Вывести 10 последних платежей за прокат фильмов.

```
query('''
SELECT
    concat(c.first_name, ' ', c.last_name) AS customer_name,
    amount
FROM customer c
INNER JOIN payment p USING(customer_id)
WHERE p.amount > 0
ORDER BY p.payment_date DESC
LIMIT 10;
''')
```

	customer_name	amount
0	Ramona Hale	2.99
1	Becky Miles	0.99
2	Sonia Gregory	5.98
3	Daisy Bates	0.99
4	Jenny Castro	0.99
5	Margie Wade	7.98
6	Dianne Shelton	4.99
7	Cassandra Walters	3.98
8	Naomi Jennings	0.99
9	John Farnsworth	4.99

Выведите магазины, имеющие больше 300 покупателей

:
```
query('''
SELECT store_id , COUNT(*) AS num_customers
FROM customer
GROUP BY store_id
HAVING COUNT(*) > 300;
''')
```

:

	store_id	num_customers
0	1	326

Выведите у каждого покупателя город в котором он живет

```
query('''
SELECT concat(c.first_name, ' ', c.last_name) AS customer_name,
    c2.city
FROM customer c
INNER JOIN address a USING(address_id)
INNER JOIN city c2 USING(city_id)
ORDER BY c2.city, c.first_name;
''')
```

	customer_name	city
0	Peggy Myers	Abha
1	Tom Milner	Abu Dhabi
2	Julie Sanchez	A Corua (La Corua)
3	Glen Talbert	Acua
4	Larry Thrasher	Adana
...	...	...
594	Constance Reid	Zaria
595	Jack Foust	Zeleznogorsk
596	Byron Box	Zhezqazghan
597	Guy Brownlee	Zhoushan
598	Ronnie Ricketts	Ziguinchor

599 rows × 2 columns

Выведите ФИО сотрудников и города магазинов, имеющих больше 300 покупателей

```
query('''
SELECT concat(s.first_name,' ', s.last_name) AS staff_name, c.city
FROM staff s
INNER JOIN store s2 USING(store_id)
INNER JOIN address a ON s2.address_id = a.address_id
INNER JOIN city c USING(city_id)
WHERE s.store_id IN (
    SELECT store_id
    FROM customer
    GROUP BY store_id
    HAVING COUNT(*) > 300
);
''')
```

	staff_name	city
0	Mike Hillyer	Lethbridge

Выведите количество актеров, снимавшихся в фильмах, которые сдаются в прокат за 2,99

```
query('''
SELECT f.title , count(fa.actor_id) AS num_actors
FROM film f
INNER JOIN film_actor fa USING(film_id)
WHERE f.rental_rate = 2.99
GROUP BY f.film_id
ORDER BY num_actors DESC;
''')
```

	title	num_actors
0	Chitty Lock	13
1	Crazy Home	13
2	Random Go	13
3	Hellfighters Sierra	12
4	Lonely Elephant	12
...	...	...
318	Baked Cleopatra	1
319	Ferris Mother	1
320	Lolita World	1
321	Dwarfs Alter	1
322	Miracle Virtual	1

323 rows × 2 columns

Проектирование базы данных

Спроектируйте базу данных для следующих сущностей:

- Язык (в смысле английский, французский и тп)
- Народность (в смысле славяне, англосаксы и тп)
- Страны (в смысле Россия, Германия и тп)

Правила следующие:

- На одном языке может говорить несколько народностей
- Одна народность может входить в несколько стран
- Каждая страна может состоять из нескольких народностей

Для того, чтобы соответсвовать описанию условию задачи - 'Таким образом должно получиться 5 таблиц. Три таблицы-справочника и две таблицы со связями.', должно быть еще одно условие:

- **Одна народность может говорить на нескольких языках**
  например: славяне - русский, украинский, белорусский

```
# создадим соединение с postgres
con_postgres = create_engine('postgresql+psycopg2://postgres:1234@localh
ost:5432').connect()
con_postgres.execute("commit")
```

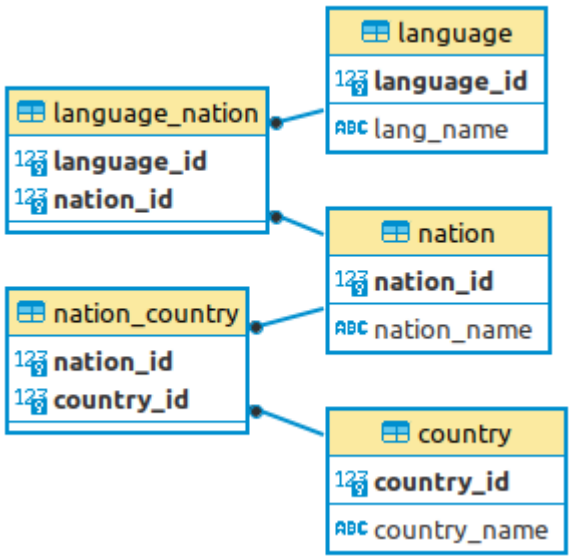
```
# создадим базу lang
con_postgres.execute("CREATE DATABASE lang;")
```

```
# закрыть соединение
con_postgres.close()
```

```
# создадим подключение к базе lang
con_lang = create_engine('postgresql+psycopg2://postgres:1234@localhost:
5432/lang')
```

```
# создаем таблицы language, nation, country со связями 'многие ко многим
'
con_lang.execute('''
CREATE TABLE language(
    language_id serial PRIMARY KEY,
    lang_name varchar(50) NOT NULL);
CREATE TABLE nation(
    nation_id serial PRIMARY KEY,
    nation_name varchar(50) NOT NULL);
CREATE TABLE country(
    country_id serial PRIMARY KEY,
    country_name varchar(50) NOT NULL);

CREATE TABLE language_nation(
    language_id int REFERENCES language (language_id),
    nation_id int REFERENCES nation (nation_id),
    PRIMARY KEY (nation_id, language_id)
);
CREATE TABLE nation_country(
    nation_id int REFERENCES nation (nation_id),
    country_id int REFERENCES country (country_id),
    PRIMARY KEY (nation_id, country_id)
);
''')
```



Сделайте запрос к таблице rental. Добавьте колонку с порядковым номером проката фильма (сортировать по rental_date) для каждого юзера

```
query('''
SELECT concat(c.first_name, ' ', c.last_name) AS customer_name,
    f.title,
    r.rental_date ,
    ROW_NUMBER() OVER (PARTITION BY c.customer_id ORDER BY r.rental_dat
e) AS r_number
FROM customer c
    JOIN rental r ON c.customer_id =r.customer_id
    JOIN inventory i ON i.inventory_id = r.inventory_id
    JOIN film f ON f.film_id = i.film_id
ORDER BY concat(c.first_name, ' ', c.last_name), r.rental_date;
''')
```

	customer_name	title	rental_date	r_number
0	Aaron Selby	Dorado Notting	2005-05-26 21:48:13	1
1	Aaron Selby	Fellowship Autumn	2005-05-27 14:17:23	2
2	Aaron Selby	Drifter Commandments	2005-05-29 09:33:33	3
3	Aaron Selby	Zhivago Core	2005-05-30 05:15:20	4
4	Aaron Selby	Muscle Bright	2005-06-15 16:38:53	5
...	...	...	...	...
16039	Zachary Hite	Name Detective	2005-08-17 13:03:13	27
16040	Zachary Hite	Victory Academy	2005-08-18 09:20:51	28
16041	Zachary Hite	Titans Jerk	2005-08-19 10:16:43	29
16042	Zachary Hite	Highball Potter	2005-08-20 18:54:59	30
16043	Zachary Hite	Frost Head	2006-02-14 15:16:03	31

16044 rows × 4 columns

Для каждого пользователя подсчитать сколько он брал в прокат фильмов со специальным атрибутом Behind the Scenes

```
query('''
SELECT concat(c.first_name, ' ', c.last_name) AS customer_name,
    count(*) AS col_films
FROM customer c
    JOIN rental r ON c.customer_id = r.customer_id
    JOIN inventory i ON i.inventory_id = r.inventory_id
    JOIN film f ON f.film_id = i.film_id
WHERE f.special_features @> '{Behind the Scenes}'
GROUP BY c.customer_id
ORDER BY col_films DESC;
''')
```

	customer_name	col_films
0	Sue Peters	29
1	Eleanor Hunt	28
2	Wesley Bull	28
3	Marcia Dean	25
4	Tommy Collazo	23
...	...	...
594	Kristen Chavez	6
595	Brent Harkins	6
596	Caroline Bowman	5
597	Chris Brothers	5
598	Brian Wyman	3

599 rows × 2 columns