

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

Кафедра: интеллектуальных информационных технологий

Отчёт
По дисциплине
«Проектирование баз знаний»
Тема: «Математические основы реляционных языков»

Написал:
Астахов А.С., гр. 321701

Проверил
Ерофеев И. А.

Минск 2025

Введение

Цель: освоить математические основы работы реляционных баз данных, научиться выполнять базовые операции описания и манипуляции с реляционными базами данных

Задача:

Составить последовательность реляционных операций для задач по варианту.

№ варианта	№ задач									
1	32	11	19	14	21	1	5	30	9	25

Содержание

Введение	2
Содержание	3
Индивидуальное задание	4
Вывод	14

Индивидуальное задание

Схема:

```
CREATE TABLE suppliers (  
    supplier_id VARCHAR(10) PRIMARY KEY,  
    name        VARCHAR(100) NOT NULL,  
    status      INTEGER,  
    city        VARCHAR(100)  
);  
  
CREATE TABLE parts (  
    part_id VARCHAR(10) PRIMARY KEY,  
    name    VARCHAR(100) NOT NULL,  
    color   VARCHAR(50),  
    size    INTEGER,  
    city    VARCHAR(100)  
);  
  
CREATE TABLE projects (  
    project_id VARCHAR(10) PRIMARY KEY,  
    name       VARCHAR(100) NOT NULL,  
    city       VARCHAR(100)  
);  
  
CREATE TABLE spj (  
    supplier_id VARCHAR(10) NOT NULL,  
    part_id      VARCHAR(10) NOT NULL,  
    project_id   VARCHAR(10) NOT NULL,  
    quantity     INTEGER NOT NULL,  
    PRIMARY KEY (supplier_id, part_id, project_id),  
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id),  
    FOREIGN KEY (part_id)     REFERENCES parts(part_id),  
    FOREIGN KEY (project_id)  REFERENCES projects(project_id)  
);
```

Операции вставки:

```
INSERT INTO suppliers (supplier_id, name, status, city)
VALUES
('П1', 'Петров',      20, 'Москва'),
('П2', 'Синицин',     10, 'Таллин'),
('П3', 'Фёдоров',     30, 'Таллин'),
('П4', 'Чаянов',      20, 'Минск'),
('П5', 'Крюков',      30, 'Киев');
--для 9
('П6', 'Богатый',     10, 'Лондон')
```

```
INSERT INTO parts (part_id, name, color, size, city) VALUES
('Д1', 'Болт',        'красный', 12, 'Москва'),
('Д2', 'Гайка',        'зелёная', 17, 'Минск'),
('Д3', 'Диск',         'чёрный', 17, 'Вильнюс'),
('Д4', 'Диск',         'чёрный', 14, 'Москва'),
('Д5', 'Корпус',       'красный', 12, 'Минск'),
('Д6', 'Крышки',       'красный', 19, 'Москва');
```

```
INSERT INTO projects (project_id, name, city) VALUES
('ПР1', 'ИПР1',       'Минск'),
('ПР2', 'ИПР2',       'Таллин'),
('ПР3', 'ИПР3',       'Псков'),
('ПР4', 'ИПР4',       'Псков'),
('ПР5', 'ИПР5',       'Москва'),
('ПР6', 'ИПР6',       'Саратов'),
('ПР7', 'ИПР7',       'Москва'),
--для 21 и 30
('ПР8', 'ИПР8',      'Лондон');
```

```
INSERT INTO spj (supplier_id, part_id, project_id, quantity)
VALUES
('П1', 'Д1', 'ПР1',  200),
('П1', 'Д1', 'ПР2',  700),
('П2', 'Д3', 'ПР1',  400),
('П2', 'Д2', 'ПР2',  200),
('П2', 'Д3', 'ПР3',  200),
('П2', 'Д3', 'ПР4',  500),
('П2', 'Д3', 'ПР5',  600),
('П2', 'Д3', 'ПР6',  400),
('П2', 'Д3', 'ПР7',  800),
('П2', 'Д5', 'ПР2',  100),
('П3', 'Д3', 'ПР1',  200),
('П3', 'Д4', 'ПР2',  500),
('П4', 'Д6', 'ПР3',  300),
('П4', 'Д6', 'ПР7',  300),
```

```
('П5','Д2','ПР2', 200),  
('П5','Д2','ПР4', 100),  
('П5','Д5','ПР5', 500),  
('П5','Д5','ПР7', 100),  
('П5','Д6','ПР2', 200),  
('П5','Д1','ПР2', 100),  
('П5','Д3','ПР4', 200),  
('П5','Д4','ПР4', 800),  
('П5','Д5','ПР4', 400),  
('П5','Д6','ПР4', 500),  
-- добавлена для 21 и 30 запросов  
('П5','Д4','ПР8', 100),  
--для 9  
('П6','Д4','ПР8', 200);
```

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

```
WITH parts AS (  
    SELECT DISTINCT part_id  
    FROM spj  
    WHERE supplier_id = 'П1'  
)  
SELECT project_id  
FROM spj  
WHERE supplier_id = 'П1'  
GROUP BY project_id  
HAVING COUNT(DISTINCT part_id) = (SELECT COUNT(*) FROM parts);
```

Результат:

"ПР1"

"ПР2"

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

```
SELECT suppliers.city AS first_city, projects.city AS second_city
FROM spj
JOIN suppliers ON spj.supplier_id = suppliers.supplier_id
JOIN projects ON spj.project_id = projects.project_id;
```

Результат:

"Москва"	"Минск"
"Москва"	"Таллин"
"Таллин"	"Минск"
"Таллин"	"Таллин"
"Таллин"	"Псков"
"Таллин"	"Псков"
"Таллин"	"Москва"
"Таллин"	"Саратов"
"Таллин"	"Москва"
"Таллин"	"Таллин"
"Таллин"	"Минск"
"Таллин"	"Таллин"
"Минск"	"Псков"
"Минск"	"Москва"
"Киев"	"Таллин"
"Киев"	"Псков"
"Киев"	"Москва"
"Киев"	"Москва"
"Киев"	"Таллин"
"Киев"	"Таллин"
"Киев"	"Псков"
"Киев"	"Псков"
"Киев"	"Псков"
"Киев"	"Псков"
"Киев"	"Лондон"
"Лондон"	"Лондон"

19. Получить имена проектов, обеспечиваемых поставщиком П1.

```
SELECT DISTINCT projects.name  
FROM spj  
JOIN projects ON spj.project_id = projects.project_id  
WHERE spj.supplier_id = 'П1';
```

Результат:

"ИПР1"

"ИПР2"

14. Получить все такие пары номеров деталей, которые обе поставляются одновременно одним поставщиком.

```
SELECT DISTINCT a.supplier_id,  
                a.part_id AS part_a,  
                b.part_id AS part_b  
FROM spj a  
CROSS JOIN spj b  
WHERE a.supplier_id = b.supplier_id  
      AND a.part_id < b.part_id  
ORDER BY a.supplier_id, a.part_id;
```

Результат:

```
"П2" "Д2" "Д3"  
"П2" "Д2" "Д5"  
"П2" "Д3" "Д5"  
"П3" "Д3" "Д4"  
"П5" "Д1" "Д2"  
"П5" "Д1" "Д3"  
"П5" "Д1" "Д4"  
"П5" "Д1" "Д5"  
"П5" "Д1" "Д6"  
"П5" "Д2" "Д3"  
"П5" "Д2" "Д4"  
"П5" "Д2" "Д5"  
"П5" "Д2" "Д6"  
"П5" "Д3" "Д4"  
"П5" "Д3" "Д5"  
"П5" "Д3" "Д6"  
"П5" "Д4" "Д5"  
"П5" "Д4" "Д6"  
"П5" "Д5" "Д6"
```

21. Получить номера деталей, поставляемых для какого-либо проекта в Лондоне.

```
SELECT DISTINCT spj.part_id
FROM spj
JOIN projects ON spj.project_id = projects.project_id
WHERE projects.city = 'Лондон'
```

Результат:

"Д4"

1. Получить полную информацию обо всех проектах.

```
SELECT * FROM projects;
```

Результат:

"ПР1"	"ИПР1"	"Минск"
"ПР2"	"ИПР2"	"Таллин"
"ПР3"	"ИПР3"	"Псков"
"ПР4"	"ИПР4"	"Псков"
"ПР5"	"ИПР5"	"Москва"
"ПР6"	"ИПР6"	"Саратов"
"ПР7"	"ИПР7"	"Москва"
"ПР8"	"ИПР8"	"Лондон"

5. Получить все сочетания "цвета деталей-города деталей".

```
SELECT color, city from parts;
```

Результат:

```
"красный" "Москва"  
"зелёная" "Минск"  
"чёрный" "Вильнюс"  
"чёрный" "Москва"  
"красный" "Минск"  
"красный" "Москва"
```

30. Получить номера деталей, поставляемых для лондонских проектов.

```
SELECT DISTINCT spj.part_id  
FROM spj  
JOIN projects ON spj.project_id = projects.project_id  
WHERE projects.city = 'Лондон'
```

Результат:

```
"Д4"
```

9. Получить номера деталей, поставляемых поставщиком в Лондоне.

```
SELECT part_id from spj  
join suppliers on spj.supplier_id = suppliers.supplier_id  
WHERE suppliers.city = 'Лондон';
```

Результат:

"Д4"

25. Получить номера проектов, город которых стоит первым в алфавитном списке городов.

```
SELECT project_id from projects ORDER by city LIMIT 1;
```

Результат:

"ПР8"

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.