

Проект: итерация 2

Велегурина Елена Б05-121

Пункт 3: Подготовить DDL скрипты и создать свою базу в СУБД.

```
CREATE SCHEMA shop;

CREATE TABLE shop.carpet
(
    carpet_id SERIAL PRIMARY KEY,
    product_id SERIAL NOT NULL,
    name VARCHAR(256) NOT NULL,
    price INTEGER CHECK (price > 0),
    amount INTEGER CHECK (amount >= 0)
);

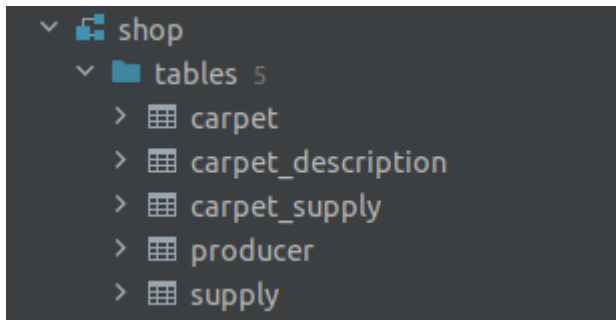
CREATE TABLE shop.producer
(
    producer_id SERIAL PRIMARY KEY,
    name VARCHAR(256) NOT NULL,
    country VARCHAR(256) NOT NULL
);

CREATE TABLE shop.supply
(
    supply_id SERIAL PRIMARY KEY,
    date_from DATE NOT NULL,
    date_to DATE NOT NULL,
    address VARCHAR(256) NOT NULL
);

CREATE TABLE shop.carpet_description
(
    carpet_id SERIAL PRIMARY KEY,
    category VARCHAR(256) NOT NULL,
    length INTEGER CHECK (length > 0),
    width INTEGER CHECK (width > 0),
    colour VARCHAR(256) NOT NULL
);

CREATE TABLE shop.carpet_supply
(
    carpet_id SERIAL NOT NULL,
    supply_id SERIAL NOT NULL,
    FOREIGN KEY (carpet_id) REFERENCES shop.carpet (carpet_id),
    FOREIGN KEY (supply_id) REFERENCES shop.supply (supply_id)
);
```

Результат:



Пункт 4: Наполнить созданную базу данными, ~10 записей в каждой таблице. По крайней мере по 1 строке в каждой таблице должно быть добавлено с использованием **INSERT**.

```
INSERT INTO shop.carpet
VALUES (1, 1, 'Кварц-виниловая LVT самоклеящаяся ПВХ плитка LAKO DECOR Делюкс', 883, 15),
(2, 1, 'Плитка ПВХ LVT Tarkett New Age DryBack 230180012 AURA', 951, 33),
(3, 2, 'Кварц-виниловый ламинат SPC Смарт Чойс Дуб Сильверсайд', 1849, 22),
(4, 1, 'Грязезащитное противоскользящее покрытие Kolckmann GmbH AKO Safety Mat Антрацит', 3649, 98),
(5, 3, 'Линолеум напольный на отрез TARKETT Sprint Pro Baden 3', 1210, 93),
(6, 4, 'Ковролин на пол офисный Betap', 1341, 103),
(7, 5, 'Коврики самоклеящиеся (ковролин/ковровая плитка)', 1427, 75),
(8, 4, 'Ламинат кроношпан Loft 4376', 1287, 63),
(9, 3, 'Искусственная трава, газон, Витебские ковры', 390, 58),
(10, 3, 'Искусственная трава газон декоративная зелень для дома сада', 519, 616);

INSERT INTO shop.producer
VALUES (1, 'France carpets', 'France'),
(2, 'Brigadir', 'Russia'),
(3, 'Магазин искусственных цветов No.1', 'Russia'),
(4, 'ИМ Грынь', 'Russia'),
(5, 'Remontnick.ru', 'Turkey');

INSERT INTO shop.supply
VALUES (1, '2022-04-03 08:30:00', '2022-04-03 18:30:00',
'Rostovskaya oblast, Rostov-na-donu, Poselkovyy 1, bld. 1/5'),
(2, '2022-04-04 09:33:00', '2022-04-05 17:34:57',
'Primorskiy kray, Vladivostok, Vatutina, bld. 26, appt. 95'),
(3, '2022-04-04 19:33:28', '2022-04-05 13:32:58',
'Saratovskaya oblast, Saratov, 2 Pionerskaya, bld. 30/36, appt. 118'),
(4, '2022-04-06 21:25:25', '2022-04-07 15:07:02',
'Vladimirskaaya oblast, Vladimir, Suzdalskiy Prt., bld. 26, appt. 134'),
(5, '2022-04-06 23:56:32', '2022-04-07 17:04:32',
'Ivanovskaya oblast, Ivanovo, Remiznaya Ul., bld. 3, appt. 54'),
(6, '2022-04-07 03:22:11', '2022-04-07 19:34:45',
'Krasnodarskiy kray, Krasnodar, Gidrostroyteley, bld. 28, appt. 236'),
(7, '2022-04-09 05:12:13', '2022-04-10 10:08:57',
'Moskovskaya oblast, Moskva, Glagoleva Generala Ul., bld. 12/K. 1, appt. 80');
```

```

INSERT INTO shop.carpet_supply
VALUES (1, 1),
      (3, 2),
      (10, 3),
      (5, 4),
      (5, 5),
      (3, 6),
      (10, 7);

INSERT INTO shop.carpet_description
VALUES (1, 'Плитка ПВХ / LVT', 15, 10, 'brown'),
      (2, 'Плитка ПВХ / LVT', 50, 40, 'aura'),
      (3, 'Плитка ПВХ / LVT', 100, 10, 'white'),
      (4, 'Грязезащитные покрытия', 5, 10, 'black'),
      (5, 'Линолеум', 3, 6, 'grey'),
      (6, 'Ковролин на резиновой основе', 3, 10, 'grey'),
      (7, 'Ковролин для гостиниц', 3, 3, 'brown'),
      (8, 'Ламинат', 2, 4, 'pine'),
      (9, 'Искусственная трава', 1, 1, 'green'),
      (10, 'Искусственная трава', 1, 1, 'green');

```

Результат:

	carpet_id	product_id	name	price	amount
1	1	1	Кварц-виниловая LVT с...	883	15
2	2	1	Плитка ПВХ LVT Tarket...	951	33
3	3	2	Кварц-виниловый ламин...	1849	22
4	4	1	Грязезащитное противо...	3649	98
5	5	3	Линолеум напольный на...	1210	93
6	6	4	Ковролин на пол офисн...	1341	103
7	7	5	Коврики самоклеящиеся...	1427	75
8	8	4	Ламинат кроношпан Lof...	1287	63
9	9	3	Искусственная трава, ...	390	58
10	10	3	Искусственная трава г...	519	616

	carpet_id	category	length	width	colour
1	1	Плитка ПВХ / LVT	15	10	brown
2	2	Плитка ПВХ / LVT	50	40	aura
3	3	Плитка ПВХ / LVT	100	10	white
4	4	Грязезащитные покрытия	5	10	black
5	5	Линолеум	3	6	grey
6	6	Ковролин на резиновой...	3	10	grey
7	7	Ковролин для гостиниц	3	3	brown
8	8	Ламинат	2	4	pine
9	9	Искусственная трава	1	1	green
10	10	Искусственная трава	1	1	green

	carpet_id	supply_id
1	1	1
2	3	2
3	10	3
4	5	4
5	5	5
6	3	6
7	10	7

	producer_id	name	country
1	1	France carpets	France
2	2	Brigadir	Russia
3	3	Магазин искусственных...	Russia
4	4	ИМ Грынн	Russia
5	5	Remontnick.ru	Turkey

	supply_id	date_from	date_to	address
1	1	2022-04-03	2022-04-03	Rostovskaya oblast, R...
2	2	2022-04-04	2022-04-05	Primorskiy kray, Vlad...
3	3	2022-04-04	2022-04-05	Saratovskaya oblast, ...
4	4	2022-04-06	2022-04-07	Vladimirskaia oblast,...
5	5	2022-04-06	2022-04-07	Ivanovskaya oblast, I...
6	6	2022-04-07	2022-04-07	Krasnodarskiy kray, K...
7	7	2022-04-09	2022-04-10	Moskovskaya oblast, M...

Пункт 5: Написать не менее 10 **INSERT**, **SELECT**, **UPDATE**, **DELETE** запросов. Загуглить, что такое CRUD-запросы. Найти соответствие.

```
-- CREATE
INSERT INTO shop.supply
VALUES (8, '2022-04-10 15:25:33', '2022-04-11 12:58:37',
      'Novosibirskaya oblast, Novosibirsk, Vybornaya Ul., bld. 106, appt. 186');

INSERT INTO shop.carpet_supply
VALUES (4, 8);
```

Результат:

	supply_id	date_from	date_to	address
1	1	2022-04-03	2022-04-03	Rostovskaya oblast, R...
2	2	2022-04-04	2022-04-05	Primorskiy kray, Vlad...
3	3	2022-04-04	2022-04-05	Saratovskaya oblast, ...
4	4	2022-04-06	2022-04-07	Vladimirskaia oblast,...
5	5	2022-04-06	2022-04-07	Ivanovskaya oblast, I...
6	6	2022-04-07	2022-04-07	Krasnodarskiy kray, K...
7	7	2022-04-09	2022-04-10	Moskovskaya oblast, M...
8	8	2022-04-10	2022-04-11	Novosibirskaya oblast...

	carpet_id	supply_id
1	1	1
2	3	2
3	10	3
4	5	4
5	5	5
6	3	6
7	10	7
8	4	8

```
-- READ
SELECT *, count(*)
FROM shop.carpet
WHERE price > 1000
GROUP BY carpet_id;

SELECT carpet_id, name, amount
FROM shop.carpet
WHERE amount > 100
GROUP BY carpet_id;

SELECT *
FROM shop.supply
ORDER BY date_from;
```

Результат:

	carpet_id	product_id	name	price	amount	count
1	4	1	Грязезащитное противоскользящее покрытие Kolckmann GmbH AKO Safety M...	3649	98	1
2	7	5	Коврики самоклеящиеся (ковролин/ковровая плитка)	1427	75	1
3	6	4	Ковролин на пол офисный Betar	1341	103	1
4	3	2	Кварц-виниловый ламинат SPC Смарт Чойс Дуб Сильверсайд	1849	22	1
5	5	3	Линолеум напольный на отрез TARKETT Sprint Pro Baden 3	1210	93	1
6	8	4	Ламинат кроношпан Loft 4376	1287	63	1

	carpet_id	name	amount
1	10	Искусственная трава газон декоративная зелень для дома сада	616
2	6	Ковролин на пол офисный Betar	103

	supply_id	date_from	date_to	address
1	1	2022-04-03	2022-04-03	Rostovskaya oblast, Rostov-na-donu, Poselkovyy 1, bld. 1/6
2	2	2022-04-04	2022-04-05	Primorskiy kray, Vladivostok, Vatutina, bld. 26, appt. 95
3	3	2022-04-04	2022-04-05	Saratovskaya oblast, Saratov, 2 Pionerskaya, bld. 30/36, appt. 118
4	4	2022-04-06	2022-04-07	Vladimirskaia oblast, Vladimir, Suzdalskiy Prt., bld. 26, appt. 134
5	5	2022-04-06	2022-04-07	Ivanovskaya oblast, Ivanovo, Remiznaya Ul., bld. 3, appt. 54
6	6	2022-04-07	2022-04-07	Krasnodarskiy kray, Krasnodar, Gidrostroyteley, bld. 28, appt. 236
7	7	2022-04-09	2022-04-10	Moskovskaya oblast, Moskva, Glagoleva Generala Ul., bld. 12/K. 1, appt. 80
8	8	2022-04-10	2022-04-11	Novosibirskaya oblast, Novosibirsk, Vybornaya Ul., bld. 106, appt. 186

```
-- UPDATE
UPDATE shop.supply
  SET address = 'Kostromskaya oblast, Kostroma, Severnoy Pravdy Ul., bld. 31, appt. 9'
WHERE supply_id = 3;

UPDATE shop.supply
  SET date_to = '2022-04-03 09:31:10'
WHERE supply_id = 1;

UPDATE shop.carpet
  SET amount = 517
WHERE carpet_id = 10;
```

Результат:

	supply_id	date_from	date_to	address
1	2	2022-04-04	2022-04-05	Primorskiy kray, Vlad...
2	4	2022-04-06	2022-04-07	Vladimirskaaya oblast,...
3	5	2022-04-06	2022-04-07	Ivanovskaya oblast, I...
4	6	2022-04-07	2022-04-07	Krasnodarskiy kray, K...
5	7	2022-04-09	2022-04-10	Moskovskaya oblast, M...
6	8	2022-04-10	2022-04-11	Novosibirskaya oblast...
7	3	2022-04-04	2022-04-05	Kostromskaya oblast, ...
8	1	2022-04-03	2022-04-03	Rostovskaya oblast, R...

	carpet_id	product_id	name	price	amount
1	1	1	Кварц-виниловая LVT с...	883	15
2	2	1	Плитка ПВХ LVT Tarket...	951	33
3	3	2	Кварц-виниловый ламин...	1849	22
4	4	1	Грязезащитное противо...	3649	98
5	5	3	Линолеум напольный на...	1210	93
6	6	4	Ковролин на пол офисн...	1341	103
7	7	5	Коврики самоклеющиеся...	1427	75
8	8	4	Ламинат кроношпан Lof...	1287	63
9	9	3	Искусственная трава, ...	390	58
10	10	3	Искусственная трава г...	519	517

```
-- DELETE
DELETE
  FROM shop.carpet_supply
WHERE supply_id = 5;

DELETE
  FROM shop.supply
WHERE supply_id = 5;
```

	supply_id	date_from	date_to	address
1	2	2022-04-04	2022-04-05	Primorskiy kray, Vlad...
2	4	2022-04-06	2022-04-07	Vladimirskaia oblast, ...
3	6	2022-04-07	2022-04-07	Krasnodarskiy kray, K...
4	7	2022-04-09	2022-04-10	Moskovskaya oblast, M...
5	8	2022-04-10	2022-04-11	Novosibirskaya oblast...
6	3	2022-04-04	2022-04-05	Kostromskaya oblast, ...
7	1	2022-04-03	2022-04-03	Rostovskaya oblast, R...

	carpet_id	supply_id
1	1	1
2	3	2
3	10	3
4	5	4
5	3	6
6	10	7
7	4	8

Пункт 6: Написать не менее 6 осмысленных **SELECT**-запросов с использованием:

1. GROUP BY + HAVING
2. ORDER BY
3. **<func>(...)** OVER(...):
 1. PARTITION BY
 2. ORDER BY
 3. PARTITION BY + ORDER BY
 4. **<func>** - все 3 типа функций - агрегирующие, ранжирующие, смещения

1.

```
-- Выбрать производителей, у которых покупаются более 1 вида ковров
SELECT carpet.producer_id, p.name, p.country
FROM shop.carpet
      INNER JOIN shop.producer p
            ON carpet.producer_id = p.producer_id
GROUP BY carpet.producer_id, p.name, p.country
HAVING count(carpet_id) > 1
```

Результат:

	producer_id	name	country
1	1	France carpets	France
2	4	ИМ Групп	Russia
3	3	Магазин искусственных цветов No...	Russia

2.

```
-- Выбрать самые дешёвые товары, стоимость которых меньше 1500
SELECT name, price
FROM shop.carpet
WHERE price < 1500
      AND amount > 1
ORDER BY name;
```

Результат:

	name	price
1	Искусственная трава газон декоративная зелень для дома сада	519
2	Искусственная трава, газон, Витебские ковры	390
3	Кварц-виниловая LVT самоклеящаяся ПВХ плитка LAKO DECOR Делюкс	883
4	Коврики самоклеящиеся (ковролин/ковровая плитка)	1427
5	Ковролин на пол офисный Ветар	1341
6	Ламинат кроношпан Loft 4376	1287
7	Линолеум напольный на отрез TARKETT Sprint Pro Baden 3	1210
8	Плитка ПВХ LVT Tarkett New Age DryBack 230180012 AURA	951

3.

```
-- Установить соответствие между коврами и категориями,
-- пронумеровав категории
SELECT name,
       category,
       dense_rank() OVER (ORDER BY category) AS category_num
FROM shop.carpet
     INNER JOIN shop.carpet_description c
           ON carpet.carpet_id = c.carpet_id
```

Результат:

	name	category	category_num
1	Грязезащитное противоскользящее пок...	Грязезащитные покрытия	1
2	Искусственная трава газон декоратив...	Искусственная трава	2
3	Искусственная трава, газон, Витебск...	Искусственная трава	2
4	Коврики самоклеящиеся (ковролин/ков...	Ковролин для гостиниц	3
5	Ковролин на пол офисный Ветар	Ковролин на резиновой основе	4
6	Ламинат кроношпан Loft 4376	Ламинат	5
7	Линолеум напольный на отрез TARKETT...	Линолеум	6
8	Кварц-виниловая LVT самоклеящаяся П...	Плитка ПВХ / LVT	7
9	Кварц-виниловый ламинат SPC Смарт Ч...	Плитка ПВХ / LVT	7
10	Плитка ПВХ LVT Tarkett New Age DryB...	Плитка ПВХ / LVT	7

4.

```
-- Сравнение цен в каждой категории с минимальной
SELECT name,
       category,
       price,
       first_value(price)
       OVER (PARTITION BY category
            ORDER BY price)
            AS min_category_price
FROM shop.carpet
     INNER JOIN shop.carpet_description c
           ON carpet.carpet_id = c.carpet_id;
```

Результат:

	name	category	price	min_category_price
1	Грязезащитное противоскользящее пок...	Грязезащитные покрытия	3649	3649
2	Искусственная трава, газон, Витебск...	Искусственная трава	390	390
3	Искусственная трава газон декоратив...	Искусственная трава	519	390
4	Коврики самоклеящиеся (ковролин/ков...	Ковролин для гостиц	1427	1427
5	Ковролин на пол офисный Betar	Ковролин на резиновой основе	1341	1341
6	Ламинат кроношпан Loft 4376	Ламинат	1287	1287
7	Линолеум напольный на отрез TARKETT...	Линолеум	1210	1210
8	Кварц-виниловая LVT самоклеящаяся П...	Плитка ПВХ / LVT	883	883
9	Плитка ПВХ LVT Tarkett New Age DryB...	Плитка ПВХ / LVT	951	883
10	Кварц-виниловый ламинат SPC Смарт Ч...	Плитка ПВХ / LVT	1849	883

5.

```
-- Посчитать число ковров в зависимости от цвета
SELECT name,
       colour,
       count(c.carpet_id) OVER (PARTITION BY c.colour) AS colour_count
FROM shop.carpet
      INNER JOIN shop.carpet_description c
      ON carpet.carpet_id = c.carpet_id;
```

Результат:

	name	colour	colour_count
1	Плитка ПВХ LVT Tarkett New Age DryB...	aura	1
2	Грязезащитное противоскользящее пок...	black	1
3	Коврики самоклеящиеся (ковролин/ков...	brown	2
4	Кварц-виниловая LVT самоклеящаяся П...	brown	2
5	Искусственная трава газон декоратив...	green	2
6	Искусственная трава, газон, Витебск...	green	2
7	Линолеум напольный на отрез TARKETT...	grey	2
8	Ковролин на пол офисный Betar	grey	2
9	Ламинат кроношпан Loft 4376	pine	1
10	Кварц-виниловый ламинат SPC Смарт Ч...	white	1

6.

```
-- Соответствие между наименованием ковра и местом и временем его доставки
SELECT name, address, date_from, date_to
FROM shop.carpet
      INNER JOIN shop.carpet_supply cs
      ON carpet.carpet_id = cs.carpet_id
      INNER JOIN shop.supply s
      ON cs.supply_id = s.supply_id
ORDER BY date_from
```

Результат:

	name	address	date_from	date_to
1	Кварц-виниловая LVT самоклеящаяся ПВХ...	Rostovskaya oblast, Rostov-na-donu, ...	2022-04-03	2022-04-03
2	Кварц-виниловый ламинат SPC Смарт Чо...	Primorskiy kray, Vladivostok, Vatuti...	2022-04-04	2022-04-05
3	Искусственная трава газон декоративн...	Kostromskaya oblast, Kostroma, Sever...	2022-04-04	2022-04-05
4	Линолеум напольный на отрез TARKETT ...	Vladimirskaia oblast, Vladimir, Suzd...	2022-04-06	2022-04-07
5	Кварц-виниловый ламинат SPC Смарт Чо...	Krasnodarskiy kray, Krasnodar, Gidro...	2022-04-07	2022-04-07
6	Искусственная трава газон декоративн...	Moskovskaya oblast, Moskva, Glagolev...	2022-04-09	2022-04-10
7	Грязезащитное противоскользящее покр...	Novosibirskaya oblast, Novosibirsk, ...	2022-04-10	2022-04-11

Пункт 7: Написать тесты к запросам из предыдущего пункта + приложить описание самих запросов и тестов к ним.

1. В первом запросе выбираем дешёвые ковры, стоимость которых не превосходит 1500. Упорядочиваем выбранные товары по названию. В тесте проверяем, что выбраны 2 столбца, товары действительно отсортированы в алфавитном порядке и их стоимость менее 1500.
2. Во втором запросе выбираем производителей, у которых покупаются более 1 вида ковров. В тесте проверяем, что выбраны 4 столбца и количество заказываемых ковров у каждого производителя более 1.
3. В третьем запросе устанавливаем соответствие между коврами и категориями, пронумеровав категории. В тесте проверяем, что выбраны 3 столбца и нумерация верна.
4. В четвёртом запросе считаем число ковров каждого цвета. В тесте проверяем, что выбраны 3 столбца и количество ковров каждого цвета действительно совпадает с найденным в sql-запросе
5. В пятом запросе сравниваем цены на ковры в каждой категории с минимальной. В тесте проверяем, что выбраны 4 столбца и минимальная цена по категории не превосходит всех остальных.
6. В шестом запросе устанавливаем соответствие между наименованием ковра и временем и местом его доставки. В тесте проверяем, что выбраны 4 столбца и дата начала отправки не превосходит даты окончания.

```
import pandas as pd
import os
import psycopg2 as pg
import unittest
from dataclasses import dataclass

@dataclass
class Credentials:
    dbname: str = "pg_db"
    host: str = "127.0.0.1"
    port: int = 5432
    user: str = "postgres"
    password: str = "postgres"
```

```
def psycopg2_conn_string():
    return f"""
        dbname='{os.getenv("DBNAME", Credentials.dbname)}'
        user='{os.getenv("DBUSER", Credentials.user)}'
        host='{os.getenv("DBHOST", Credentials.host)}'
        port='{os.getenv("DBPORT", Credentials.port)}'
        password='{os.getenv("DBPASSWORD", Credentials.password)}'
    """
```

```
def set_connection():
    return pg.connect(psycopg2_conn_string())
```

```
class TestHardQueries(unittest.TestCase):
    def __init__(self, *args, **kwargs):
        super(TestHardQueries, self).__init__(*args, **kwargs)
        self.conn = set_connection()
        self.cursor = self.conn.cursor()
```

```
def test1(self):
    query = """
```

```
        SELECT name, price
        FROM shop.carpet
        WHERE price < 1500
        AND amount > 1
        ORDER BY name;
    """
```

```
    result = pd.read_sql(query, con=self.conn)
    assert result.shape[1] == 2
    assert result.iloc[0].loc['price'] < 1500
    for i in range(1, result.shape[0]):
        assert result.iloc[i].loc['name'] >= \
            result.iloc[i - 1].loc['name']
        assert result.iloc[i].loc['price'] < 1500
```

```
def test2(self):
    query = """
```

```
        SELECT carpet.producer_id,
               p.name,
               p.country,
               count(carpet_id) AS carpet_number
        FROM shop.carpet
        INNER JOIN shop.producer p
        ON carpet.producer_id =
p.producer_id
```

```
        GROUP BY carpet.producer_id, p.name, p.country
        HAVING count(carpet_id) > 1;
```

```

"""
result = pd.read_sql(query, con=self.conn)
assert result.shape[1] == 4
for i in range(result.shape[0]):
    assert result.iloc[i].loc['carpet_number'] > 1

def test3(self):
    query = """
        SELECT name,
               category,
               dense_rank() OVER (ORDER BY category) AS
category_num
        FROM shop.carpet
             INNER JOIN shop.carpet_description c
             ON carpet.carpet_id = c.carpet_id;
    """
    result = pd.read_sql(query, con=self.conn)
    assert result.shape[1] == 3
    for i in range(1, result.shape[0]):
        assert result.iloc[i].loc['category_num'] >= \
            result.iloc[i - 1].loc['category_num']

def test4(self):
    query = """
        SELECT name,
               colour,
               count(c.carpet_id) OVER (PARTITION BY c.colour)
AS colour_count
        FROM shop.carpet
             INNER JOIN shop.carpet_description c
             ON carpet.carpet_id = c.carpet_id;
    """
    result = pd.read_sql(query, con=self.conn)
    assert result.shape[1] == 3

    d = dict()
    for i in range(result.shape[0]):
        col = result.iloc[i].loc['colour']
        if col in d:
            d[col] += 1
        else:
            d[col] = 1

    for i in range(result.shape[0]):
        col = result.iloc[i].loc['colour']
        assert d[col] == result.iloc[i].loc['colour_count']

```

```

def test5(self):
    query = """
        SELECT name,
               category,
               price,
               first_value(price)
               OVER (PARTITION BY category
                     ORDER BY price)
               AS min_category_price
        FROM shop.carpet
             INNER JOIN shop.carpet_description c
             ON carpet.carpet_id = c.carpet_id;
    """
    result = pd.read_sql(query, con=self.conn)
    assert result.shape[1] == 4
    for i in range(result.shape[0]):
        assert result.iloc[i].loc['min_category_price'] <= \
            result.iloc[i].loc['price']

def test6(self):
    query = """
        SELECT name, address, date_from, date_to
        FROM shop.carpet
             INNER JOIN shop.carpet_supply cs
             ON carpet.carpet_id = cs.carpet_id
             INNER JOIN shop.supply s
             ON cs.supply_id = s.supply_id
        ORDER BY date_from
    """
    result = pd.read_sql(query, con=self.conn)
    assert result.shape[1] == 4
    assert result.iloc[0].loc['date_from'] <=
result.iloc[0].loc['date_to']
    for i in range(1, result.shape[0]):
        assert result.iloc[i].loc['date_from'] >= \
            result.iloc[i - 1].loc['date_from']
        assert result.iloc[i].loc['date_from'] <=
result.iloc[i].loc['date_to']

def end(self):
    self.cursor.close()
    self.conn.close()

```

Результат:

✓ Tests passed: 6 of 6 tests – 117 ms

```
main.py::TestHardQueries::test1
main.py::TestHardQueries::test2
main.py::TestHardQueries::test3
main.py::TestHardQueries::test4
main.py::TestHardQueries::test5
main.py::TestHardQueries::test6
```

```
===== 6 passed, 6 warnings in 0.81s =====
PASSED [ 16%]PASSED
Process finished with exit code 0
```