of_students DESC;
```

**Результат работы:**

	название_курса character varying	количество_студентов integer
1	Введение в РСУБД	300
2	Цифровая трансформация	50