## МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

#### "НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ" КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

### **Лабораторная работа №1 по дисциплине** Технологии баз данных

вариант задания № 1

Факультет: ФПМИ Группа: ПМ-13 Бригада №3

Студенты: Вострецова Екатерина

Исакин Даниил

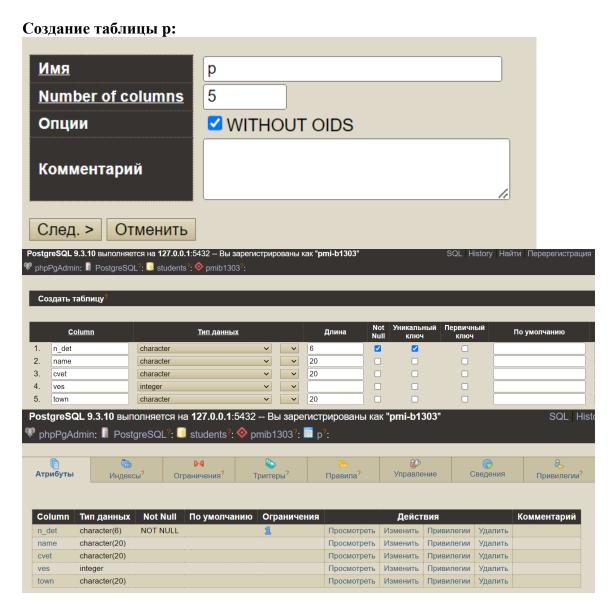
Преподаватели: Стасышина Т.Л. Сивак М.А.

#### Часть 1:

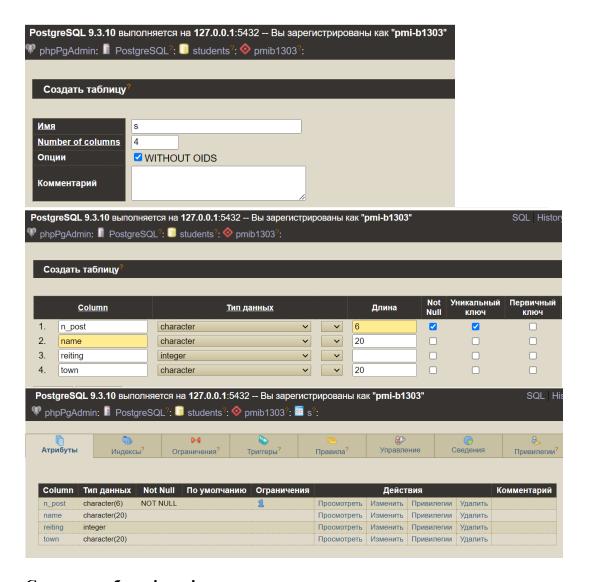
Описание процесса создания таблиц:

- 1. Используя программу phpPgAdmin, в созданной схеме создать четыре таблицы. При создании таблиц предусмотреть выполнение следующих условий:
  - поля номер поставщика, номер детали, номер изделия во всех таблицах имеет символьный тип и длину 6;
  - поля рейтинг, вес и количество имеют целочисленный тип;
  - поля фамилия, город (поставщика, детали или изделия), название (детали или изделия) имеют символьный тип и длину 20;
  - ни для одного поля не предусматривается использование индексов;
  - для всех полей допускаются значения NULL и значения дубликаты, кроме полей номер поставщика из таблицы S, номер детали из таблицы P, номер изделия из таблины J.

Таблицы S и P создать средствами системы меню программы phpPgAdmin, таблицы J и SPJ создать, написав и выполнив соответствующие запросы для создания таблиц (команда Create table).



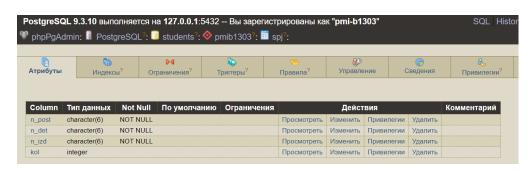
Создание таблицы s:

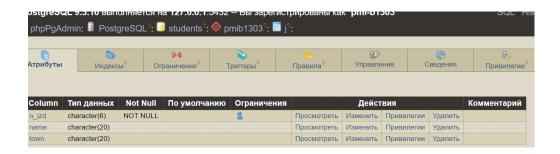


#### Создание таблиц ј и ѕрј:

## CREATE TABLE j( n\_izd character(6) NOT NULL UNIQUE, name character(20), town character(20) );

# CREATE TABLE spj( n\_post character(6) NOT NULL, n\_det character(6) NOT NULL, n\_izd character(6) NOT NULL, kol integer ) witn OIDS;



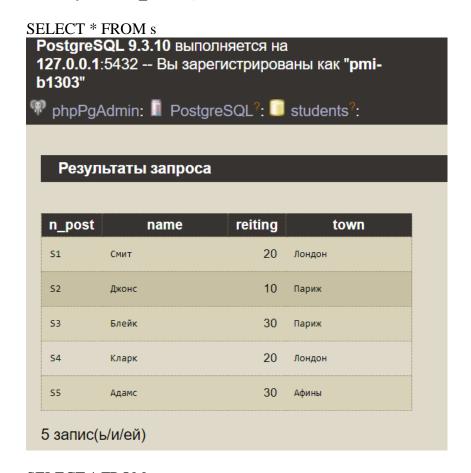


2. Записать и выполнить совокупность запросов для занесения нижеприведенных данных

```
INSERT INTO s (n_post, name, reiting, town) VALUES
('S1', 'Смит', 20, 'Лондон'),
('S2', 'Джонс', 10, 'Париж'),
('S3', 'Блейк', 30, 'Париж'),
('S4', 'Кларк', 20, 'Лондон'),
('S5', 'Адамс', 30, 'Афины');
INSERT INTO p (n_det, name, cvet, ves, town) VALUES
('Р1', 'Гайка', 'Красный', 12, 'Лондон'),
('Р2', 'Болт', 'Зелёный', 17, 'Париж'),
('Р3', 'Винт', 'Голубой', 17, 'Рим'),
('Р4', 'Винт', 'Красный', 14, 'Лондон'),
('Р5', 'Кулачок', 'Голубой', 12, 'Париж'),
('Р6', 'Блюм', 'Красный', 19, 'Лондон');
INSERT INTO j (n_izd, name, town) VALUES
('Ј1', 'Жёсткий диск', 'Париж'),
('Ј2', 'Перфоратор', 'Рим'),
('ЈЗ', 'Считыватель', 'Афины'),
('Ј4', 'Принтер', 'Афины'),
('Ј5', 'Флоппи-диск', 'Лондон'),
('Ј6', 'Терминал', 'Осло'),
('Ј7', 'Лента', 'Лондон');
INSERT INTO spj (n_post, n_det, n_izd, kol) VALUES
('S1', 'P1', 'J1', 200),
('S1', 'P1', 'J4', 700),
('S2', 'P3', 'J1', 400),
('S2', 'P3', 'J2', 200),
('S2', 'P3', 'J3', 200),
('S2', 'P3', 'J4', 500),
('S2', 'P3', 'J5', 600),
('S2', 'P3', 'J6', 400),
('S2', 'P3', 'J7', 800),
('S2', 'P5', 'J2', 100),
('S3', 'P3', 'J1', 200),
('S3', 'P4', 'J2', 500),
('S4', 'P6', 'J3', 300),
('S4', 'P6', 'J7', 300),
('S5', 'P2', 'J2', 200),
('S5', 'P2', 'J4', 100),
('S5', 'P5', 'J5', 500),
```

```
('S5', 'P5', 'J7', 100),
('S5', 'P6', 'J2', 200),
('S5', 'P1', 'J4', 100),
('S5', 'P3', 'J4', 200),
('S5', 'P4', 'J4', 800),
('S5', 'P5', 'J4', 400),
('S5', 'P6', 'J4', 500);
```

3. Проверить результат заполнения таблиц, написав и выполнив простейший запрос select \* from имя таблицы



#### SELECT \* FROM p



SELECT \* FROM j



SELECT \* FROM spj

PostgreSQL 9.3.10 выполняется на **127.0.0.1**:5432 -- Вы зарегистрированы как "pmi-b1303"







#### Результаты запроса

n_post	n_det	n_izd	kol
51	P1	J1	200
51	P1	J4	700
52	Р3	J1	400
S2	Р3	J2	200
52	Р3	J3	200
52	Р3	Ј4	500
52	Р3	J5	600
S2	Р3	Ј6	400

S2	Р3	37	800
52	P5	Ј2	100
<b>S</b> 3	Р3	J1	200
S3	P4	Ј2	500
S4	P6	J3	300
S4	P6	37	300
<b>S</b> 5	P2	Ј2	200
<b>S</b> 5	P2	J4	100
S5	P5	J5	500
S5	P5	37	100
<b>S</b> 5	Р6	Ј2	200
<b>S</b> 5	P1	J4	100
<b>S</b> 5	Р3	34	200
<b>S</b> 5	P4	34	800
S5	P5	J4	400
S5	Р6	J4	500
24 запи	іс(ь/и/е	й)	

4. Средствами системы меню программы phpPgAdmin выполнить модификацию структуры таблицы SPJ, добавив поле с датой поставки. Убедиться в успешности выполненных действий. При необходимости исправить ошибки. Занести произвольные даты поставки, использую меню программы phpPgAdmin



₱ phpPgAdmin: 
■ PostgreSQL?: 
■ students?: 
♦ pmib1303?: 
■ spj?:



#### Просмотреть

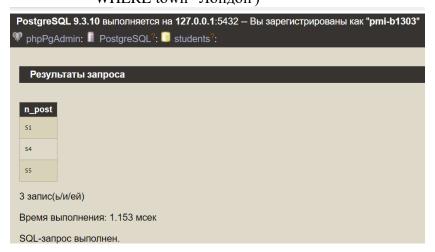
#### Запись обновлена.

Действ	Я	n_post	n_det	n_izd	kol	date
Редактировать	Удалить	S1	P1	J1	200	2023-04-03
Редактировать	Удалить	51	P1	Ј4	700	2023-12-04
Редактировать	Удалить	52	Р3	J1	400	2023-09-23
Редактировать	Удалить	52	Р3	J2	200	2024-09-19
Редактировать	Удалить	52	P3	J3	200	2024-09-23
Редактировать	Удалить	52	Р3	Ј4	500	1969-05-04
Редактировать	Удалить	52	Р3	J5	600	1959-06-03
Редактировать	Удалить	52	Р3	J6	400	1981-03-16
Редактировать	Удалить	52	Р3	37	800	1989-12-19
Редактировать	Удалить	52	P5	Ј2	100	1997-06-17
Редактировать	Удалить	53	Р3	J1	200	1972-02-23
Редактировать	Удалить	53	P4	Ј2	500	2009-02-16

#### Часть 2:

1. Выбрать поставщиков, поставляющих детали, поставляемые поставщиками, проживающими в Лондоне.

SELECT DISTINCT n\_post FROM spi WHERE n\_det in (SELECT n\_det FROM spj JOIN s ON s.n\_post=spj.n\_post WHERE town='Лондон')



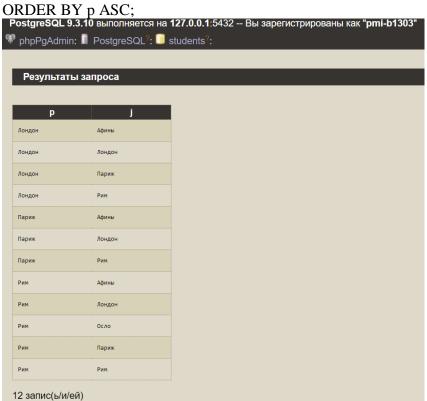
2. Вывести без повторений пары городов таких, что была поставка детали из первого города для изделия во втором городе. Упорядочить список по алфавиту.

SELECT DISTINCT p.town as p, j.town as j

FROM spj

JOIN p on (p.n\_det=spj.n\_det)

JOIN j on (j.n\_izd=spj.n\_izd)

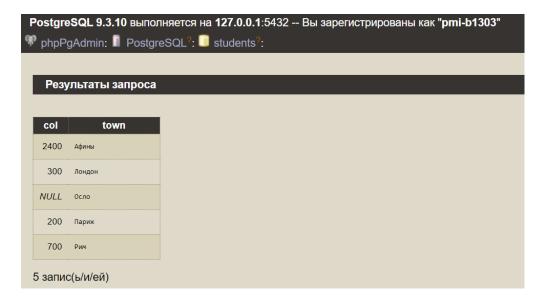


3. Получить список городов, в которые выполнили поставки ТОЛЬКО поставщики, поставлявшие голубые детали.

```
SELECT town
FROM j
JOIN spj ON spj.n_izd=j.n_izd
WHERE n_post IN(SELECT DISTINCT n_post
         FROM spj our
         JOIN p ON our.n_det=p.n_det
         WHERE cvet='Голубой')
EXCEPT
SELECT town
FROM i
JOIN spj ON spj.n_izd=j.n_izd
WHERE NOT n_post IN(SELECT DISTINCT n_post
         FROM spj our
         JOIN p ON our.n_det=p.n_det
         WHERE cvet='Голубой')
 PostgreSQL 9.3.10 выполняется на 127.0.0.1:5432 -- Вы зарегистрированы как "pmi-b1303"
💔 phpPgAdmin: 📗 PostgreSQL?: 🧻 students?:
   Результаты запроса
      town
  Осло
 2 запис(ь/и/ей)
```

4. Вывести полный список городов и для каждого города найти суммарное количество деталей красного цвета, которые были в него поставлены. Города в списке должны быть ВСЕ. Список должен быть упорядочен по алфавиту.

```
SELECT res.col, town.town
FROM (SELECT town FROM p
union
SELECT town FROM s
union
SELECT town FROM j) town
LEFT JOIN (SELECT sum(kol) col, j.town
FROM spj
JOIN p ON spj.n_det=p.n_det
JOIN j ON spj.n_izd=j.n_izd
WHERE p.cvet='Красный'
GROUP BY j.town
) res ON res.town=town.town
ORDER BY 2
```



#### Модификация информации

1. Построить таблицу с упорядоченным списком городов, в которых размещается только один поставщик, или производится только одна деталь, или собирается только одно изделие.

CREATE TABLE town (town character(20));

INSERT INTO town(SELECT town

FROM s

**GROUP BY town** 

HAVING COUNT(\*) = 1

**UNION** 

SELECT town

FROM p

**GROUP BY town** 

HAVING COUNT(\*) = 1

**UNION** 

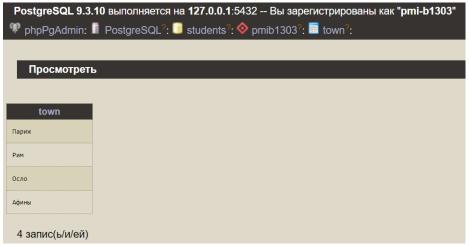
SELECT town

FROM i

GROUP BY town

HAVING COUNT(\*) = 1





2. Увеличить рейтинг всех поставщиков, имеющих в настоящее время рейтинг меньше, чем у поставщика S5, на величину, равную умноженному на 2 числу выполненных поставщиком поставок.

Таблица до модификации:



UPDATE s SET reiting=reiting + (SELECT COUNT(\*)\*2
FROM spj
WHERE s.n\_post = spj.n\_post
GROUP BY s.n\_post)

WHERE reiting<br/>
FROM s<br/>
WHERE s.n\_post = 'S5')

Таблица после обновления:

