

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет информационных технологий и управления  
Кафедра интеллектуальных информационных технологий  
Дисциплина «Проектирование баз знаний»

**ОТЧЁТ**  
к лабораторной работе №1  
на тему  
**«МАТЕМАТИЧЕСКИЕ ОСНОВЫ РЕЛЯЦИОННЫХ ЯЗЫКОВ»**

БГУИР 6-05-0611-03 130

Выполнил студент группы 321701  
СЕМЕНЯКО Владимир Дмитриевич

---

(дата, подпись студента)

Проверил  
ЕРОФЕЕВ Илья Александрович

---

(дата, подпись преподавателя)

Минск 2025

# 1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант **22** Для выполнения лабораторной работы была выбрана СУБД PostgreSQL. Создаем таблицы, выполняя следующий запрос:

```
CREATE TABLE provider(  
    provider_code VARCHAR(255) PRIMARY KEY,  
    name VARCHAR(255),  
    status INT,  
    city VARCHAR(255)  
);  
  
CREATE TABLE detail(  
    detail_code VARCHAR(255) PRIMARY KEY,  
    name VARCHAR(255),  
    color VARCHAR(255),  
    size INT,  
    city VARCHAR(255)  
);  
  
CREATE TABLE project(  
    project_code VARCHAR(255) PRIMARY KEY,  
    name VARCHAR(255),  
    city VARCHAR(255)  
);  
  
CREATE TABLE number_of_details(  
    provider_code VARCHAR(255) REFERENCES provider(provider_code),  
    detail_code VARCHAR(255) REFERENCES detail(detail_code),  
    project_code VARCHAR(255) REFERENCES project(project_code),  
    number INT  
);  
  
INSERT INTO provider (provider_code, name, status, city)  
VALUES ('P1', 'Petrov', 20, 'Moscow'),  
       ('P2', 'Sinicin', 10, 'Tallin'),  
       ('P3', 'Federov', 30, 'Tallin'),  
       ('P4', 'Chaianov', 20, 'Minsk'),  
       ('P5', 'Krykov', 30, 'Kiev');  
  
INSERT INTO detail (detail_code, name, color, size, city)  
VALUES ('D1', 'Bolt', 'Red', 12, 'Moscow'),  
       ('D2', 'Gaika', 'Green', 17, 'Minsk'),  
       ('D3', 'Disk', 'Black', 17, 'Vilnus'),  
       ('D4', 'Disk', 'Black', 14, 'Moscow'),  
       ('D5', 'Korpus', 'Red', 12, 'Minsk'),  
       ('D6', 'Krishki', 'Red', 19, 'Moscow');  
  
INSERT INTO project (project_code, name, city)  
VALUES ('PR1', 'IPR1', 'Minsk'),  
       ('PR2', 'IPR2', 'Tallin'),  
       ('PR3', 'IPR3', 'Pskov'),  
       ('PR4', 'IPR4', 'Pskov');
```

```

('PR5', 'IPR4', 'Moscow'),
('PR6', 'IPR6', 'Saratov'),
('PR7', 'IPR7', 'Moscow');

INSERT INTO number_of_details (provider_code, detail_code, project_code,
number)
VALUES ('P1', 'D1', 'PR1', 200),
('P1', 'D1', 'PR2', 700),
('P2', 'D3', 'PR1', 400),
('P2', 'D2', 'PR2', 200),
('P2', 'D3', 'PR3', 200),
('P2', 'D3', 'PR4', 500),
('P2', 'D3', 'PR5', 600),
('P2', 'D3', 'PR6', 400),
('P2', 'D3', 'PR7', 800),
('P2', 'D5', 'PR2', 100),
('P3', 'D3', 'PR1', 200),
('P3', 'D4', 'PR2', 500),
('P4', 'D6', 'PR3', 300),
('P4', 'D6', 'PR7', 300),
('P5', 'D2', 'PR2', 200),
('P5', 'D2', 'PR4', 100),
('P5', 'D5', 'PR5', 500),
('P5', 'D5', 'PR7', 100),
('P5', 'D6', 'PR2', 200),
('P5', 'D1', 'PR2', 100),
('P5', 'D3', 'PR4', 200),
('P5', 'D4', 'PR4', 800),
('P5', 'D5', 'PR4', 400),
('P5', 'D6', 'PR4', 500);

```

## 2 ВЫПОЛНЕНИЕ РАБОТЫ

Запросы:

а) Получить все пары названий городов, для которых поставщик из первого города обеспечивает проект во втором городе.

```

SELECT DISTINCT
P.city AS provider_city,
PR.city AS project_city
FROM number_of_details ND
JOIN provider P ON ND.provider_code = P.provider_code
JOIN project PR ON ND.project_code = PR.project_code;

```

	provider_city	project_city
1	Tallin	Tallin
2	Moscow	Minsk
3	Minsk	Pskov
4	Tallin	Pskov
5	Tallin	Moscow
6	Tallin	Saratov
7	Minsk	Moscow
8	Tallin	Minsk
9	Kiev	Pskov
10	Kiev	Tallin
11	Moscow	Tallin
12	Kiev	Moscow

Рисунок 1 – Результат выполнения запроса а)

б) Получить номера проектов, обеспечиваемых по крайней мере всеми деталями поставщика П1.

```
SELECT DISTINCT project_code
FROM number_of_details
WHERE detail_code IN (
SELECT detail_code
FROM number_of_details
WHERE provider_code = 'P1'
);
```

	project_code
1	PR1
2	PR2

Рисунок 2 – Результат выполнения запроса б)

в) Получить все отправки, где количество находится в диапазоне от 300 до 750 включительно.

```
SELECT * FROM number_of_details
WHERE number BETWEEN 300 AND 750;
```

	provider_code	detail_code	project_code	number
1	P1	D1	PR2	700
2	P2	D3	PR1	400
3	P2	D3	PR4	500
4	P2	D3	PR5	600
5	P2	D3	PR6	400
6	P3	D4	PR2	500
7	P4	D6	PR3	300
8	P4	D6	PR7	300
9	P5	D5	PR5	500
10	P5	D5	PR4	400
11	P5	D6	PR4	500

Рисунок 3 – Результат выполнения запроса в)

г) Получить все такие тройки ”номера поставщиков-номера деталей-номера проектов для которых выводимые поставщик, деталь и проект не размещены в одном городе.

```

SELECT DISTINCT
P.provider_code,
D.detail_code,
PR.project_code
FROM number_of_details ND
JOIN provider P ON ND.provider_code = P.provider_code
JOIN detail D ON ND.detail_code = D.detail_code
JOIN project PR ON ND.project_code = PR.project_code
WHERE NOT (P.city = D.city AND D.city = PR.city);

```

	provider_code	detail_code	project_code
1	P1	D1	PR1
2	P5	D2	PR4
3	P5	D2	PR2
4	P5	D6	PR4
5	P5	D6	PR2
6	P2	D3	PR3
7	P5	D5	PR4
8	P5	D5	PR7
9	P2	D2	PR2
10	P3	D3	PR1
11	P4	D6	PR7
12	P2	D3	PR6
13	P2	D5	PR2
14	P2	D3	PR5
15	P5	D1	PR2
16	P4	D6	PR3
17	P5	D3	PR4
18	P5	D5	PR5
19	P2	D3	PR1
20	P3	D4	PR2
21	P2	D3	PR4
22	P5	D4	PR4
23	P1	D1	PR2
24	P2	D3	PR7

Рисунок 4 – Результат выполнения запроса г)

д) Получить общее количество деталей Д1, поставляемых поставщиком П1.

```
SELECT COUNT(*) FROM number_of_details
WHERE provider_code = 'P1' AND detail_code = 'D1';
```

	count
1	2

Рисунок 5 – Результат выполнения запроса д)

е) Получить номера поставщиков, поставляющих одну и ту же деталь для всех проектов.

```
SELECT provider_code
FROM number_of_details
GROUP BY provider_code
HAVING COUNT(DISTINCT detail_code) = 1;
```

provider_code	
1	P1
2	P4

Рисунок 6 – Результат выполнения запроса е)

ж) Получить номера проектов, для которых среднее количество поставляемых деталей Д1 больше, чем наибольшее количество любых деталей, поставляемых для проекта ПР1.

```
SELECT ND.project_code
FROM number_of_details ND
WHERE ND.detail_code = 'D1'
GROUP BY ND.project_code
HAVING AVG(ND.number) > (
SELECT MAX(ND2.number)
FROM number_of_details ND2
WHERE ND2.project_code = 'PR1'
);
```

project_code	
1	PR1
2	PR2

Рисунок 7 – Результат выполнения запроса ж)

з) Получить номера проектов, использующих по крайней мере одну деталь, имеющуюся у поставщика П1.

```
SELECT DISTINCT project_code
FROM number_of_details
WHERE detail_code IN (
SELECT detail_code
FROM number_of_details
WHERE provider_code = 'P1'
);
```

project_code	
1	PR1
2	PR2

Рисунок 8 – Результат выполнения запроса з)

и) Получить все такие тройки ”номера поставщиков-номера деталей-номера проектов для которых никакие из двух выводимых поставщиков, деталей и проектов не размещены в одном городе.

```
SELECT ND.provider_code, ND.detail_code, ND.project_code
FROM number_of_details ND
JOIN provider P ON ND.provider_code = P.provider_code
JOIN detail D ON ND.detail_code = D.detail_code
JOIN project PR ON ND.project_code = PR.project_code
WHERE P.city <> D.city
AND P.city <> PR.city
AND D.city <> PR.city;
```

	provider_code	detail_code	project_code
1	P2	D3	PR1
2	P2	D3	PR3
3	P2	D3	PR4
4	P2	D3	PR5
5	P2	D3	PR6
6	P2	D3	PR7
7	P3	D3	PR1
8	P4	D6	PR3
9	P5	D2	PR2
10	P5	D2	PR4
11	P5	D5	PR5
12	P5	D5	PR7
13	P5	D6	PR2
14	P5	D1	PR2
15	P5	D3	PR4
16	P5	D4	PR4
17	P5	D5	PR4
18	P5	D6	PR4

Рисунок 9 – Результат выполнения запроса и)

к) Получить номера проектов, для которых не поставляются красные детали поставщиками из Лондона.

```
SELECT P.project_code
FROM project P
WHERE NOT EXISTS (
SELECT 1
FROM number_of_details ND
JOIN provider PR ON ND.provider_code = PR.provider_code
JOIN detail D ON ND.detail_code = D.detail_code
WHERE ND.project_code = P.project_code
```



```
AND D.color = 'Red'  
AND PR.city = 'London'  
);
```



Рисунок 10 – Результат выполнения запроса к)

## ВЫВОД

В ходе выполнения лабораторной работы были закреплены навыки проектирования и работы с реляционными базами данных на примере СУБД PostgreSQL: созданы таблицы, заполненные тестовыми данными, а также реализован ряд SQL-запросов различной сложности. В результате удалось продемонстрировать понимание математических основ реляционных языков, использование связей между таблицами и применение условий отбора данных для решения практических задач.