

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

Дисциплина: «Проектировка баз знаний»

Лабораторная работа №1 по теме:
«Математические основы реляционных языков»

Студент гр. 121702

Заломов Р.А.

Проверила:

Липницкая Н. Г.

Минск 2023

Тема

Математические основы реляционных языков

Цель

Освоение основ работы с реляционными базами данных

Задание

1. По словесному запросу к базе данных преподавателей, предметов и студенческих групп составить последовательность реляционных операций, необходимых для выполнения этого запроса (текст операций).
2. Привести результат выполнения каждой операции в виде отношения (таблицы)

Условие задания

1. Требуется создать базы данных с таблицами, представленными ниже (таблицы 1-4), а также написать запросы к ним

ПРЕПОДАВАТЕЛЬ

ТАБЛИЦА 1.1.

ЛичныйНомер	Фамилия	Должность	Кафедра	Специальность	ТелефонДомашний
221Л	Фролов	Доцент	ЭВМ	АСОИ, ЭВМ	487
222Л	Костин	Доцент	ЭВМ	ЭВМ	543
225Л	Бойко	Профессор	АСУ	АСОИ, ЭВМ	112
430Л	Глазов	Ассистент	ТФ	СД	421
110Л	Петров	Ассистент	Экономики	Международная экономика	324

ПРЕДМЕТ

ТАБЛИЦА 1.2.

КодовыйНомерПредмета	НазваниеПредмета	КоличествоЧасов	Специальность	Семестр
12П	Мини ЭВМ	36	ЭВМ	1
14П	ПЭВМ	72	ЭВМ	2
17П	СУБД ПК	48	АСОИ	4
18П	ВКСС	52	АСОИ	6
34П	Физика	30	СД	6
22П	Аудит	24	Бухучета	3

СТУДЕНЧЕСКАЯ ГРУППА

ТАБЛИЦА 1.3.

КодовыйНомер Группы	Название Группы	Количество Человек	Специальность	Фамилия Старосты
8Г	Э-12	18	ЭВМ	Иванова
7Г	Э-15	22	ЭВМ	Сеткин
4Г	АС-9	24	АСОИ	Балабанов
3Г	АС-8	20	АСОИ	Чижов
17Г	С-14	29	СД	Амросов
12Г	М-6	16	Международная экономика	Трубин
10Г	Б-4	21	Бухучет	Зязюткин

ПРЕПОДАВАТЕЛЬ_ПРЕПОДАЕТ_ПРЕДМЕТЫ_В_ГРУППАХ ТАБЛИЦА 1.4.

КодовыйНом ер Группы	КодовыйНом ер Предмета	ЛичныйНом ер	НомерАудитор ии
8Г	12П	222Л	112
8Г	14П	221Л	220
8Г	17П	222Л	112
7Г	14П	221Л	220
7Г	17П	222Л	241
7Г	18П	225Л	210
4Г	12П	222Л	112
4Г	18П	225Л	210
3Г	12П	222Л	112
3Г	17П	221Л	241
3Г	18П	225Л	210
17Г	12П	222Л	112
17Г	22П	110Л	220
17Г	34П	430Л	118
12Г	12П	222Л	112
12Г	22П	110Л	210
10Г	12П	222Л	210
10Г	22П	110Л	210

2. Требуется создать базы данных с таблицами, представленными ниже (таблицы 1-4), а также написать запросы к ним

Поставщики S

П#	Имя П	Статус	Город
П1	Петров	20	Москва
П2	Синицин	10	Таллинн
П3	Федоров	30	Таллинн
П4	Чаянов	20	Минск
П5	Крюков	30	Киев

Детали Р

Д#	Имя Д	Цвет	Размер	Город
Д1	Болт	Красный	12	Москва
Д2	Гайка	Зеленая	17	Минск
Д3	Диск	Черный	17	Вильнюс
Д4	Диск	Черный	14	Москва
Д5	Корпус	Красный	12	Минск
Д6	Крышки	Красный	19	Москва

Проекты J

ПР#	Имя ПР	Город
ПР1	ИПР1	Минск
ПР2	ИПР2	Таллинн
ПР3	ИПР3	Псков
ПР4	ИПР4	Псков
ПР5	ИПР4	Москва
ПР6	ИПР6	Саратов
ПР7	ИПР7	Москва

Кол-во деталей поставляемых одним поставщиком для одного проекта

П#	Д#	ПР#	S
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

Выполнение задания

Задание 1.

Запросы для создания таблиц

Таблица 1.1.

```
CREATE TABLE professors (  
    personal_number VARCHAR (4) PRIMARY KEY,  
    last_name        VARCHAR (50),  
    position         VARCHAR (50),  
    department       VARCHAR (50),  
    speciality       VARCHAR (50),  
    domestic_phone   VARCHAR (3)  
);
```

Таблица 1.2.

```
CREATE TABLE subjects (  
    code_number  VARCHAR (3) PRIMARY KEY,  
    name         VARCHAR (50),  
    hours_amount SMALLINT,  
    speciality   VARCHAR (50),  
    semester     TINYINT  
);
```

Таблица 1.3.

```
CREATE TABLE student_groups (
    code_number      VARCHAR (3) PRIMARY KEY,
    name             VARCHAR (4),
    student_amount   TINYINT,
    speciality       VARCHAR (50),
    headman_last_name VARCHAR (50)
);
```

Таблица 1.4.

```
CREATE TABLE professors_subjects (
    group_code_number VARCHAR (3),
    subject_code_number VARCHAR (3),
    professor_personal_number VARCHAR (4),
    audience VARCHAR (3),
    FOREIGN KEY (group_code_number) REFERENCES student_groups (code_number) ON DELETE SET NULL,
    FOREIGN KEY (subject_code_number) REFERENCES subjects (code_number) ON DELETE SET NULL,
    FOREIGN KEY (professor_personal_number) REFERENCES professors (personal_number) ON DELETE SET NULL
);
```

Запросы для заполнения таблиц

Таблица 1.1.

```
INSERT INTO professors (personal_number, last_name, position, department, speciality, domestic_phone) VALUES
('221Л', 'Фролов', 'Доцент', 'ЭВМ', 'АСОИ, ЭВМ', 487),
('222Л', 'Костин', 'Доцент', 'ЭВМ', 'ЭВМ', 543),
('225Л', 'Бойко', 'Профессор', 'АСУ', 'АСОИ, ЭВМ', 112),
('430Л', 'Глазов', 'Ассистент', 'ТФ', 'СД', 421),
('110Л', 'Петров', 'Ассистент', 'Экономики', 'Международная экономика', 324);
```

Таблица 1.2.

```
INSERT INTO subjects (code_number, name, hours_amount, speciality, semester) VALUES
('12П', 'Мини ЭВМ', 36, 'ЭВМ', 1),
('14П', 'ПЭВМ', 72, 'ЭВМ', 2),
('17П', 'СУБД ПК', 48, 'АСОИ', 4),
('18П', 'ВКСС', 52, 'АСОИ', 6),
('34П', 'Физика', 30, 'СД', 6),
('22П', 'Аудит', 24, 'Бухучета', 3);
```

Таблица 1.3.

```
INSERT INTO student_groups (code_number, name, student_amount, speciality, headman_last_name) VALUES
('8Г', 'Э-12', 18, 'ЭВМ', 'Иванова'),
('7Г', 'Э-15', 22, 'ЭВМ', 'Сеткин'),
('4Г', 'АС-9', 24, 'АСОИ', 'Балабанов'),
('3Г', 'АС-8', 20, 'АСОИ', 'Чижев'),
('17Г', 'С-14', 29, 'СД', 'Амросов'),
('12Г', 'М-6', 16, 'Международная экономика', 'Трубин'),
('10Г', 'Б-4', 21, 'Бухучет', 'Зязюткин');
```

Таблица 1.4.

```
INSERT INTO professors_subjects (group_code_number, subject_code_number, professor_personal_number, audience) VALUES
('8Г', '12П', '222Л', 112),
('8Г', '14П', '221Л', 220),
('8Г', '17П', '222Л', 112),
('7Г', '14П', '221Л', 220),
('7Г', '17П', '222Л', 241),
('7Г', '18П', '225Л', 210),
('4Г', '12П', '222Л', 112),
('4Г', '18П', '225Л', 210),
('3Г', '12П', '222Л', 112),
('3Г', '17П', '221Л', 241),
('3Г', '18П', '225Л', 210),
('17Г', '12П', '222Л', 112),
('17Г', '22П', '110Л', 220),
('17Г', '34П', '430Л', 118),
('12Г', '12П', '222Л', 112),
('12Г', '22П', '110Л', 210),
('10Г', '12П', '222Л', 210),
('10Г', '22П', '110Л', 210);
```

Запросы для работы с таблицами

1.1. Получить полную информацию обо всех преподавателях.

Запрос:

```
-- Task 1. Получить полную информацию обо всех преподавателях.
SELECT * FROM professors;
```

Результат:

	personal_number	last_name	position	department	speciality	domestic_phone
1	221Л	Фролов	Доцент	ЭВМ	АСОИ, ЭВМ	487
2	222Л	Костин	Доцент	ЭВМ	ЭВМ	543
3	225Л	Бойко	Профессор	АСУ	АСОИ, ЭВМ	112
4	430Л	Глазов	Ассистент	ТФ	СД	421
5	110Л	Петров	Ассистент	Экономики	Международная экономика	324

1.2. Получить полную информацию обо всех студенческих группах на специальности ЭВМ.

Запрос:

```
-- Task 2. Получить полную информацию обо всех студенческих группах на специальности ЭВМ.
SELECT * FROM student_groups WHERE speciality = 'ЭВМ';
```

Результат:

	code_number	name	student_amount	speciality	headman_last_name
1	8Г	Э-12	18	ЭВМ	Иванова
2	7Г	Э-15	22	ЭВМ	Сеткин

1.3. Получить личный номер преподавателя и номера аудиторий, в которых они преподают предмет с кодовым номером 18П.

Запрос:

```
-- Task 3. Получить личный номер преподавателя и номера аудиторий, в которых они преподают предмет с кодовым номером 18П.
SELECT professor_personal_number, audience FROM professors_subjects WHERE subject_code_number = '18П';
```


Результат:

	professor_personal_number	audience
1	225Л	210
2	225Л	210
3	225Л	210

- 1.4. Получить номера предметов и названия предметов, которые ведет преподаватель Костин.

Запрос:

```
10 -- Task 4. Получить номера предметов и названия предметов, которые ведет преподаватель Костин.
11 SELECT code_number, name FROM subjects WHERE code_number in
12 (SELECT subject_code_number from professors_subjects WHERE professor_personal_number =
13 (SELECT personal_number FROM professors WHERE last_name = 'Костин'));
```

Результат:

	code_number	name
1	12П	Мини ЭВМ
2	17П	СУБД ПК

- 1.5. Получить номер группы, в которой ведутся предметы преподавателем Фроловым.

Запрос:

```
-- Task 5. Получить номер группы, в которой ведутся предметы преподавателем Фроловым.
SELECT code_number FROM student_groups WHERE code_number in
(SELECT group_code_number from professors_subjects WHERE professor_personal_number =
(SELECT personal_number FROM professors WHERE last_name = 'Фролов'));
```

Результат:

	code_number
1	3Г
2	7Г
3	8Г

- 1.6. Получить информацию о предметах, которые ведутся на специальности АСОИ.

Запрос:

```
-- Task 6. Получить информацию о предметах, которые ведутся на специальности АСОИ.
SELECT * FROM subjects WHERE speciality = 'АСОИ';
```

Результат:

	personal_number	last_name	position	department	speciality	domestic_phone
1	221Л	Фролов	Доцент	ЭВМ	АСОИ, ЭВМ	487
2	225Л	Бойко	Профессор	АСУ	АСОИ, ЭВМ	112

- 1.7. Получить информацию о преподавателях, которые ведут предметы на специальности АСОИ.

Запрос:

```
-- Task 7. Получить информацию о преподавателях, которые ведут предметы на специальности АСОИ.  
SELECT * FROM professors WHERE speciality LIKE '%АСОИ%';
```

Результат:

	personal_number	last_name	position	department	speciality	domestic_phone
1	221Л	Фролов	Доцент	ЭВМ	АСОИ, ЭВМ	487
2	225Л	Бойко	Профессор	АСУ	АСОИ, ЭВМ	112

- 1.8. Получить фамилии преподавателей, которые ведут предметы в 210 аудитории.

Запрос:

```
-- Task 8. Получить фамилии преподавателей, которые ведут предметы в 210 аудитории.  
SELECT DISTINCT last_name FROM professors JOIN professors_subjects  
ON professors.personal_number = professors_subjects.professor_personal_number  
WHERE professors_subjects.audience = 210;
```

Результат:

	last_name
1	Бойко
2	Петров
3	Костин

- 1.9. Получить названия предметов и названия групп, которые ведут занятия в аудиториях с 100 по 200.

Запрос:

```
-- Task 9. Получить названия предметов и названия групп, которые ведут занятия в аудиториях с 100 по 200.  
SELECT DISTINCT subjects.name AS subject_name, student_groups.name AS group_name FROM professors_subjects  
JOIN subjects ON professors_subjects.subject_code_number = subjects.code_number  
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number  
WHERE professors_subjects.audience >= 100 AND professors_subjects.audience <= 200;
```

Результат:

	subject_name	group_name
1	Мини ЭВМ	Э-12
2	СУБД ПК	Э-12
3	Мини ЭВМ	АС-9
4	Мини ЭВМ	АС-8
5	Мини ЭВМ	С-14
6	Физика	С-14
7	Мини ЭВМ	М-6

- 1.10. Получить пары номеров групп с одной специальности.

Запрос:

```
-- Task 10. Получить пары номеров групп с одной специальностью.  
SELECT first_group.code_number, second_group.code_number FROM student_groups AS first_group JOIN student_groups AS second_group ON  
first_group.speciality = second_group.speciality;
```

Результат:

	code_number	code_number:1
1	8Г	7Г
2	8Г	8Г
3	7Г	7Г
4	7Г	8Г
5	4Г	3Г
6	4Г	4Г
7	3Г	3Г
8	3Г	4Г
9	17Г	17Г
10	12Г	12Г
11	10Г	10Г

- 1.11. Получить общее количество студентов, обучающихся на специальности ЭВМ.

Запрос:

```
-- Task 11. Получить общее количество студентов, обучающихся на специальности ЭВМ.  
SELECT SUM(student_groups.student_amount) AS student_amount FROM student_groups WHERE  
speciality = 'ЭВМ';
```

Результат:

	student_amount
1	40

- 1.12. Получить номера преподавателей, обучающихся студентов по специальности ЭВМ.

Запрос:

```
-- Task 12. Получить номера преподавателей, обучающихся студентов по специальности ЭВМ.  
SELECT DISTINCT professors_subjects.professor_personal_number FROM professors_subjects  
JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number  
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number  
WHERE student_groups.speciality = 'ЭВМ';
```

Результат:

	professor_personal_number
1	222Л
2	221Л
3	225Л

- 1.13. Получить номера предметов, изучаемых всеми студенческими группами.

Запрос:

```
-- Task 13. Получить номера предметов, изучаемых всеми студенческими группами.  
SELECT professors_subjects.subject_code_number FROM professors_subjects  
GROUP BY professors_subjects.subject_code_number  
HAVING COUNT(*) = (SELECT COUNT(*) FROM student_groups);
```

Результат:

subject_code_number

- 1.14. Получить фамилии преподавателей, преподающих те же предметы, что и преподаватель преподающий предмет с номером 14П.

Запрос:

```
-- Task 14. Получить фамилии преподавателей, преподающих те же предметы, что и преподаватель преподающий предмет с номером 14П.
SELECT professors.last_name FROM professors WHERE professors.last_name NOT IN
(
    SELECT DISTINCT professors.last_name FROM professors
    JOIN professors_subjects ON professors.personal_number = professors_subjects.professor_personal_number
    WHERE professors_subjects.subject_code_number NOT IN
        (SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects
        WHERE professors_subjects.professor_personal_number =
            (SELECT DISTINCT professors_subjects.professor_personal_number FROM professors_subjects WHERE
            professors_subjects.subject_code_number = '14П'))
);
```

Результат:

	last_name
1	Фролов

- 1.15. Получить информацию о предметах, которые не ведет преподаватель с личным номером 221Л.

Запрос:

```
-- Task 15. Получить информацию о предметах, которые не ведет преподаватель с личным номером 221Л.
SELECT DISTINCT * FROM subjects WHERE subjects.code_number NOT IN
(SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects
WHERE professors_subjects.professor_personal_number = '221Л');
```

Результат:

	code_number	name	hours_amount	speciality	semester
1	12П	Мини ЭВМ	36	ЭВМ	1
2	18П	ВКСС	52	АСОИ	6
3	22П	Аудит	24	Бухучета	3
4	34П	Физика	30	СД	6

- 1.16. Получить информацию о предметах, которые не изучаются в группе М-6.

Запрос:

```
-- Task 16. Получить информацию о предметах, которые не изучаются в группе М-6.
SELECT DISTINCT * FROM subjects WHERE subjects.code_number NOT IN
(SELECT professors_subjects.subject_code_number FROM professors_subjects
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number
WHERE student_groups.name = 'М-6');
```

Результат:

	code_number	name	hours_amount	speciality	semester
1	14П	ПЭВМ	72	ЭВМ	2
2	17П	СУБД ПК	48	АСОИ	4
3	18П	ВКСС	52	АСОИ	6
4	34П	Физика	30	СД	6

1.17. Получить информацию о доцентах, преподающих в группах 3Г и 8Г.

Запрос:

```
-- Task 17. Получить информацию о доцентах, преподающих в группах 3Г и 8Г.  
SELECT DISTINCT professors.* FROM professors_subjects  
JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number  
WHERE professors.position = 'Доцент' AND professors_subjects.group_code_number IN ('3Г', '8Г');
```

Результат:

	personal_number	last_name	position	department	speciality	domestic_phone
1	222Л	Костин	Доцент	ЭВМ	ЭВМ	543
2	221Л	Фролов	Доцент	ЭВМ	АСОИ, ЭВМ	487

1.18. Получить номера предметов, номера преподавателей, номера групп, в которых ведут занятия преподаватели с кафедры ЭВМ, имеющих специальность АСОИ.

Запрос:

```
-- Task 18. Получить номера предметов, номера преподавателей, номера групп, в которых ведут занятия преподаватели с кафедры ЭВМ, имеющих специальность АСОИ.  
SELECT professors_subjects.subject_code_number, professors_subjects.professor_personal_number, professors_subjects.group_code_number  
FROM professors_subjects JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number  
WHERE professors.department = 'ЭВМ' AND professors.speciality LIKE '%АСОИ%';
```

Результат:

	subject_code_number	professor_personal_number	group_code_number
1	14П	221Л	8Г
2	14П	221Л	7Г
3	17П	221Л	3Г

1.19. Получить номера групп с такой же специальностью, что и специальность преподавателей.

Запрос:

```
-- Task 19. Получить номера групп с такой же специальностью, что и специальность преподавателей.  
SELECT DISTINCT student_groups.code_number FROM professors  
JOIN student_groups ON professors.speciality LIKE (SELECT '%' || student_groups.speciality || '%' FROM student_groups);
```

Результат:

	code_number
1	10Г
2	12Г
3	17Г
4	3Г
5	4Г
6	7Г
7	8Г

- 1.20. Получить номера преподавателей с кафедры ЭВМ, преподающих предметы по специальности, совпадающей со специальностью студенческой группы.

Запрос:

```
-- Task 20. Получить номера преподавателей с кафедры ЭВМ, преподающих предметы по специальности, совпадающей со специальностью студенческой группы.  
SELECT DISTINCT professors.personal_number FROM professors_subjects  
JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number  
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number  
WHERE professors.department = 'ЭВМ' AND professors.speciality = student_groups.speciality;
```

Результат:

	personal_number
1	222Л

- 1.21. Получить специальности студенческой группы, на которых работают преподаватели кафедры АСУ.

Запрос:

```
-- Task 21. Получить специальности студенческой группы, на которых работают преподаватели кафедры АСУ.  
SELECT DISTINCT student_groups.speciality FROM professors_subjects  
JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number  
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number  
WHERE professors.department = 'АСУ';
```

Результат:

	speciality
1	ЭВМ
2	АСОИ

- 1.22. Получить номера предметов, изучаемых группой АС-8.

Запрос:

```
-- Task 22. Получить номера предметов, изучаемых группой АС-8.  
SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects  
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number  
WHERE student_groups.name = 'АС-8';
```

Результат:

	subject_code_number
1	12П
2	17П
3	18П

- 1.23. Получить номера студенческих групп, которые изучают те же предметы, что и студенческая группа АС-8.

Запрос:

```
-- Task 23. Получить номера студенческих групп, которые изучают те же предметы, что и студенческая группа АС-8.
SELECT DISTINCT professors_subjects.group_code_number FROM professors_subjects
WHERE professors_subjects.subject_code_number IN (
SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number
WHERE student_groups.name = 'АС-8'
);
```

Результат:

	group_code_number
1	8Г
2	7Г
3	4Г
4	3Г
5	17Г
6	12Г
7	10Г

- 1.24. Получить номера студенческих групп, которые не изучают предметы, преподаваемых в студенческой группе АС-8.

Запрос:

```
-- Task 24. Получить номера студенческих групп, которые не изучают предметы, преподаваемых в студенческой группе АС-8.
SELECT DISTINCT professors_subjects.group_code_number FROM professors_subjects
WHERE professors_subjects.subject_code_number NOT IN (
SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number
WHERE student_groups.name = 'АС-8'
);
```

Результат:

	group_code_number
1	8Г
2	7Г
3	17Г
4	12Г
5	10Г

- 1.25. Получить номера студенческих групп, которые не изучают предметы, преподаваемых преподавателем 430Л.

Запрос:

```
-- Task 25. Получить номера студенческих групп, которые не изучают предметы, преподаваемых преподавателем 430Л.
SELECT DISTINCT professors_subjects.group_code_number FROM professors_subjects
WHERE professors_subjects.subject_code_number NOT IN (
SELECT DISTINCT professors_subjects.subject_code_number FROM professors_subjects
JOIN professors ON professors_subjects.professor_personal_number = professors.personal_number
WHERE professors.personal_number = '430Л'
);
```

Результат:

	group_code_number
1	8Г
2	7Г
3	4Г
4	3Г
5	17Г
6	12Г
7	10Г

- 1.26. Получить номера преподавателей, работающих с группой Э-15, но не преподающих предмет 12П.

Запрос:

```
-- Task 26. Получить номера преподавателей, работающих с группой Э-15, но не преподающих предмет 12П.
SELECT professors_subjects.professor_personal_number FROM professors_subjects
JOIN student_groups ON professors_subjects.group_code_number = student_groups.code_number
WHERE student_groups.name = 'Э-15' AND professors_subjects.professor_personal_number NOT IN
(SELECT DISTINCT professors_subjects.professor_personal_number FROM professors_subjects
WHERE professors_subjects.subject_code_number = '12П');
```

Результат:

	professor_personal_number
1	221Л
2	225Л

Задание 2

Запросы для создания таблиц

Таблица «Поставщики S»

```
CREATE TABLE vendors (
  id VARCHAR (3) PRIMARY KEY,
  name varchar(50),
  status SMALLSERIAL,
  city varchar(50)
);
```


Таблица «Детали Р»

```
CREATE TABLE details (  
    id VARCHAR (3) PRIMARY KEY,  
    name VARCHAR (50),  
    color VARCHAR (50),  
    size VARCHAR (3),  
    city VARCHAR (50)  
);
```

Таблица «Проекты J»

```
CREATE TABLE projects (  
    id VARCHAR (4) PRIMARY KEY,  
    name VARCHAR (50),  
    city VARCHAR (50)  
);
```

Таблица «Количество деталей, поставляемых одним поставщиком для одного проекта»

```
CREATE TABLE details_for_projects (  
    vendor_id VARCHAR (3),  
    detail_id VARCHAR (3),  
    project_id VARCHAR (4),  
    amount SMALLINT,  
    FOREIGN KEY (vendor_id) REFERENCES vendors (id) ON DELETE SET NULL,  
    FOREIGN KEY (detail_id) REFERENCES details (id) ON DELETE SET NULL,  
    FOREIGN KEY (project_id) REFERENCES projects (id) ON DELETE SET NULL  
);
```


Запросы для работы с таблицами

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

Запрос:

```
-- Task 26. Получить номера проектов, для которых среднее количество поставляемых  
-- деталей Д1 больше, чем наибольшее количество любых деталей, поставляемых для проекта ПР1.  
SELECT DISTINCT project_id FROM details_for_projects WHERE  
(SELECT AVG(amount) FROM details_for_projects WHERE detail_id = 'Д1') >  
(SELECT MAX(amount) FROM details_for_projects WHERE project_id = 'ПР1')  
ORDER BY project_id;
```

Результат:

project_id
character varying (4) 

17. Для каждой детали, поставляемой для проекта, получить номер детали, номер проекта и соответствующее общее количество.

Запрос:

```
-- Task 17. Для каждой детали, поставляемой для проекта, получить номер детали, номер проекта и соответствующее общее количество.  
SELECT DISTINCT detail_id, project_id, SUM(amount) FROM details_for_projects  
GROUP BY detail_id, project_id  
ORDER BY detail_id;
```

Результат:


	detail_id character varying (3)	project_id character varying (4)	sum bigint
1	Д1	ПР1	200
2	Д1	ПР2	800
3	Д2	ПР2	400
4	Д2	ПР4	100
5	Д3	ПР1	600
6	Д3	ПР3	200
7	Д3	ПР4	700
8	Д3	ПР5	600
9	Д3	ПР6	400
10	Д3	ПР7	800
11	Д4	ПР2	500
12	Д4	ПР4	800
13	Д5	ПР2	100
14	Д5	ПР4	400
15	Д5	ПР5	500
16	Д5	ПР7	100
17	Д6	ПР2	200
18	Д6	ПР3	300
19	Д6	ПР4	500
20	Д6	ПР7	300

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

Запрос:

```
-- Task 25. Получить номера проектов, город которых стоит первым в алфавитном списке городов.  
SELECT id FROM projects  
ORDER BY city  
LIMIT 1;
```

Результат:

	id [PK] character varying (4) 
1	ПР1

8. 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.

Запрос:

```
-- Task 8. 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 DISTINCT vendors.id, details.id, projects.id FROM vendors  
JOIN details ON vendors.city != details.city  
JOIN projects ON vendors.city != projects.city AND details.city != projects.city;
```

Результат:

	id character varying (3)	id character varying (3)	id character varying (4)
1	П4	Д3	ПР2
2	П5	Д2	ПР2
3	П4	Д3	ПР5
4	П4	Д3	ПР7
5	П5	Д2	ПР5
6	П4	Д3	ПР3
7	П4	Д1	ПР2
8	П2	Д2	ПР6
9	П2	Д3	ПР1
10	П4	Д4	ПР4
11	П2	Д6	ПР1
12	П5	Д6	ПР2
13	П5	Д3	ПР2
14	П5	Д6	ПР1
15	П1	Д5	ПР4
16	П2	Д4	ПР3
17	П2	Д5	ПР7
18	П3	Д3	ПР3
19	П3	Д3	ПР5
20	П3	Д4	ПР1
21	П2	Д1	ПР1
22	П3	Д2	ПР4
23	П2	Д6	ПР6
24	П2	Д2	ПР3
25	П3	Д1	ПР4
26	П4	Д6	ПР2
27	П3	Д3	ПР7
28	П3	Д4	ПР6
29	П1	Д3	ПР1
30	П3	Д4	ПР4
31	П5	Д5	ПР3
32	П3	Д2	ПР6
33	П1	Д5	ПР2
34	П2	Д2	ПР4
35	П2	Д1	ПР6
36	П5	Д4	ПР1
37	П2	Д6	ПР4
38	П5	Д5	ПР4
39	П3	Д6	ПР3
40	П1	Д3	ПР4
41	П4	Д4	ПР6
42	П3	Д4	ПР3
43	П5	Д1	ПР1
44	П3	Д5	ПР4
45	П5	Д3	ПР1

45	П5	Д3	ПР1
46	П1	Д2	ПР4
47	П3	Д1	ПР1
48	П4	Д1	ПР6
49	П3	Д2	ПР5
50	П2	Д5	ПР3
51	П5	Д1	ПР4
52	П3	Д3	ПР1
53	П4	Д6	ПР4
54	П2	Д2	ПР7
55	П4	Д4	ПР2
56	П3	Д5	ПР6
57	П2	Д3	ПР5
58	П3	Д3	ПР6
59	П1	Д2	ПР2
60	П5	Д5	ПР2
61	П5	Д3	ПР5
62	П5	Д6	ПР3
63	П2	Д1	ПР3
64	П1	Д5	ПР6
65	П5	Д3	ПР3
66	П1	Д5	ПР3
67	П2	Д5	ПР6
68	П3	Д2	ПР3
69	П2	Д4	ПР6
70	П2	Д3	ПР3
71	П1	Д2	ПР3
72	П2	Д6	ПР3
73	П5	Д1	ПР2
74	П5	Д2	ПР3
75	П5	Д2	ПР7
76	П5	Д2	ПР6
77	П4	Д3	ПР6
78	П3	Д5	ПР5
79	П3	Д6	ПР1
80	П5	Д4	ПР6
81	П2	Д1	ПР4
82	П5	Д6	ПР6
83	П1	Д2	ПР6
84	П2	Д5	ПР5
85	П2	Д5	ПР4
86	П5	Д3	ПР6
87	П3	Д6	ПР4
88	П3	Д2	ПР7
89	П2	Д3	ПР6

90	П1	Д3	ПР2
91	П5	Д1	ПР3
92	П5	Д5	ПР7
93	П3	Д1	ПР3
94	П3	Д1	ПР6
95	П4	Д4	ПР3
96	П2	Д4	ПР1
97	П4	Д6	ПР3
98	П4	Д1	ПР4
99	П3	Д3	ПР4
100	П5	Д4	ПР2
101	П3	Д6	ПР6
102	П5	Д2	ПР4
103	П4	Д3	ПР4
104	П2	Д4	ПР4
105	П3	Д5	ПР3
106	П5	Д4	ПР4
107	П5	Д5	ПР5
108	П5	Д5	ПР6
109	П4	Д6	ПР6
110	П4	Д1	ПР3
111	П5	Д4	ПР3
112	П1	Д3	ПР6
113	П5	Д6	ПР4
114	П1	Д3	ПР3
115	П2	Д2	ПР5
116	П5	Д3	ПР4
117	П3	Д5	ПР7
118	П2	Д3	ПР7
119	П2	Д3	ПР4
120	П5	Д3	ПР7
121	П5	Д1	ПР6

2. Получить полную информацию обо всех проектах в Минске.

Запрос:

```
-- Task 2. Получить полную информацию обо всех проектах в Минске.  
SELECT * FROM projects WHERE city = 'Минск';
```

Результат:

	id [PK] character varying (4)	name character varying (50)	city character varying (50)
1	ПР1	ИПР1	Минск

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

Запрос:

```
-- Task 7. Получить все такие тройки "номера поставщиков-номера деталей-номера проектов",  
-- для которых выводимые поставщик, деталь и проект не размещены в одном городе.  
SELECT dfp.vendor_id, dfp.detail_id, dfp.project_id FROM details_for_projects AS dfp  
JOIN vendors ON dfp.vendor_id = vendors.id  
JOIN details ON dfp.detail_id = details.id  
JOIN projects ON dfp.project_id = projects.id  
WHERE vendors.city != details.city AND vendors.city != projects.city AND details.city != projects.city;
```

Результат:

	vendor_id character varying (3)	detail_id character varying (3)	project_id character varying (4)
1	П2	Д3	ПР1
2	П2	Д3	ПР3
3	П2	Д3	ПР4
4	П2	Д3	ПР5
5	П2	Д3	ПР6
6	П2	Д3	ПР7
7	П3	Д3	ПР1
8	П4	Д6	ПР3
9	П5	Д2	ПР2
10	П5	Д2	ПР4
11	П5	Д5	ПР5
12	П5	Д5	ПР7
13	П5	Д6	ПР2
14	П5	Д1	ПР2
15	П5	Д3	ПР4
16	П5	Д4	ПР4
17	П5	Д5	ПР4
18	П5	Д6	ПР4

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

Запрос:

```
-- Task 11. Получить все пары названий городов, для которых поставщик из первого города обеспечивает проект во втором городе.  
SELECT DISTINCT vendors.city AS vend_city, projects.city AS pj_city FROM details_for_projects AS dfp  
JOIN vendors ON dfp.vendor_id = vendors.id  
JOIN projects ON dfp.project_id = projects.id  
WHERE vendors.city != projects.city;
```

Результат:

	vend_city character varying (50)	pj_city character varying (50)
1	Киев	Таллинн
2	Таллинн	Минск
3	Киев	Псков
4	Минск	Москва
5	Киев	Москва
6	Таллинн	Псков
7	Таллинн	Москва
8	Москва	Таллинн
9	Минск	Псков
10	Москва	Минск
11	Таллинн	Саратов

36. Получить все пары номеров поставщиков, скажем, Пх и Пу, такие, что оба эти поставщика поставляют в точности одно и то же множество деталей.

Запрос:


```

WITH NOT_EQ AS
(
    SELECT DISTINCT V1.VENDOR_ID AS VEN1,
                   V2.VENDOR_ID AS VEN2,
                   V1.DETAIL_ID,
                   V2.DETAIL_ID
    FROM DETAILS_FOR_PROJECTS AS V1
    JOIN DETAILS_FOR_PROJECTS AS V2 ON V1.DETAIL_ID != V2.DETAIL_ID
    WHERE
        (
            (SELECT COUNT(DISTINCT DETAIL_ID)
             FROM DETAILS_FOR_PROJECTS
             WHERE VENDOR_ID = V1.VENDOR_ID) !=
            (SELECT COUNT(DISTINCT DETAIL_ID)
             FROM DETAILS_FOR_PROJECTS
             WHERE VENDOR_ID = V2.VENDOR_ID)
        OR
            (SELECT COUNT(DETAIL_ID)
             FROM DETAILS_FOR_PROJECTS
             WHERE (DETAIL_ID = V1.DETAIL_ID
                    AND VENDOR_ID = V2.VENDOR_ID)
                  OR (DETAIL_ID = V2.DETAIL_ID
                    AND VENDOR_ID = V1.VENDOR_ID)) = 0 )
)
SELECT DISTINCT V1.VENDOR_ID,
               V2.VENDOR_ID
FROM DETAILS_FOR_PROJECTS AS V1
JOIN DETAILS_FOR_PROJECTS AS V2 ON V1.VENDOR_ID != V2.VENDOR_ID
WHERE V1.VENDOR_ID NOT IN
    (SELECT VEN1
     FROM NOT_EQ
     WHERE VEN2 = V2.VENDOR_ID)
AND V2.VENDOR_ID NOT IN
    (SELECT VEN2
     FROM NOT_EQ
     WHERE VEN1 = V1.VENDOR_ID);

```

Результат:

vendor_id character varying (3) 	vendor_id character varying (3) 
--	--

16. Получить общее количество деталей Д1, поставляемых поставщиком П1.


Запрос:

```

-- Task 16. Получить общее количество деталей Д1, поставляемых поставщиком П1.
SELECT SUM(details_for_projects.amount) FROM details_for_projects
WHERE details_for_projects.vendor_id = 'П1' AND details_for_projects.detail_id = 'Д1';

```

Результат:


	sum bigint 
1	900

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

Запрос:

```
-- Task 31. Получить номера поставщиков, поставляющих одну и ту же деталь для всех проектов.  
SELECT vendor_id FROM details_for_projects  
GROUP BY vendor_id  
HAVING COUNT(DISTINCT detail_id) = 1;
```

Результат:

	vendor_id character varying (3) 
1	П1
2	П4

Вывод

В лабораторной работе были изучены основы работы с реляционными СУБД и языком SQL. Были выполнены практические задания по созданию базы данных и выполнению запросов. Для работы использовались СУБД PostgreSQL и SQLite. В качестве программ для администрирования были использованы PgAdmin 4 и SQLite Studio. Стоит отметить, что дальнейший выбор СУБД зависит от масштабов проекта, в котором она используется – PostgreSQL подходит для проектов с огромными базами данных, SQLite – для маленьких приложений. Однако на данном этапе в силу небольших баз данных для обоих заданий, отличительной разницы между обоими СУБД обнаружено не было.