|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Технологии баз данных» | | |
|  | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-01 |
| Вариант: | 3 |
| Студенты: | Куликов Д., Иовенко Д. |
|  |  |
| Преподаватель: | Стасышина Т. Л.,  Сивак М.А. |
|
|
|
|  |
|  |
| Новосибирск | | |

2023

**ЧАСТЬ 1**

1. Цель работы: Создать схему базы данных, ознакомиться с возможностями веб приложения phpPgAdmin, создать с его помощью набор таблиц и заполнить таблицы данными для последующей работы.
2. Из командной строки в существующей базе данных students командой new\_schema [имя схемы] [имя базы данных] создать схему базы данных с произвольным именем (имя базы данных students).



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 таблицы 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).

Запрос, для создания необходимых таблиц:

CREATE TABLE s(

n\_post character(6) NOT NULL,

name character(20),

reiting integer,

town character(20)) WITH OIDS;

CREATE TABLE p (

n\_det character(6) NOT NULL,

name character(20),

cvet character(20),

ves integer,

town character(20)) WITH OIDS;

CREATE TABLE j (

n\_izd character(6) NOT NULL,

name character(20),

town character(20)) WITH OIDS;

CREATE TABLE spj (

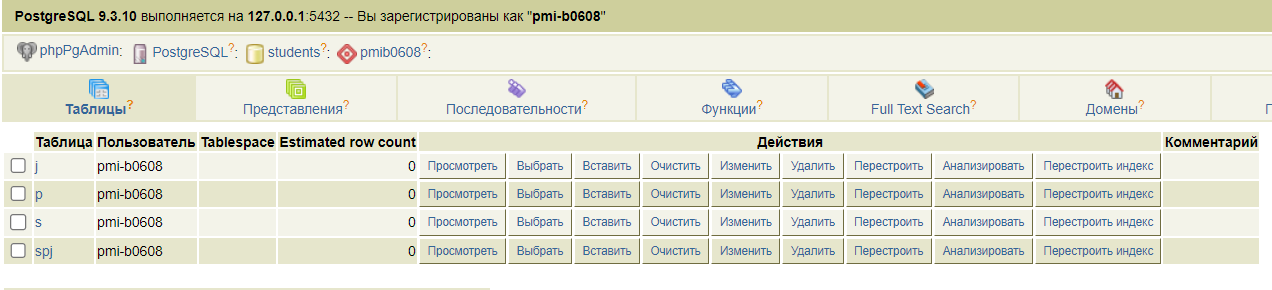
n\_post character(6) NOT NULL,

n\_det character(6) NOT NULL,

n\_izd character(6) NOT NULL,

kol integer) WITH OIDS;

Результат:



1. Записать и выполнить совокупность запросов для занесения нижеприведенных данных в созданные таблицы: insert into имя\_таблицы [(поле [,поле]...)] values (константа [,константа]...)

Insert into s values ('S1','Смит ',20,'Лондон'),

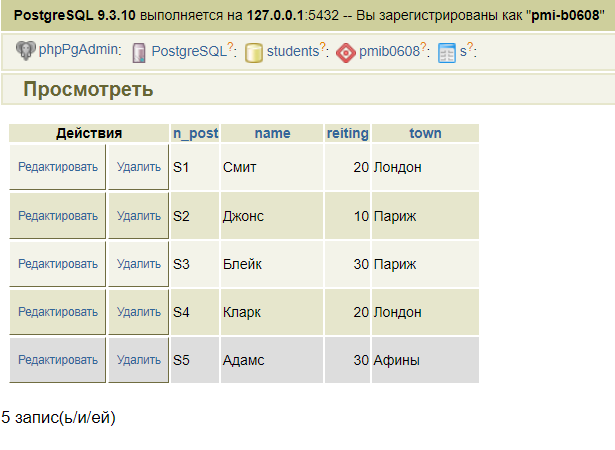
('S2','Джонс',10,'Париж '),

('S3','Блейк',30,'Париж '),

('S4','Кларк',20,'Лондон'),

('S5','Адамс',30,'Афины ');

Результат:



Insert into p values ('P1','Гайка','Красный',12,'Лондон'),

('P2','Болт','Зеленый',17,'Париж '),

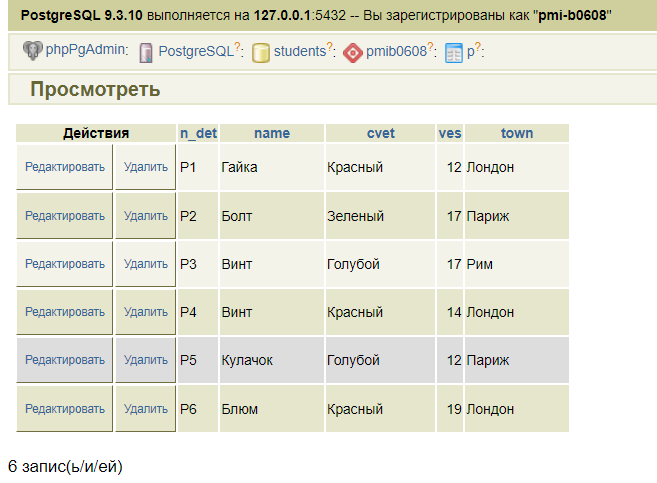
('P3','Винт','Голубой',17,'Рим'),

('P4','Винт','Красный',14,'Лондон'),

('P5','Кулачок','Голубой',12,'Париж '),

('P6','Блюм','Красный',19,'Лондон');

Результат:



Insert into j values ('J1','Жесткий диск', 'Париж'),

('J2','Перфоратор', 'Рим'),

('J3','Считыватель','Афины'),

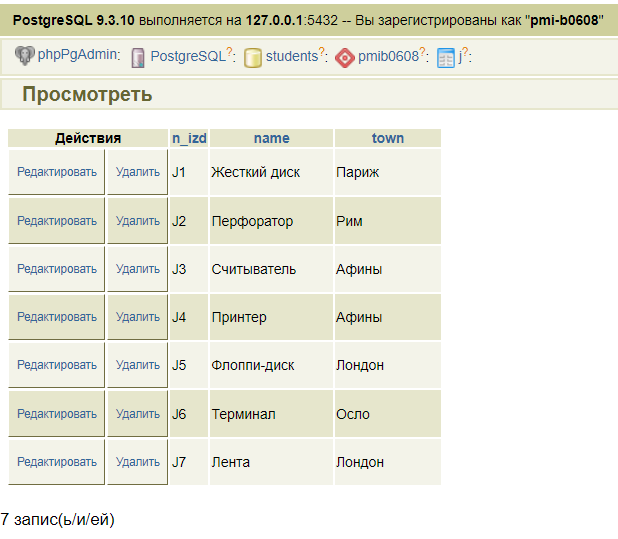
('J4','Принтер','Афины'),

('J5','Флоппи-диск','Лондон'),

('J6','Терминал','Осло'),

('J7','Лента','Лондон');

Результат:



insert into spj 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),

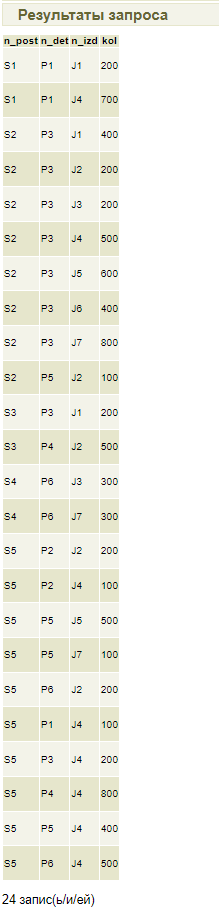
('S5', 'P3', 'J4', 200),

('S5', 'P4', 'J4', 800),

('S5', 'P5', 'J4', 400),

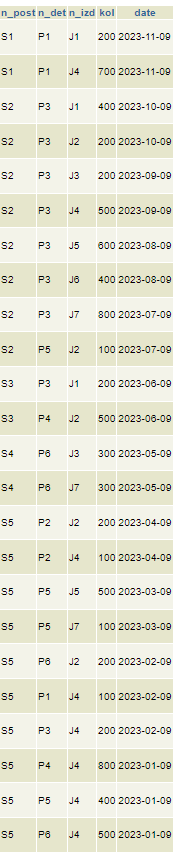
('S5', 'P6', 'J4', 500);

Результат:



1. 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.

Результат:

****

**ЧАСТЬ 2**

Цель работы: Используя данные таблиц, созданных подготовленной в первой лабораторной работе, подготовить и реализовать серию запросов, связанных с выборкой информации и модификацией данных таблиц.

1. Выбрать изделия, для которых поставщик с рейтингом 20 поставлял детали, поставлявшиеся для изделия J2.

**select j.name**

**from**

**spj join**

**(select distinct n\_det AS p\_j2**

**from spj**

**where n\_izd = 'J2') AS CP**

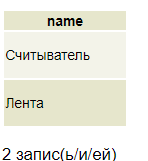
**ON spj.n\_det = CP.p\_j2**

**join j ON j.n\_izd = spj.n\_izd**

**join s ON s.n\_post = spj.n\_post**

**where s.reiting = 20**

Результат:



1. Найти изделия, для которых детали с весом от 17 до 19 поставлялись поставщиком с рейтингом больше 20. Вывести полную информацию об изделиях: номер, название, город.

**select j.n\_izd, j.name, j.town**

**from j**

**inner join spj on j.n\_izd = spj.n\_izd**

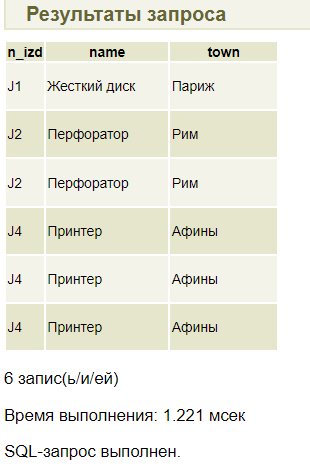
**inner join p on spj.n\_det = p.n\_det**

**inner join s on spj.n\_post = s.n\_post**

**where p.ves between 17 and 19**

**and s.reiting > 20;**

Результат:



1. Получить список поставщиков, выполнивших поставки ТОЛЬКО для изделий с красными деталями.

**select distinct s.n\_post, s.name**

**from s**

**join spj on s.n\_post = spj.n\_post**

**join p on spj.n\_det = p.n\_det**

**left join (**

**select distinct spj.n\_post, p.cvet**

**from spj**

**join p on spj.n\_det = p.n\_det**

**where p.cvet != 'Красный'**

**) as NonRed on s.n\_post = NonRed.n\_post**

**where NonRed.n\_post is NULL;**

Результат:



4) Вывести полный список изделий и для каждого изделия определить, из скольких разных городов для него поставлялись детали с весом 12. Изделия в списке должны быть ВСЕ. Список должен быть упорядочен по номеру изделия.

**select j.n\_izd, j.name, count(distinct p.town)**

**from j**

**left join spj on j.n\_izd = spj.n\_izd**

**left join p on spj.n\_det = p.n\_det and p.ves = 12**

**group by j.n\_izd, j.name**

**order by j.n\_izd;**

Результат:

