|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Теоретической и прикладной математики | | |
|  | | |
| Лабораторная работа № 1,2 | | |
| по дисциплине «Технологии баз данных» | | |
|  | | |
|  | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-02 |
| Вариант: | 6 |
| Студент: | Сидоров Даниил, |
|  | Дюков Богдан |
| Преподаватель: | Стасышина Т.Л., |
|  | Сивак М.А. |
|
|  |  |
| Новосибирск | | |
| 2023 | | |

**Часть 1**

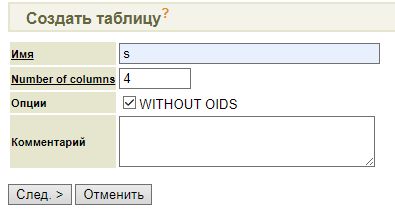
1. Ознакомились с интерфейсом и возможностями программы phpPgAdmin.
2. Изучили набор команд языка SQL, связанный с созданием базы данных, созданием, модификацией структуры таблиц и их удалением, вставкой, модификацией и удалением записей таблиц.
3. Из командной строки в существующей базе данных students командой:

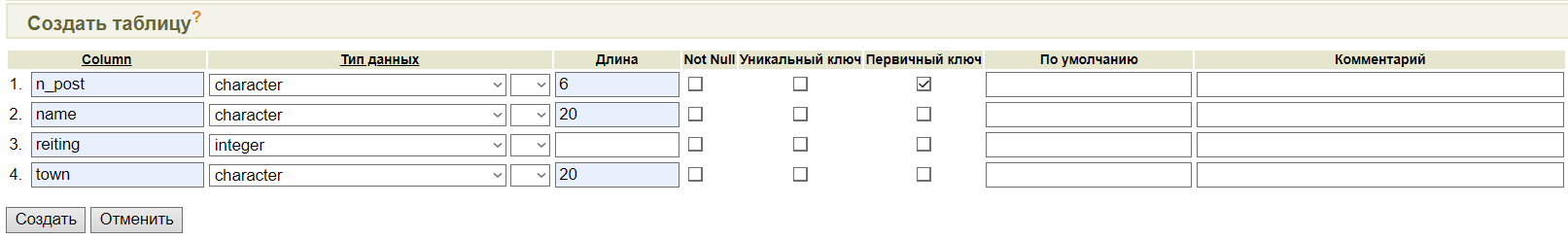
*new\_schema pmib0704 students*

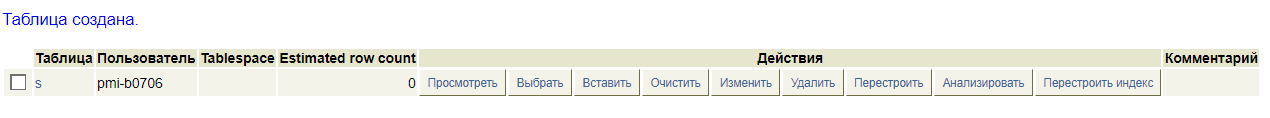
создали схему базы данных.

1. Используя программу phpPgAdmin, в созданной схеме создали четыре таблицы.

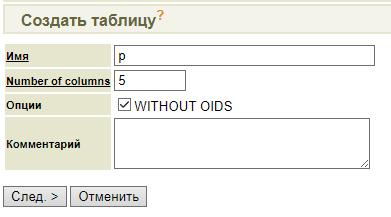
* Таблица s:

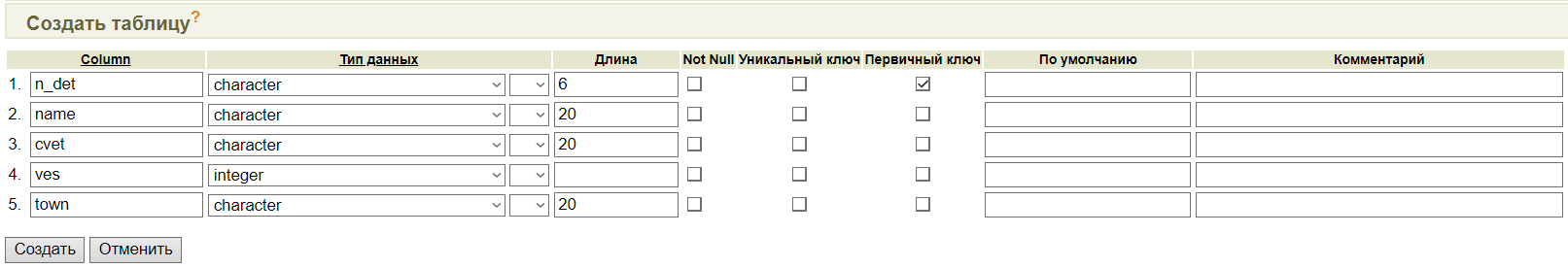


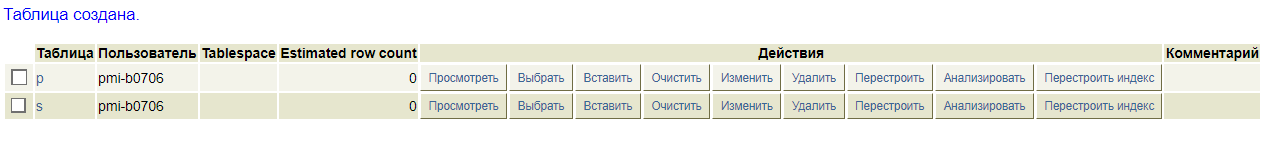




* Таблица p:







* Таблица j:

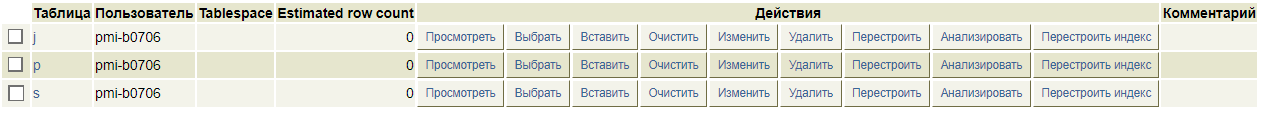
CREATE TABLE j (

n\_izd CHARACTER(6) PRIMARY KEY,

name CHARACTER(20),

town CHARACTER(20)

);



* Таблица spj:

CREATE TABLE spj (

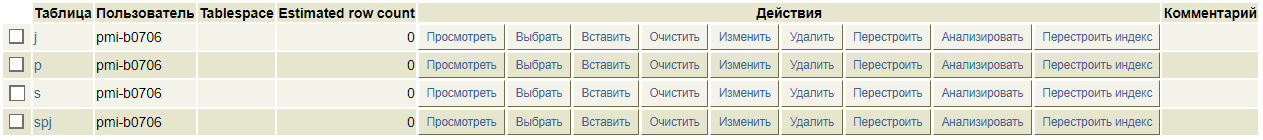
n\_post CHARACTER(6) NOT NULL,

n\_det CHARACTER(6) NOT NULL,

n\_izd CHARACTER(6) NOT NULL,

kol INTEGER

) WITH OIDS;



1. Записали и выполнили совокупность запросов для занесения данных в созданные таблицы.

* Таблица s:

**Запрос:**

INSERT INTO s

VALUES ('S1', 'Cмит', 20, 'Лондон'),

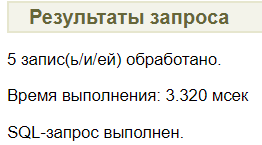
('S2', 'Джонс', 10, 'Париж'),

('S3', 'Блейк', 30, 'Париж'),

('S4', 'Кларк', 20, 'Лондон'),

('S5', 'Адамс', 30, 'Афины');

**Результат:**





* Таблица p:

**Запрос:**

INSERT INTO p

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

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

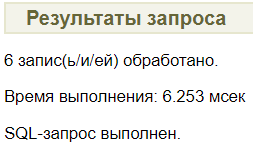
('P3', 'Винт', 'Голубой', 17, 'Рим'),

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

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

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

**Результат:**





* Таблица j:

**Запрос:**

INSERT INTO j

VALUES ('J1', 'Жесткий диск', 'Париж'),

('J2', 'Перфоратор', 'Рим'),

('J3', 'Cчитыватель', 'Афины'),

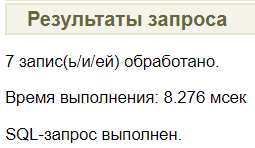
('J4', 'Принтер', 'Афины'),

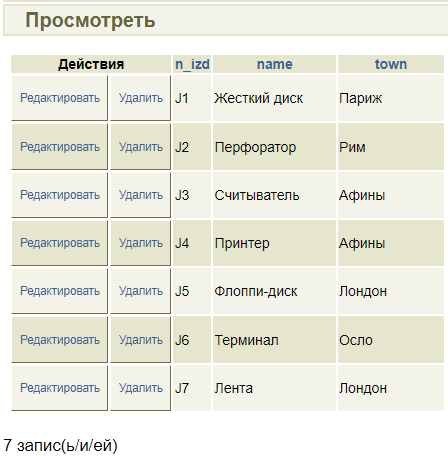
('J5', 'Флоппи-диск', 'Лондон'),

('J6', 'Терминал', 'Осло'),

('J7', 'Лента', 'Лондон');

**Результат:**





* Таблица spj:

**Запрос:**

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),

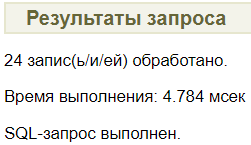
('S5', 'P3', 'J4', 200),

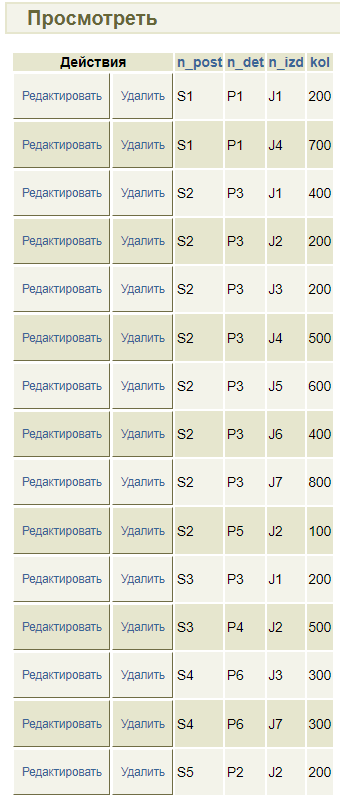
('S5', 'P4', 'J4', 800),

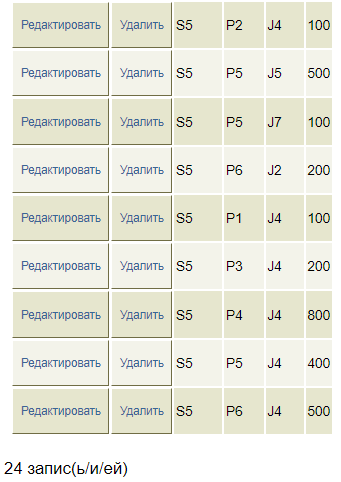
('S5', 'P5', 'J4', 400),

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

**Результат:**







1. Проверили результат заполнения таблиц, написав и выполнив простейший запрос select \* from имя\_таблицы.

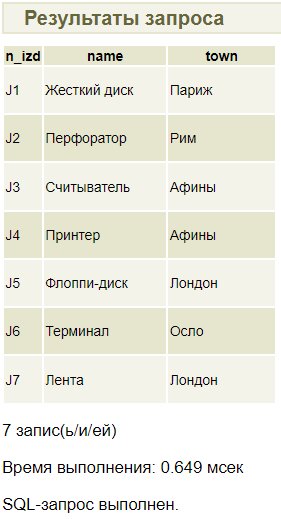
SELECT \* FROM s;



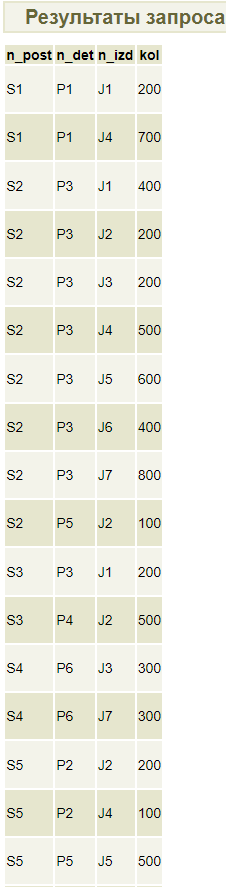
SELECT \* FROM p;

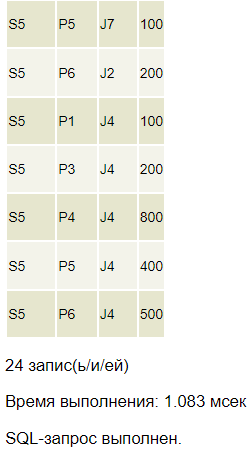


SELECT \* FROM j;

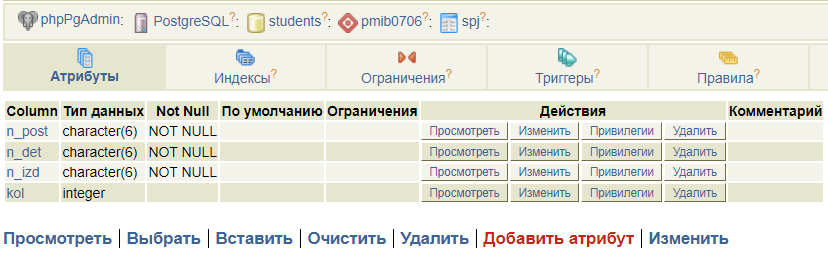


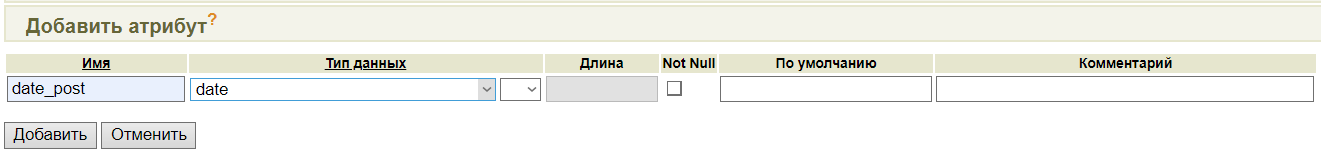
SELECT \* FROM spj;





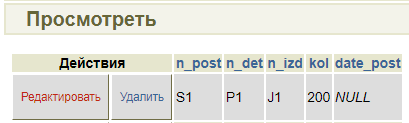
1. Средствами системы меню программы phpPgAdmin выполнили модификацию структуры таблицы SPJ, добавив поле с датой поставки. Убедились в успешности выполненных действий. Занесли произвольные даты поставки, используя меню программы phpPgAdmin.

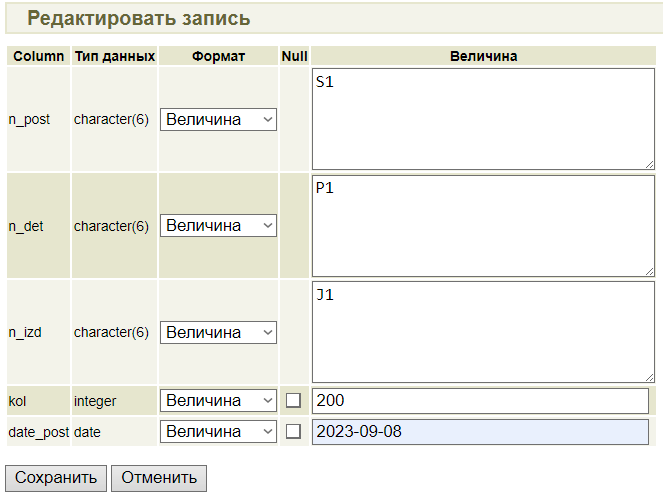


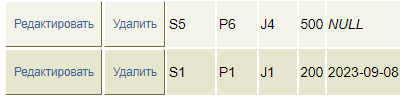




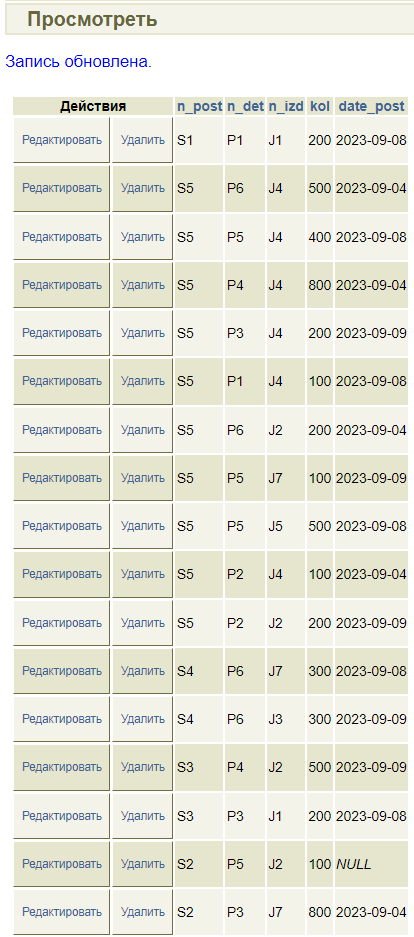
Нажимая “Посмотреть”, мы можем редактировать каждую запись в таблице:

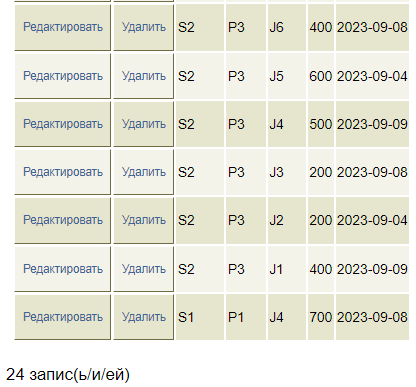






В результате занесения произвольных дат поставки:





**Часть 2**

**Выборка информации**

**Задание 1**. Выбрать поставщиков, поставляющих детали, поставляемые для

изделий из Осло.

**Запрос**:

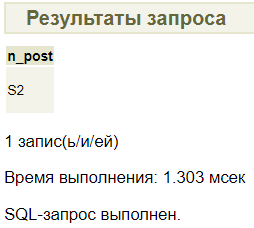
SELECT DISTINCT n\_post

FROM spj

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

WHERE j.town='Осло';

**Результат:**



**Задание 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.

**Запрос**:

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

FROM spj

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

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

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

WHERE s.town=p.town;

**Результат:**



**Задание 3**. Получить список поставщиков, которые поставляли ТОЛЬКО

детали, в названии которых присутствует буква ‘к’.

**Запрос**:

SELECT DISTINCT n\_post

FROM spj

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

WHERE p.name LIKE '%к%';

**Результат:**



**Задание 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.

**Запрос**:

SELECT s.n\_post AS n\_post,

COALESCE(SUM(kol), 0) AS total\_det\_number

FROM p

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

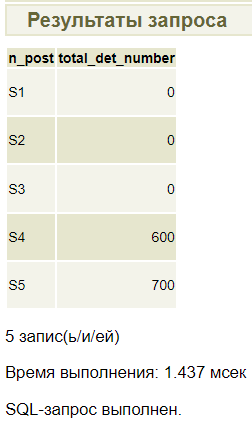
AND p.ves>17

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

GROUP BY s.n\_post

ORDER BY s.n\_post;

**Результат:**

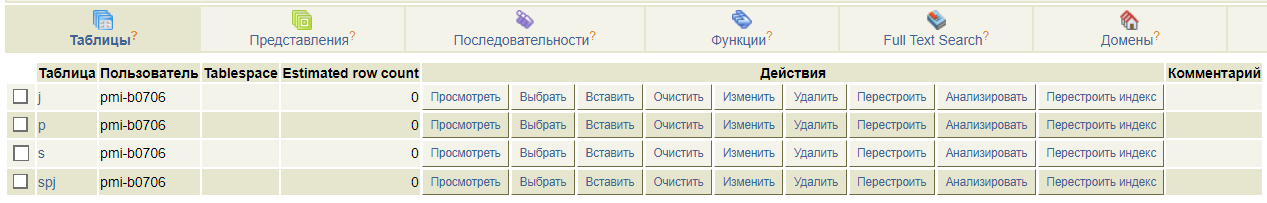


**Модификация информации**

**Задание 1**. Построить таблицу с упорядоченным списком городов таких, что

в городе есть и поставщик, и деталь, и изделие.

**Состояние до:**



**Запрос**:

CREATE TABLE cities\_with\_everyone (town character(6));

INSERT INTO cities\_with\_everyone (SELECT DISTINCT town

FROM p

WHERE p.town in(SELECT town

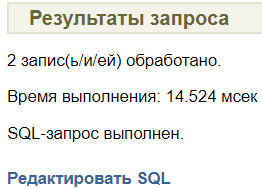
FROM s

WHERE s.town in (SELECT town

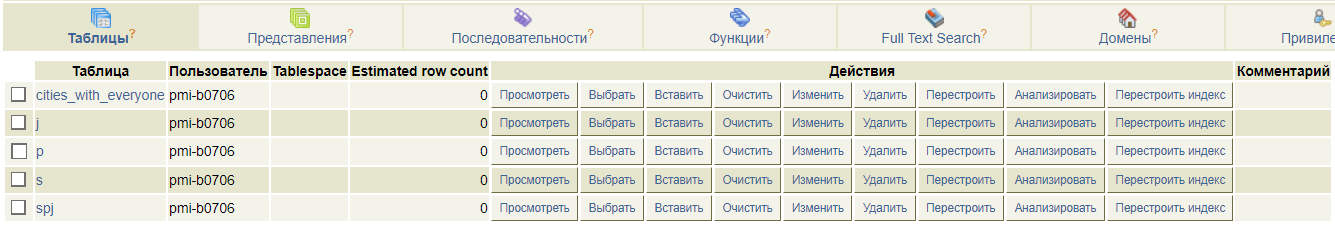
FROM j))

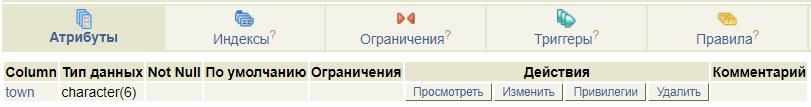
ORDER BY town);

**Результат:**



**Состояние после:**







**Задание 2**. Для всех поставщиков, имеющих в настоящее время наименьший рейтинг, установить рейтинг равным числу изделий, для которых

поставщик выполнил поставки, умноженному на 5.

**Состояние до:**



**Запрос:**

UPDATE s SET reiting=(SELECT COUNT(n\_det)

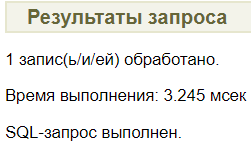
FROM spj

WHERE s.n\_post=spj.n\_post)\*5

WHERE s.reiting=(SELECT MIN(reiting)

FROM s);

**Результат:**



**Состояние после:**

