|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Технологии баз данных» | | |
| **Создание и модификация баз данных и таблиц. Запросы к базе данных** | | |
|  | | |
|  | Бригада 10 | Субботин дмитрий |
| Группа ПМИ-12 | Швадченко артем |
| Вариант 6 |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Стасышина Татьяна леонидовна СИВАк мария алексеевна |
|  |  |
| Новосибирск, 2024 | | |

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

1. **Ход работы**

**Задание:**

* **Используя набор команд языка SQL:**

1. Создать четыре таблицы
2. 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).

1. Проверить результат заполнения таблиц, написав и выполнив простейший запрос select \* from имя\_таблицы
2. Средствами системы меню программы phpPgAdmin выполнить модификацию структуры таблицы SPJ, добавив поле с датой поставки. Занести произвольные даты поставки, использую меню программы phpPgAdmin.

* **Задания по выборке информации:**

1. Выбрать детали, поставлявшиеся для изделий, для которых поставщик S3 поставлял детали из Рима.

2. Найти поставщиков, которые поставляли детали с весом не меньше 17 для изделий из Рима. Вывести номер поставщика, фамилию, рейтинг и город, где он проживает.

3. Получить список деталей, которые поставлялись ТОЛЬКО поставщиками, выполнившие поставки для изделия J7.

4. Вывести полный список городов и для каждого города найти общее число поставок, выполненных из этого города. Города в списке должны быть ВСЕ. Список должен быть упорядочен по алфавиту.

* **Задания по модификации информации:**

1. Построить таблицу с упорядоченным списком городов таких, что в городе производится какая-либо деталь или собирается какое-либо изделие, но не размещается ни один поставщик.

2. Каждое изделие с длиной названия > 10 перевести в город, в котором проживает первый по алфавиту поставщик деталей для этого изделия.

1. **Описание работы программы:**
2. Создаем базу данных в putty командой **new\_schema**pmib1610**students.**
3. Отдельное создание таблиц баз данных через меню phpPgAdmin

* Таблицу j и spj создаем с помощью скрипта
* Таблицу s и p создаем с помощью меню, используя клавишу Create table

**Используемые команды:**

*create database* – создание базы данных;

*database* – выбор существующей базы данных;

*create table* – создание таблицы базы данных;

*select* – осуществление запроса по выборке информации из таблиц базы данных;

*insert* – добавление одной или нескольких строк в таблицу;

*update* – модификация одной или нескольких строк таблицы;

*from* – описание таблицы, из которой следует выбирать данные (обязательная секция);

*join* – описание дополнительной таблицы, из которой следует выбирать данные, и условия ее соединения с ранее описанными таблицами;

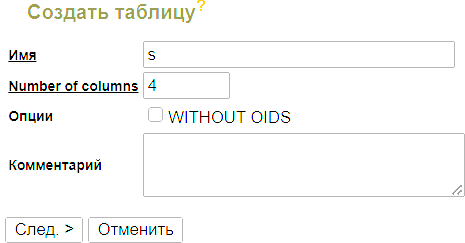
*where* – описание условий отбора строк (записей);

*group by* – описание критерия формирования групп строк (группой называется множество строк, имеющих одинаковые значения в указанных столбцах, для каждой группы возвращается одна строка результата);

*order by* – сортировка результата выполнения запроса по одному или нескольким столбцам;

**Результат работы меню программы и SQL-скриптов**

*Создание таблиц s и p:*

**

*Изображение выглядит как текст, число, Шрифт, линия

Автоматически созданное описание*

*p создается по аналогии с s*

**Текст sql-скрипта:**

CREATE TABLE j(

n\_izd character(6) NOT NULL UNIQUE,

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;

INSERT INTO j VALUES ('J1','Жесткий диск','Париж'),

('J2','Перфоратор','Рим'),

('J3','Cчитыватель','Афины'),

('J4','Принтер','Афины'),

('J5','Флоппи-диск','Лондон'),

('J6','Терминал','Осло'),

('J7','Лента','Лондон');

INSERT INTO s VALUES ('S1','Смит ',20,'Лондон'),

('S2','Джонс',10,'Париж'),

('S3','Блейк', 30,'Париж'),

('S4','Кларк', 20,'Лондон'),

('S5','Адамс', 30,'Атенс');

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');

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);

*Конфигурация таблицы j(Изделия):*



*Конфигурация таблицы p(Изделия):*

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Конфигурация таблицы s(Изделия):*

Изображение выглядит как текст, снимок экрана, Шрифт, линия

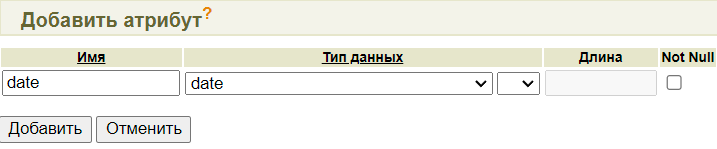
Автоматически созданное описание

*Конфигурация таблицы spj(Изделия):*

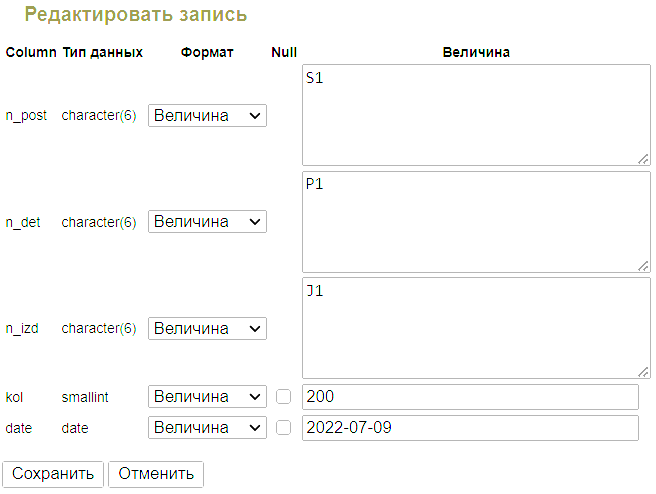
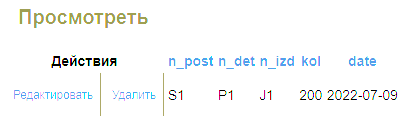
Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Модификация структуры таблицы SPJ с добавлением поля с датой поставки:*


*Занос произвольной даты поставки, используя меню программы phpPgAdmin:*

** **

*Проверка результата заполнения таблиц:*

*Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание*

1. **Задания по выборке информации**

**Вариант 6**

1. Выбрать детали, поставлявшиеся для изделий, для которых поставщик S3 поставлял детали из Рима.

select distinct n\_det

from spj

where n\_izd in

(select spj.n\_izd

from spj

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

where spj.n\_post='S3'

and p.town='Рим')

order by spj.n\_det



1. Найти поставщиков, которые поставляли детали с весом не меньше 17 для изделий из Рима. Вывести номер поставщика, фамилию, рейтинг и город, где он проживает.

select distinct s.n\_post,

s.name,

s.reiting,

s.town

from s

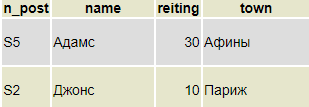
join spj on s.n\_post=spj.n\_post

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

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

where j.town='Рим'

and p.ves>=17



1. Получить список деталей, которые поставлялись ТОЛЬКО поставщиками, выполнившие поставки для изделия J7.

select distinct n\_det

from spj

where n\_post in

(select n\_post

from spj

where n\_izd='J7')

except

select distinct n\_det

from spj

where n\_post not in

(select n\_post

from spj

where n\_izd='J7')

order by n\_det



1. Вывести полный список городов и для каждого города найти общее число поставок, выполненных из этого города. Города в списке должны быть ВСЕ. Список должен быть упорядочен по алфавиту.

select p.town,

count(spj.kol)

from p

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

group by town

order by town

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Задания по модификации информации**

**Вариант 6**

1. Построить таблицу с упорядоченным списком городов таких, что в городе производится какая-либо деталь или собирается какое-либо изделие, но не размещается ни один поставщик.

create table town as

(select distinct town

from p

union

select distinct town

from j

except

select distinct town

from s

order by town)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

1. Каждое изделие с длиной названия > 10 перевести в город, в котором проживает первый по алфавиту поставщик деталей для этого изделия.

update j

set town =

(select s.town

from s

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

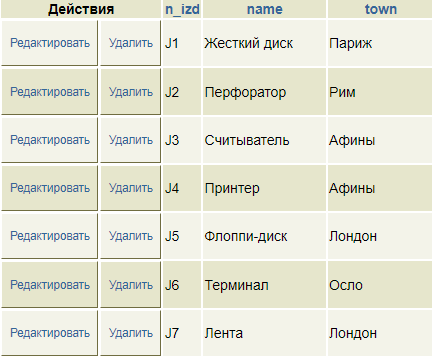
where spj.n\_izd = j.n\_izd

order by s.name asc,

s.town desc

limit 1)

where length(j.name) > 10

 Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание