|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования  «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Базы данных» | | |
|  | | |
|  | | |
|  | Бригада 7 | Барсукова наталья |
| Группа ПМ-91 | затолоцкая юлия |
| Вариант 2 |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Стасышина Татьяна Леонидовна |
|  | Сивак Мария Алексеевна |
| Новосибирск, 2022 | | |

Вариант 2

Часть 1. СОЗДАНИЕ И МОДИФИКАЦИЯ ТАБЛИЦ БАЗЫ ДАННЫХ

Цель задания

Создать схему базы данных, ознакомиться с возможностями интерактивной

программы phpPgAdmin, создать с ее помощью набор таблиц в ней и заполнить

таблицы данными для последующей работы.

Выполнение задания

1. **Ознакомиться с правилами работы и структурой меню интерактивной программы phpPgAdmin.**

Ознакомлены.

1. **Изучить набор команд языка SQL, связанный с созданием базы данных, созданием, модификацией структуры таблиц и их удалением, вставкой, модификацией и удалением записей таблиц (Приложение 2):**

Изучено.

1. **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).**

С помощью putty заходим на сервер students.ami.nstu.ru и создаём схему базы данных.

Схема создана:

****

1. **Используя программу phpPgAdmin в созданной схеме создать четыре таблицы.**

**При создании таблиц выполнить следующие условия:**

**- поля номер\_поставщика, номер\_детали, номер\_изделия во всех таблицах имеет символьный тип и длину 6;**

**- поля рейтинг, вес и количество имеют целочисленный тип;**

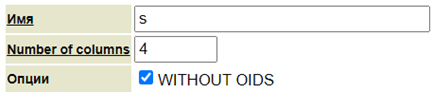
**- поля фамилия, город (поставщика, детали или изделия), название (детали или изделия) имеют символьный тип и длину 20;**

**- ни для одного поля не предусматривается использование индексов;**

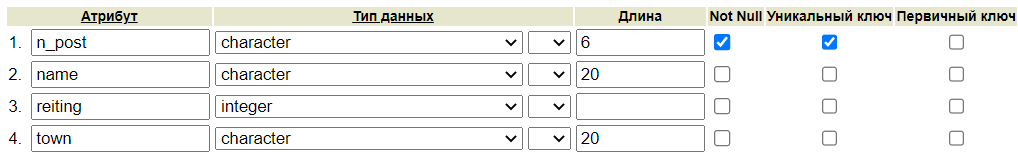
**- для всех полей допускаются значения NULL и значения-дубликаты, кроме поля номер\_поставщика из таблицы S, номер детали из таблицы P, номер изделия из таблицы J.**

1. **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):**

**Таблица S:**



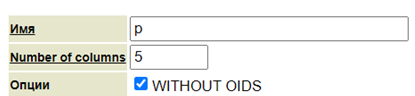
Поля таблицы S:



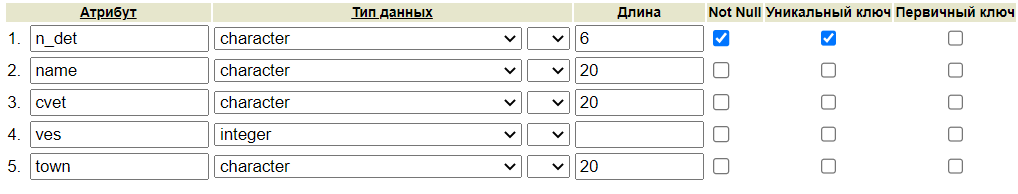
*Поле n\_post не может принимать значение Null и иметь дубликаты*



**Таблица P:**

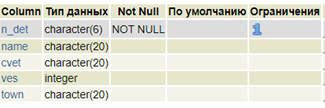


**Поля таблицы P:**



*Поле n\_det не может принимать значение Null и иметь дубликаты*

Результат:



**Таблица J:**

Скрипт для создания:

CREATE TABLE j(

n\_izd character(6) NOT NULL UNIQUE,

name character(20),

town character(20)

);

Результат:



**Таблица SPJ:**

Скрипт для создания:

CREATE TABLE spj(

n\_post character(6) NOT NULL,

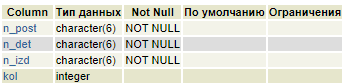
n\_det character(6) NOT NULL,

n\_izd character(6) NOT NULL,

kol integer

) WITH OIDS;

Результат:



1. **Записать и выполнить совокупность запросов для занесения вышеприведенных данных в созданные таблицы**

Скрипт, который заносит данные в таблицы:

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

('P1', 'Гайка', 'Красный', 12, 'Лондон'),

('P2', 'Болт', 'Зеленый', 17, 'Париж'),

('P3', 'Винт', 'Голубой', 17, 'Париж'),

('P4', 'Винт', 'Красный', 14, 'Лондон'),

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

('P6', 'Блюм', 'Красный', 19, 'Афины');

INSERT INTO j (n\_izd, name, town) VALUES

('J1', 'Жесткий диск', 'Париж'),

('J2', 'Перфоратор', 'Рим'),

('J3', 'Считыватель', 'Афины'),

('J4', 'Принтер', 'Афины'),

('J5', 'Флоппи-диск', 'Лондон'),

('J6', 'Терминал', 'Осло'),

('J7', 'Лента', '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);

Запрос выполнен успешно.

1. **Проверить результат заполнения таблиц, написав и выполнив простейший**

**запрос:**

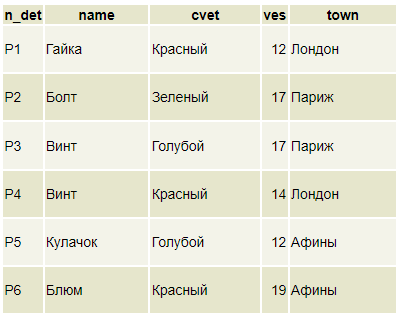
SELECT \* FROM s

Результат:



SELECT \* FROM p

Результат:



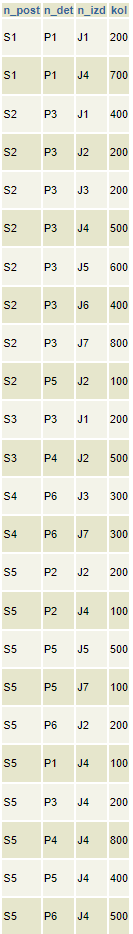
SELECT \* FROM j

Результат:



SELECT \* FROM spj

Результат:

Данные введены верно.

1. **Средствами системы меню программы phpPgAdmin выполнить модификацию структуры таблицы SPJ, добавив в нее поле с датой поставки. Убедиться в успешности выполненных действий. При необходимости исправить ошибки:**

Выбираем “Добавить атрибут”:



****

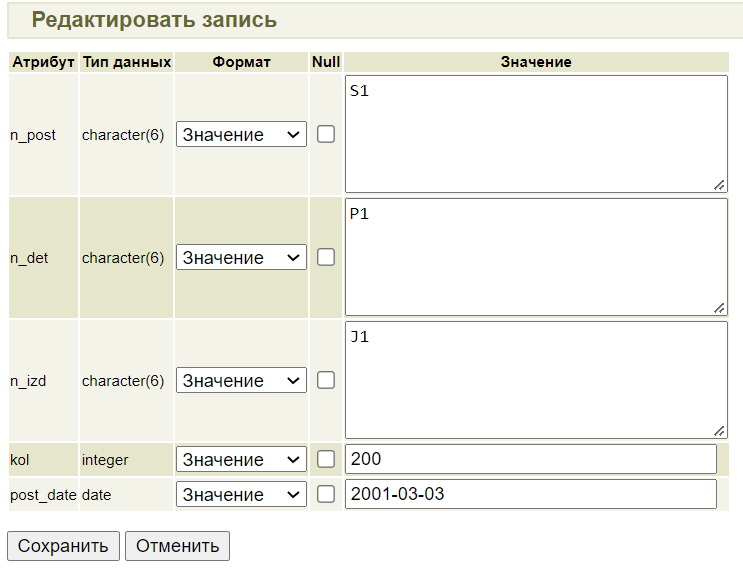
Теперь можем добавить даты поставки:

Выбираем “Посмотреть”

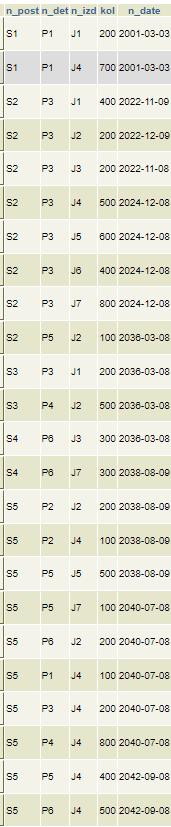
****

А теперь редактируем записи:

****

****

**Результат:**

****

Часть 2. ВЫБОР И МОДИФИКАЦИЯ ДАННЫХ ТАБЛИЦ

Цель задания

Используя данные таблиц созданной схемы базы данных, подготовленной в первой

лабораторной работе, подготовить и реализовать серию запросов, связанных с

выборкой информации и модификацией данных таблиц.

Выполнение задания

1. **Изучить набор команд языка SQL, связанный с созданием запросов, добавлением, модификацией и удалением строк таблицы:**

Изучили.

1. **Изучить состав, правила и порядок использования ключевых фраз оператора**

Изучили.

1. **Подготовить и выполнить средствами программы phpPgAdmin четыре запроса по выборке информации из таблиц схемы базы данных для решения нижеприведенных задач:**

1. Выбрать поставщиков, поставляющих детали, поставляемые для изделий из Осло

* Получить список изделий из Осло
* Получить информацию о деталях
* Получить поставщиков, поставляющих детали, поставляемые для изделий из Осло

**Запрос:**

SELECT DISTINCT spj.n\_post

FROM spj

JOIN p ON p.n\_det = spj.n\_det

WHERE spj.n\_izd IN (

SELECT n\_izd

FROM j

WHERE town='Осло'

)

**Результат:**



2. Найти изделия такие, что детали для изделия поставлялись из того же города, где находится поставщик, выполнивший поставку. Вывести полную информацию об изделиях: номер, название, город

* Получить информацию о поставщиках, деталях и изделиях
* Найти поставки, где города поставщиков и деталей пересекаются

**Запрос:**

SELECT j.n\_izd, j.name, j.town

FROM spj

JOIN p ON p.n\_det = spj.n\_det

JOIN j ON j.n\_izd = spj.n\_izd

JOIN s ON s.n\_post = spj.n\_post

WHERE s.town = p.town

**Результат:**



3. Получить список поставщиков, которые поставляли ТОЛЬКО детали, в названии которых присутствует буква ‘к’:

* Найти поставщиков, которые выполнили поставки для деталей, в названии которых присутствует буква ‘к’:
* Исключить их из списка поставщиков.

**Запрос:**

CREATE TABLE temp (town character(20));

INSERT INTO temp (SELECT town

FROM p

INTERSECT

(SELECT town

FROM j

INTERSECT

(SELECT town

FROM s )))

ORDER BY town ASC

**Результат:**



4. Вывести полный список поставщиков и для каждого поставщика найти суммарное количество деталей с весом больше 17, которые были им поставлены. Поставщики в списке должны быть ВСЕ. Список должен быть упорядочен по номеру поставщика.

* Получить детали с весом>17
* Получить всех поставщиков
* Найти поставщиков и общее количество деталей, но считать только детали, с весом> 17

**Запрос:**

SELECT DISTINCT post.n\_post, pt.sves

FROM (SELECT n\_post

FROM spj) post

LEFT JOIN (SELECT spj.n\_post, SUM(spj.kol) sves

FROM spj

JOIN p ON spj.n\_det=p.n\_det

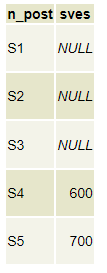
WHERE p.ves>17

GROUP BY spj.n\_post) pt

ON pt.n\_post=post.n\_post

ORDER BY 1

**Результат:**



**4. Подготовить и выполнить средствами программы phpPgAdmin два запроса по**

**модификации информации (вставка, удаление, замещение) из таблиц схемы базы**

**данных для решения нижеприведенных задач. При этом в тех заданиях, где речь**

**идет о создании таблиц, предполагается формирование постоянной таблицы базы данных:**

1. Построить таблицу с упорядоченным списком городов таких, что в городе есть и поставщик, и деталь, и изделие.

CREATE TABLE temp (town character(20));

INSERT INTO temp (SELECT town

FROM p

INTERSECT

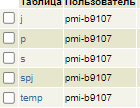
(SELECT town

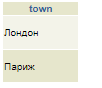
FROM j

INTERSECT

(SELECT town

FROM s )))

ORDER BY town ASC



2. Для всех поставщиков, имеющих в настоящее время наименьший рейтинг, установить рейтинг равным числу изделий, для которых поставщик выполнил поставки, умноженному на 5.

UPDATE s SET reiting=(

SELECT COUNT(DISTINCT n\_izd)\*5

FROM spj

WHERE n\_post = s.n\_post)

WHERE reiting=(

SELECT MIN(reiting) FROM s

)

|  |  |
| --- | --- |
|  |  |

Изменена 1 строка.

**5. Защитить лабораторную работу, ответив на контрольные вопросы.**