



МИНИСТЕРСТВО НАУКИ  
И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Новосибирский государственный технический университет»



**НГТУ  
НЭТИ** | **Факультет прикладной  
математики и информатики**

Кафедра прикладной математики и информатики

Лабораторная работа 1  
по дисциплине «Технологии баз данных»

**Создание и модификация баз данных и таблиц. Запросы к базе  
данных**

Группа	ПМ-92
Бригада	pmi-b9209
Вариант	4
Студенты	ИВАНОВ ВЛАДИСЛАВ КУТУЗОВ ИВАН
Преподаватели	СТАСЫШИНА Т. Л. СИВАК М. А.

Новосибирск, 2022

# 1 Создание и модификация таблиц

## 1.1 Создание таблиц

Используя программу phpPgAdmin, в созданной схеме создать четыре таблицы. При создании таблиц предусмотреть выполнение следующих условий:

1. поля номер поставщика, номер детали, номер изделия во всех таблицах имеет символьный тип и длину 6
2. поля рейтинг, вес и количество имеют целочисленный тип
3. поля фамилия, город (поставщика, детали или изделия), название (детали или изделия) имеют символьный тип и длину 20
4. ни для одного поля не предусматривается использование индексов
5. 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).

Создадим таблицы с помощью запроса:

```
1 drop table if exists j cascade;
2
3 create table j (
4     n_izd character(6) primary key,
5     name character(20),
6     town character(20)
7 );
8
9
10 drop table if exists s cascade;
11
12 create table s (
13     n_post character(6) primary key,
14     name character(20),
15     reiting integer,
16     town character(20)
17 );
18
19
20 drop table if exists p cascade;
21
22 create table p (
23     n_det character(6) primary key,
24     name character(20),
25     cvet character(20),
26     ves integer,
27     town character(20)
28 );
29
30
31 drop table if exists spj cascade;
32
33 create table spj (
34     n_post character(6),
35     n_det character(6),
36     n_izd character(6),
37     kol integer
38 ) with oids;
```

Заносим данные в созданные таблицы:

```
1 insert into s values
2   ('S1', 'Смит', 20, 'Лондон'),
3   ('S2', 'Джонс', 10, 'Париж'),
4   ('S3', 'Блейк', 30, 'Париж'),
5   ('S4', 'Кларк', 20, 'Лондон'),
6   ('S5', 'Адамс', 30, 'Афины');
7
8 insert into p values
9   ('P1', 'Гайка', 'Красный', 12, 'Лондон'),
10  ('P2', 'Болт', 'Зеленый', 17, 'Париж'),
11  ('P3', 'Винт', 'Голубой', 17, 'Рим'),
12  ('P4', 'Винт', 'Красный', 14, 'Лондон'),
13  ('P5', 'Кулачок', 'Голубой', 12, 'Париж'),
14  ('P6', 'Блюм', 'Красный', 19, 'Лондон');
15
16 insert into j values
17  ('J1', 'Жесткий диск', 'Париж'),
18  ('J2', 'Перфоратор', 'Рим'),
19  ('J3', 'Считыватель', 'Афины'),
20  ('J4', 'Принтер', 'Афины'),
21  ('J5', 'Флоппидиск-', 'Лондон'),
22  ('J6', 'Терминал', 'Осло'),
23  ('J7', 'Лента', 'Лондон');
24
25 insert into spj values
26  ('S1', 'P1', 'J1', 200),
27  ('S1', 'P1', 'J4', 700),
28  ('S2', 'P3', 'J1', 400),
29  ('S2', 'P3', 'J2', 200),
30  ('S2', 'P3', 'J3', 200),
31  ('S2', 'P3', 'J4', 500),
32  ('S2', 'P3', 'J5', 600),
33  ('S2', 'P3', 'J6', 400),
34  ('S2', 'P3', 'J7', 800),
35  ('S2', 'P5', 'J2', 100),
36  ('S3', 'P3', 'J1', 200),
37  ('S3', 'P4', 'J2', 500),
38  ('S4', 'P6', 'J3', 300),
39  ('S4', 'P6', 'J7', 300),
40  ('S5', 'P2', 'J2', 200),
41  ('S5', 'P2', 'J4', 100),
42  ('S5', 'P5', 'J5', 500),
43  ('S5', 'P5', 'J7', 100),
44  ('S5', 'P6', 'J2', 200),
45  ('S5', 'P1', 'J4', 100),
46  ('S5', 'P3', 'J4', 200),
47  ('S5', 'P4', 'J4', 800),
48  ('S5', 'P5', 'J4', 400),
49  ('S5', 'P6', 'J4', 500);
```

Таблица S:

Actions		n_post	name	reiting	town
Edit	Delete	S1	Смит	20	Лондон
Edit	Delete	S2	Джонс	10	Париж
Edit	Delete	S3	Блейк	30	Париж
Edit	Delete	S4	Кларк	20	Лондон
Edit	Delete	S5	Адамс	30	Афины

Таблица P:

Actions		n_det	name	cvet	ves	town
Edit	Delete	P1	Гайка	Красный	12	Лондон
Edit	Delete	P2	Болт	Зеленый	17	Париж
Edit	Delete	P3	Винт	Голубой	17	Рим
Edit	Delete	P4	Винт	Красный	14	Лондон
Edit	Delete	P5	Кулачок	Голубой	12	Париж
Edit	Delete	P6	Блюм	Красный	19	Лондон

Таблица J:

Actions		n_izd	name	town
Edit	Delete	J1	Жесткий диск	Париж
Edit	Delete	J2	Перфоратор	Рим
Edit	Delete	J3	Считыватель	Афины
Edit	Delete	J4	Принтер	Афины
Edit	Delete	J5	Флоппи-диск	Лондон
Edit	Delete	J6	Терминал	Осло
Edit	Delete	J7	Лента	Лондон

Таблица SPJ:

Actions		n_post	n_det	n_izd	kol
Edit	Delete	S1	P1	J1	200
Edit	Delete	S1	P1	J4	700
Edit	Delete	S2	P3	J1	400
Edit	Delete	S2	P3	J2	200
Edit	Delete	S2	P3	J3	200

## 1.2 Модификация таблиц

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, используя меню программы phpPgAdmin.

Добавляем поле с датой поставки и заносим произвольные даты:

```
1 alter table spj
2 add column delivery_date date;
3
4
5 update spj
6 set delivery_date = '2022-09-12'
7 where n_post = 'S1';
8
9 update spj
10 set delivery_date = '2022-02-19'
11 where n_post = 'S2';
12
13 update spj
14 set delivery_date = '2022-03-11'
15 where n_post = 'S3';
16
17 update spj
18 set delivery_date = '2022-04-08'
19 where n_post = 'S4';
20
21 update spj
22 set delivery_date = '2022-07-21'
23 where n_post = 'S5';
```

Результат:

Actions		n_post	n_det	n_izd	kol	delivery_date
Edit	Delete	S1	P1	J1	200	2022-09-12
Edit	Delete	S1	P1	J4	700	2022-09-12
Edit	Delete	S2	P3	J1	400	2022-02-19
Edit	Delete	S2	P3	J2	200	2022-02-19
Edit	Delete	S2	P3	J3	200	2022-02-19
Edit	Delete	S2	P3	J4	500	2022-02-19
Edit	Delete	S2	P3	J5	600	2022-02-19

## 2 Запросы к базе данных

### 2.1 Задания по выборке информации

Выбрать изделия, для которых поставляли детали поставщики, поставлявшие зеленые детали:

```
1 select distinct n_izd
2 from spj
3 where n_post in (
4     select distinct n_post
5     from spj
6     join p on p.n_det = spj.n_det
7     where p.cvet = 'Зеленый'
8 )
```

Результат:

n_izd
J4
J5
J7
J2



Найти поставки такие, что поставщик, изделие и деталь размещены в одном и том же городе. Вывести номер поставщика, номер изделия, номер детали и город, где размещены изделие, поставщик и деталь:

```
1 select spj.n_post, spj.n_det, spj.n_izd, s.town
2 from spj
3 join p on p.n_det = spj.n_det
4 join j on j.n_izd = spj.n_izd
5 join s on s.n_post = spj.n_post
6 where s.town = p.town and s.town = j.town
```

Результат:

n_post	n_det	n_izd	town
S4	P6	J7	Лондон

Получить список деталей, поставленных ТОЛЬКО первым по алфавиту поставщиком:

```
1 select distinct n_det
2 from spj
3 join s on s.n_post = spj.n_post
4 where s.name = (
5     select name
6     from s
7     order by name
8     limit 1
9 )
10 except
11 select distinct n_det
12 from spj
13 join s on s.n_post = spj.n_post
14 where s.name != (
15     select name
16     from s
17     order by name
18     limit 1
19 )
```

Результат:

**n\_det**

P2

Вывести полный список деталей и для каждой детали определить, сколько поставщиков с рейтингом меньше 30 поставляли эту деталь. Детали в списке должны быть ВСЕ. Список должен быть упорядочен по номеру детали:

```
1 select temp_spj.n_det, count(rating_less_than_30.n_post)
2 from (
3     select n_det, n_post
4     from spj
5     group by n_det, n_post
6     ) as temp_spj
7 left join (
8     select n_post
9     from s
10    where reiting < 30
11    ) as rating_less_than_30 on rating_less_than_30.n_post = temp_spj.n_post
12 group by temp_spj.n_det
```

Результат:

n_det	count
P2	0
P6	1
P4	0
P3	1
P5	1
P1	1

## 2.2 Задания по модификации информации

Построить таблицу с упорядоченным списком городов таких, что в городе размещается 2 поставщика и собирается 2 изделия, но не производится ни одна деталь:

```
1 drop table if exists towns cascade;
2
3 create table towns (town character(20));
4 insert into towns (
5     select distinct town
6     from s
7     group by town
8     having count(town) = 1
9     intersect
10    select distinct town
11    from j
12    group by town
13    having count(town) = 2
14    except
15    select town
16    from
17        p
18    );
```

Результат:

	Table	Owner	Tablespace	Estimated row count	
<input type="checkbox"/>	j	pmi-b9209		0	
<input type="checkbox"/>	p	pmi-b9209		0	
<input type="checkbox"/>	s	pmi-b9209		0	
<input type="checkbox"/>	spj	pmi-b9209		0	town
<input type="checkbox"/>	towns	pmi-b9209		0	Афины

Всех поставщиков, имеющих в настоящее время наибольший рейтинг, перевести в город, откуда поставщик сделал наибольшее число поставок. Если таких городов больше одного, перевести в первый по алфавиту из этих городов:

```

1 update s set town = (
2     select p.town
3     from spj
4     join p on p.n_det = spj.n_det
5     where spj.n_post = s.n_post
6     group by spj.n_post, p.town
7     order by spj.count desc, lower(p.town)
8     limit 1
9 )
10 where s.reiting = (
11     select max(s.reiting)
12     from s
13 );

```

Результат:

Actions		n_post	name	reiting	town
Edit	Delete	S1	Смит	20	Лондон
Edit	Delete	S4	Кларк	20	Лондон
Edit	Delete	S2	Джонс	10	Париж
Edit	Delete	S3	Блейк	30	Лондон
Edit	Delete	S5	Адамс	30	Париж

(a) После

Actions		n_post	name	reiting	town
Edit	Delete	S1	Смит	20	Лондон
Edit	Delete	S2	Джонс	10	Париж
Edit	Delete	S3	Блейк	30	Париж
Edit	Delete	S4	Кларк	20	Лондон
Edit	Delete	S5	Адамс	30	Афины

(b) До