МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Базы данных»

Тема: Реализация базы данных в СУБД PostgreSQL

Студентка гр. 1304 _____ Чернякова В.А.

Преподаватель _____ Заславский М.М.

Санкт-Петербург

2023

Цель работы.

Реализовать базу данных в СУБД PostgreSQL.

Задание.

Вариант 3(25).

Необходимо развернуть PostgreSQL локально:

- Написать запросы для создания таблиц из предыдущей лабораторный работы
- Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
- Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
- Исходный код выложить на www.db-fiddle.com для проверки работоспособности
- Исходный код в виде .sql файла запушить в виде PR в репо
- В отчете описать:
    - Цель
    - Текст задания в соответствии с вариантом
    - Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
    - Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
    - Исходный код в приложении
    - Ссылку на исходный код www.db-fiddle.com в приложении
    - Ссылка на PR в приложении
    - Вывод

Выполнение работы.

1. Была установлена кроссплатформенная среда разработки для баз данных DataGrip, скачан клиент для баз данных PostgreSQL.

2. Определение новой схемы.

Создана схема базы данных школы – *school_schema*. То есть такая структура и организация базы данных, которая определяет ее таблицы, поля, связи, ограничения и типы данных.

При создании использовался параметр *if not exists* – не делать ничего, если схема с таким именем уже существует.

На рисунке 1 представлена *school_schema* с ее содержимым.

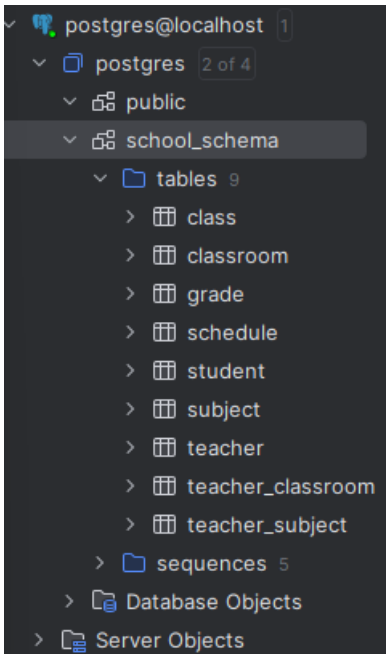


Рисунок 1 – *school_schema*.

3. Создание таблиц.

На основе структуры базы данных, спроектированной в лабораторной работе 1, были созданы соответствующие таблицы.

Для создания использовались SQL-запросы. В нем указывались какая таблица создается, из каких атрибутов она состоит и какой тип данных имеет каждое поле.

Создание таблиц происходит с помощью ключевых слов *create table*.

В теле создания таблицы использовались названия полей, типы данных.

Конкретизация некоторых типов данных:

• *serial primary key* - автоинкрементирующееся числовое значение, которое является первичным ключом.

• *primary key* – первичный ключ.

• *foreign key* – внешний(ие) ключ(и) для связи между таблицы. Синтаксис следующий – *foreign key* (название внешнего ключа), затем пишется ключевое слово *references* после которого указывается имя связанной таблицы и далее в скобках имя столбца из этой таблицы, на который будет указывать внешний ключ. С помощью идущего за описанными раннее данными указывается выражение *on delete*, которое устанавливает выполняемое действие при удалении связанной строки из главной таблицы. В коде данной лабораторной работы использовалось действие *cascade* – автоматически удаление в данном случае строки из зависимой таблицы при удалении связанных строк в главной таблице.

В приложении А представлены запросы по созданию таблиц.

4. Добавление данных.

Для добавления данных в таблицы была использована команда *insert into*. После ключевых словах команды указывается таблица, в которую необходимо добавить данные. Добавляемые данные, а именно столбцы, в которые будут вносится значения, записываются в скобках через запятую. В конце после слова *values* в скобках перечисляются добавляемые значения в соответствии с типами данных, объявленных при создании таблиц.

В приложении А представлены запросы по добавлению данных в таблицы.

На рисунках 2 – 10 изображены итоговые таблицы вместе с данными.

	class_number	class_letter
1	5	А
2	5	Б
3	6	Б
4	6	Г
5	7	В
6	7	Д
7	8	А
8	9	А

Рисунок 2 – таблица *Class*.

	classroom_number
1	101
2	205
3	220
4	135
5	310
6	222
7	112
8	300
9	315

Рисунок 3 – таблица *Classroom*.

	grade_id	student_id	quater_number	subject_id	mark
1	1	1	1	2	4
2	2	2	1	10	5
3	3	3	1	5	3
4	4	4	1	3	3
5	5	5	1	1	4
6	6	6	1	9	4
7	7	7	1	6	5
8	8	8	1	4	5
9	9	9	1	7	5
10	10	10	1	8	3

Рисунок 4 – таблица *Grade*.

	schedule_id	day_name	lesson_number	subject_id	teacher_id	classroom_number	class_number	class_letter
1	1	Понедельник	1	2	1	310	5	А
2	2	Понедельник	2	1	5	220	5	А
3	3	Понедельник	3	3	3	112	5	Б
4	4	Понедельник	4	5	4	220	5	Б
5	5	Вторник	1	7	7	101	6	Б
6	6	Вторник	2	1	2	205	6	Б
7	7	Вторник	5	9	8	135	6	Г
8	8	Вторник	6	8	1	310	6	Г
9	9	Среда	5	4	1	310	7	В
10	10	Среда	6	10	6	300	7	В
11	11	Среда	3	2	1	310	7	Д
12	12	Среда	4	5	4	315	7	Д
13	13	Четверг	1	6	4	135	8	А
14	14	Четверг	2	3	3	112	8	А
15	15	Пятница	1	9	8	135	9	А
16	16	Пятница	2	4	1	310	9	А

Рисунок 5 – таблица *Schedule*.

	student_id	student_name	student_surname	class_number	class_letter
1	1	Басалаев	Леонид	5	А
2	2	Кудашкина	Анастасия	5	А
3	3	Карасев	Алексей	5	Б
4	4	Максимлчкина	Софья	5	Б
5	5	Мангутов	Тимур	6	Б
6	6	Аракчеева	Арина	6	Г
7	7	Чернякова	Валерия	7	В
8	8	Дубровин	Игорь	7	Д
9	9	Никитин	Никита	8	А
10	10	Бомштейн	Юлия	9	А

Рисунок 6 – таблица *Student*.

	subject_id	subject_name
1	1	Математика
2	2	Русский язык
3	3	История
4	4	Физика
5	5	Биология
6	6	Химия
7	7	Иностранный язык
8	8	Литература
9	9	География
10	10	Искусство

Рисунок 7 – таблица *Subject.*

	teacher_id	teacher_name	teacher_surname	teacher_patronymic
1	1	Иванов	Александр	Петрович
2	2	Смирнова	Мария	Игоревна
3	3	Петров	Андрей	Сергеевич
4	4	Козлов	Елена	Владимировна
5	5	Соколов	Игорь	Анатольевич
6	6	Волкова	Наталья	Александровна
7	7	Михайлов	Артем	Дмитриевич
8	8	Передерий	Ольга	<null>

Рисунок 8 – таблица *Teacher.*

	teacher_id	classroom_number
1	7	101
2	2	205
3	5	220
4	8	135
5	1	310
6	3	112
7	6	300

Рисунок 9 – таблица *Teacher_classroom.*

	teacher_id	subject_id
1	2	1
2	5	1
3	7	2
4	1	2
5	3	3
6	1	4
7	4	5
8	4	6
9	7	7
10	1	8
11	8	9
12	6	10

Рисунок 10 – таблица *Teacher_subject.*

5. Написание запросов к БД, отвечающих на вопросы из первой лабораторной работы.

На данном этапе лабораторной работы использовались следующие основные команды и операторы:

• *select* – команда извлечения данных из БД. После указывается список столбцов, который необходимо вывести. Далее идет ключевое слово *from* за которым следует имя таблицы для извлечения данных.

• *inner join* – оператор соединения таблиц. Так как таблицы связаны с друг другом в основном через свои первичнвые ключи, то для получения необходимых значений их необходимо объединять. После оператора указывается присоединяемая таблица за которой следует *using*(условие соединение, а именно столбец).

• *where* – оператор фильтрации. Для вывода корректного ответа на запрос необходимо отфлитровать, выбрать нужные данные. После оператора указываются условия, на основании которых и происходит окончательная выборка.

Вопрос 1. Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

Запрос выглядит следующим образом:

```
select subject_name
from school_schema.Subject
    inner join school_schema.Schedule using(subject_id)
where class_number = 9 and class_letter = 'A' and day_name =
'Пятница' and lesson_number = 2;
```

На рисунке 11 представлен результат запроса по вопросу 1.

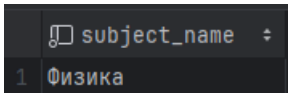


Рисунок 11 – результат запроса по вопросу 1.

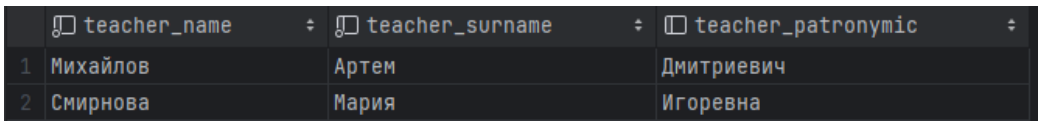
Вопрос 2. Кто из учителей преподает в заданном классе?

Запрос выглядит следующим образом:

```
select teacher_name, teacher_surname, teacher_patronymic
from school_schema.Teacher
```

```
    inner join school_schema.Schedule using(teacher_id)
where class_letter = 'Б' and class_number = 6;
```

На рисунке 12 представлен результат запроса по вопросу 2.



	teacher_name	teacher_surname	teacher_patronymic
1	Михайлов	Артем	Дмитриевич
2	Смирнова	Мария	Игоревна

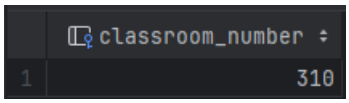
Рисунок 12 – результат запроса по вопросу 2.

Вопрос 3. В каком кабинете будет 5-й урок в среду у некоторого класса?

Запрос выглядит следующим образом:

```
select classroom_number from school_schema.Schedule
where lesson_number = 5 and day_name = 'Среда' and class_number = 7
and class_letter = 'В';
```

На рисунке 13 представлен результат запроса по вопросу 3.



	classroom_number
1	310

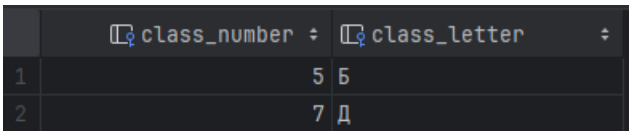
Рисунок 13 – результат запроса по вопросу 3.

Вопрос 4. В каких классах преподает заданный предмет заданный учитель?

Запрос выглядит следующим образом:

```
select class_number, class_letter
from school_schema.Schedule
    inner join school_schema.Teacher using(teacher_id)
    inner join school_schema.Subject using(subject_id)
where teacher_name = 'Козлов' and teacher_surname = 'Елена' and
teacher_patronymic = 'Владимировна' and subject_name = 'Биология';
```

На рисунке 14 представлен результат запроса по вопросу 4.



	class_number	class_letter
1	5	Б
2	7	Д

Рисунок 14 – результат запроса по вопросу 4.

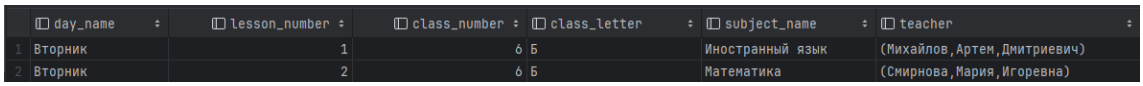
Вопрос 5. Расписание на заданный день недели для указанного класса?

Запрос выглядит следующим образом:

```
select day_name, lesson_number, class_number,class_letter,
subject_name,(teacher_name, teacher_surname, teacher_patronymic) as
teacher
```

```
from school_schema.Schedule
    inner join  school_schema.Subject using(subject_id)
    inner join  school_schema.Teacher using(teacher_id)
    inner join  school_schema.Classroom using(classroom_number)
where class_letter = 'Б' and class_number = 6;
```

На рисунке 15 представлен результат запроса по вопросу 5.



	day_name	lesson_number	class_number	class_letter	subject_name	teacher
1	Вторник	1	6	Б	Иностранный язык	(Михайлов,Артем,Дмитриевич)
2	Вторник	2	6	Б	Математика	(Смирнова,Мария,Игоревна)

Рисунок 15 – результат запроса по вопросу 5.

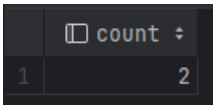
Вопрос 6. Сколько учеников в указанном классе?

Для данного запроса использовалась агрегатная функция для вычиления одного значения над некоторым набором строк. Нахождение количества строк в запросе – *count*.

Запрос выглядит следующим образом:

```
select count(student_id)
from school_schema.Student
    inner join school_schema.Class using(class_number,
class_letter)
where class_number = 5 and class_letter = 'A';
```

На рисунке 16 представлен результат запроса по вопросу 6.



	count
1	2

Рисунок 16 – результат запроса по вопросу 6.

Также в приложении А представлен исходный код целиком. В прииложении Б предоставлена ссылка на исходный код www.db-fiddle.com, ссылка на PR.

Выводы.

В ходе лабораторной работы была реализована база данных в СУБД PostgreSQL.

Изучены основные методы, команды и функции для работы с СУБД PostgreSQL.

Создана основная схема БД, таблицы, соответствующие таблицы, заполненные данными. Выполнены запросы в соответствии с предложенными вопросами к лабораторной работе.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД

Название файла: lab2.sql

```
create schema if not exists school_schema;

create table school_schema.Teacher(
    teacher_id serial primary key,
    teacher_name varchar(50) not null,
    teacher_surname varchar(50) not null,
    teacher_patronymic varchar(50)
);

create table school_schema.Classroom(
    classroom_number int,
    primary key (classroom_number)
);

create table school_schema.Teacher_classroom(
    teacher_id int,
    classroom_number int,
    primary key (teacher_id, classroom_number),
    foreign key (teacher_id) references
school_schema.Teacher(teacher_id) on delete cascade,
    foreign key (classroom_number) references
school_schema.Classroom(classroom_number) on delete cascade
);

create table school_schema.Subject(
    subject_id serial primary key,
    subject_name varchar(50) not null
);

create table school_schema.Teacher_subject(
    teacher_id int,
    subject_id int,
    primary key (teacher_id, subject_id),
```

```
        foreign key (teacher_id) references
school_schema.Teacher(teacher_id) on delete cascade,
        foreign key (subject_id) references
school_schema.Subject(subject_id) on delete cascade
    );

    create table school_schema.Class(
        class_number int,
        class_letter varchar(2),
        primary key (class_number, class_letter)
    );

    create table school_schema.Student(
        student_id serial primary key,
        student_name varchar(50) not null,
        student_surname varchar(50) not null,
        class_number int,
        class_letter varchar(2),
        foreign key (class_number, class_letter) references
school_schema.Class(class_number, class_letter) on delete cascade
    );

    create table school_schema.Grade(
        grade_id serial primary key,
        student_id int,
        quater_number int,
        subject_id int,
        mark int,
        foreign key (student_id) references
school_schema.Student(student_id) on delete cascade,
        foreign key (subject_id) references
school_schema.Subject(subject_id) on delete cascade
    );

    create table school_schema.Schedule(
        schedule_id serial primary key,
        day_name varchar(30) not null,
        lesson_number int,
        subject_id int,
```

```
        teacher_id int,
        classroom_number int,
        class_number int,
        class_letter varchar(2),
        foreign key (subject_id) references
school_schema.Subject(subject_id) on delete cascade,
        foreign key (teacher_id) references
school_schema.Teacher(teacher_id) on delete cascade,
        foreign key (classroom_number) references
school_schema.Classroom(classroom_number) on delete cascade,
        foreign key (class_number, class_letter) references
school_schema.Class(class_number, class_letter) on delete cascade
    );

    insert into school_schema.Teacher(teacher_name, teacher_surname,
teacher_patronymic) values
        ('Иванов', 'Александр', 'Петрович'),
        ('Смирнова', 'Мария', 'Игоревна'),
        ('Петров', 'Андрей', 'Сергеевич'),
        ('Козлов', 'Елена', 'Владимировна'),
        ('Соколов', 'Игорь', 'Анатольевич'),
        ('Волкова', 'Наталья', 'Александровна'),
        ('Михайлов', 'Артем', 'Дмитриевич'),
        ('Передерий', 'Ольга', null);

    insert into school_schema.Classroom(classroom_number) values
        (101), (205), (220), (135), (310), (222), (112), (300), (315);

    insert into school_schema.Teacher_classroom(teacher_id,
classroom_number) values
        (7, 101), (2, 205), (5, 220), (8, 135), (1, 310), (3, 112), (6,
300);

    insert into school_schema.Subject(subject_name) values
        ('Математика'), ('Русский язык'), ('История'), ('Физика'),
('Биология'), ('Химия'), ('Иностранный язык'),
        ('Литература'), ('География'), ('Искусство');
```

```
insert into school_schema.Teacher_subject(teacher_id, subject_id) values
        (2, 1), (5,1), (7, 2), (1, 2), (3, 3), (1, 4), (4, 5), (4, 6), (7, 7), (1, 8), (8, 9), (6, 10);

insert into school_schema.Class(class_number, class_letter) values
        (5, 'А'), (5, 'Б'), (6, 'Б'), (6, 'Г'), (7, 'В'), (7, 'Д'), (8, 'А'), (9, 'А');

insert into school_schema.Student(student_name, student_surname, class_number, class_letter) values
        ('Басалаев', 'Леонид', 5, 'А'),
        ('Кудашкина', 'Анастасия', 5, 'А'),
        ('Карасев', 'Алексей', 5, 'Б'),
        ('Максимлчкина', 'Софья', 5, 'Б'),
        ('Мангутов', 'Тимур', 6, 'Б'),
        ('Аракчеева', 'Арина', 6, 'Г'),
        ('Чернякова', 'Валерия', 7, 'В'),
        ('Дубровин', 'Игорь', 7, 'Д'),
        ('Никитин', 'Никита', 8, 'А'),
        ('Бомштейн', 'Юлия', 9, 'А');

insert into school_schema.Grade(student_id, quater_number, subject_id, mark) values
        (1, 1, 2, 4),
        (2, 1, 10, 5),
        (3, 1, 5, 3),
        (4, 1, 3, 3),
        (5, 1, 1, 4),
        (6, 1, 9, 4),
        (7, 1, 6, 5),
        (8, 1, 4, 5),
        (9, 1, 7, 5),
        (10, 1, 8, 3);

insert into school_schema.Schedule(day_name, lesson_number, subject_id, teacher_id, classroom_number, class_number, class_letter) values
        ('Понедельник', 1, 2, 1, 310, 5,'А'),
```

```
        ('Понедельник', 2, 1, 5, 220, 5,'А'),
        ('Понедельник', 3, 3, 3, 112, 5,'Б'),
        ('Понедельник', 4, 5, 4, 220, 5,'Б'),
        ('Вторник', 1, 7, 7, 101, 6,'Б'),
        ('Вторник', 2, 1, 2, 205, 6,'Б'),
        ('Вторник', 5, 9, 8, 135, 6,'Г'),
        ('Вторник', 6, 8, 1, 310, 6,'Г'),
        ('Среда', 5, 4, 1, 310, 7,'В'),
        ('Среда', 6, 10, 6, 300, 7,'В'),
        ('Среда', 3, 2, 1, 310, 7,'Д'),
        ('Среда', 4, 5, 4, 315, 7,'Д'),
        ('Четверг', 1, 6, 4, 135, 8,'А'),
        ('Четверг', 2, 3, 3, 112, 8,'А'),
        ('Пятница', 1, 9, 8, 135, 9,'А'),
        ('Пятница', 2, 4, 1, 310, 9,'А');

    select subject_name
    from school_schema.Subject
        inner join school_schema.Schedule using(subject_id)
    where class_number = 9 and class_letter = 'А' and day_name =
'Пятница' and lesson_number = 2;

    select teacher_name, teacher_surname, teacher_patronymic
    from school_schema.Teacher
        inner join school_schema.Schedule using(teacher_id)
    where class_letter = 'Б' and class_number = 6;

    select classroom_number from school_schema.Schedule
    where lesson_number = 5 and day_name = 'Среда' and class_number = 7
and class_letter = 'В';

    select class_number, class_letter
    from school_schema.Schedule
        inner join school_schema.Teacher using(teacher_id)
        inner join school_schema.Subject using(subject_id)
    where teacher_name = 'Козлов' and teacher_surname = 'Елена' and
teacher_patronymic = 'Владимировна'
    and subject_name = 'Биология';
```

```
    select    day_name,    lesson_number,    class_number,class_letter,
subject_name,
    (teacher_name, teacher_surname, teacher_patronymic) as teacher
    from school_schema.Schedule
        inner join  school_schema.Subject using(subject_id)
        inner join  school_schema.Teacher using(teacher_id)
        inner join  school_schema.Classroom using(classroom_number)
    where class_letter = 'Б' and class_number = 6;

    select count(student_id)
    from school_schema.Student
        inner     join     school_schema.Class     using(class_number,
class_letter)
    where class_number = 5 and class_letter = 'A';
```

ПРИЛОЖЕНИЕ Б
ССЫЛКИ

Ссылка на PR:

https://github.com/moevm/sql-2023-1304/pull/39

Ссылка на исходный код www.db-fiddle.com:

https://www.db-fiddle.com/f/8oZXoQkUbdrQ2ehz4MjdKa/0