

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

БЕЗОПАСНОСТЬ СЕТЕЙ БАЗ ДАННЫХ

СОЗДАНИЕ БАЗЫ ДАННЫХ В POSTGRESQL

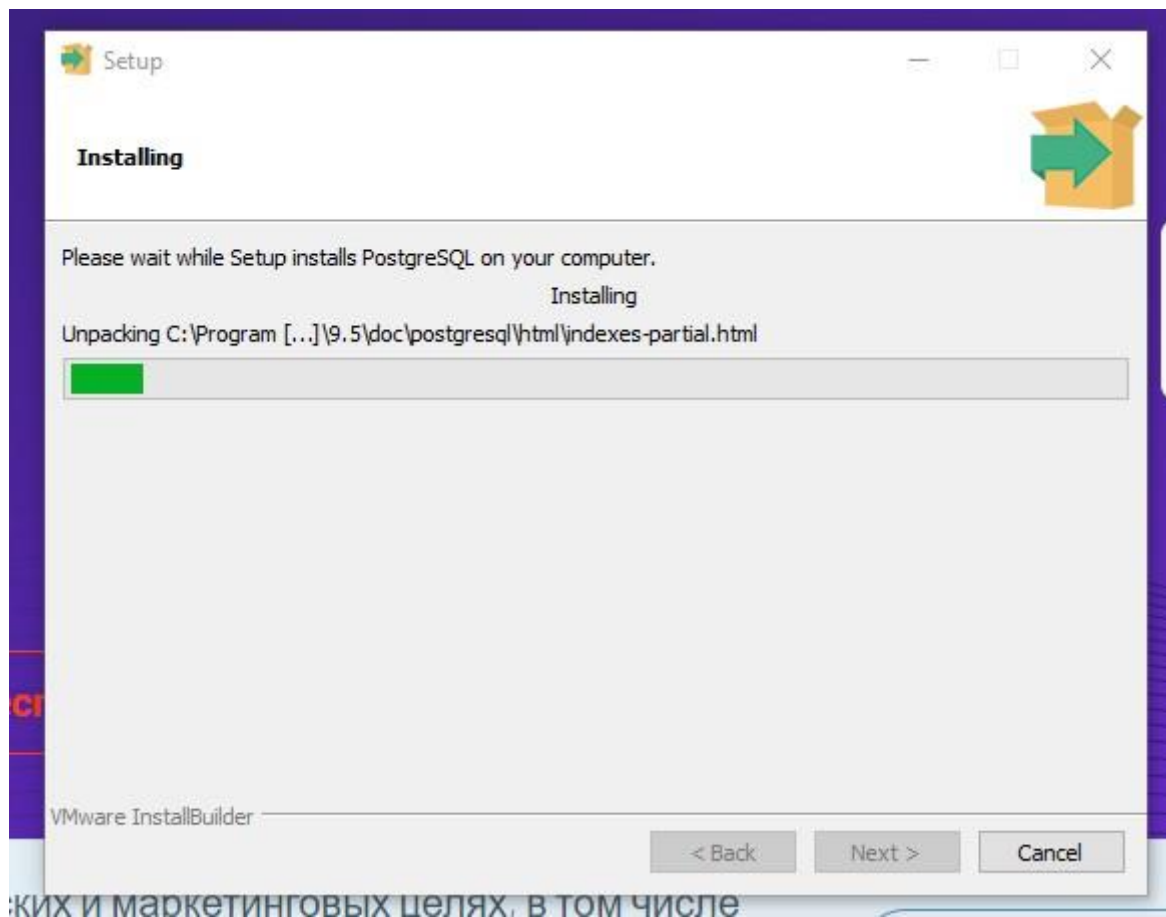
Выполнил: Петрашов Никита Андреевич

Группа: УБ-02

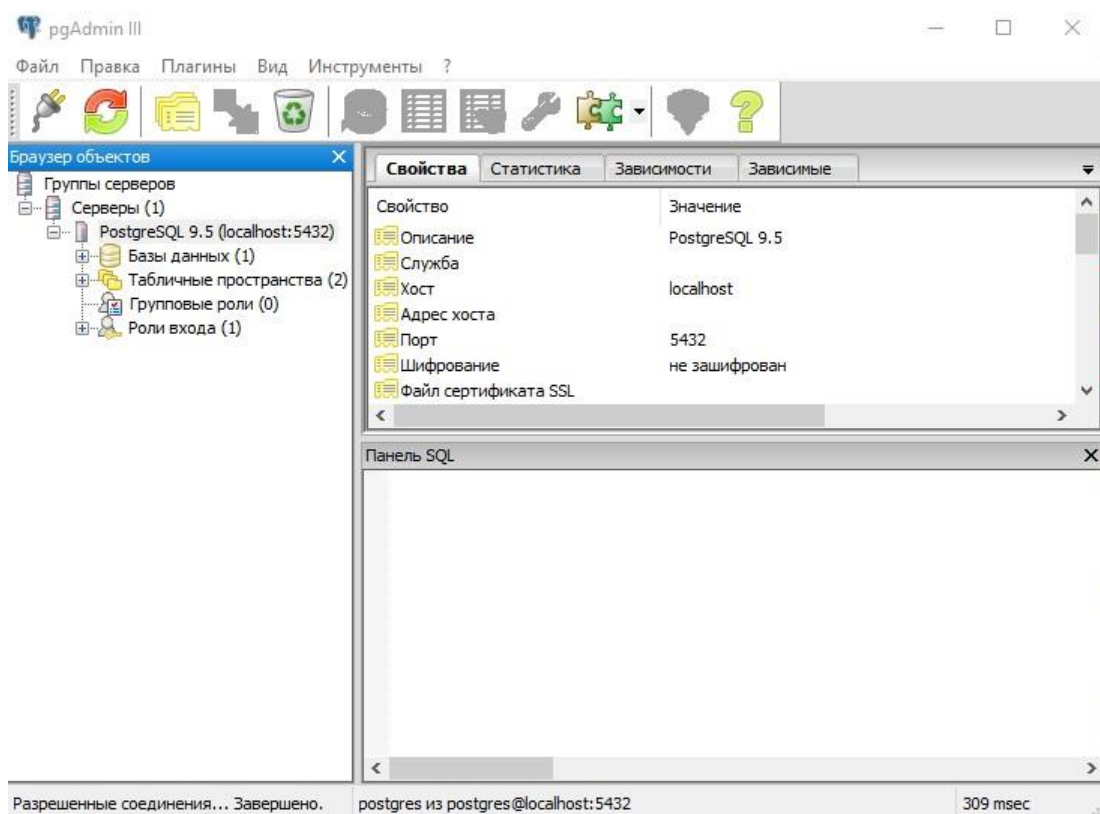
Номер зачетной книжки: 207416

**ВОРОНЕЖ
2023**

1) Устанавливаем PostgreSQL



2) Заходим в PgAdmin и проверяем корректность установки, помимо этого авторизовываемся (вводим пароль, который указали при установке).

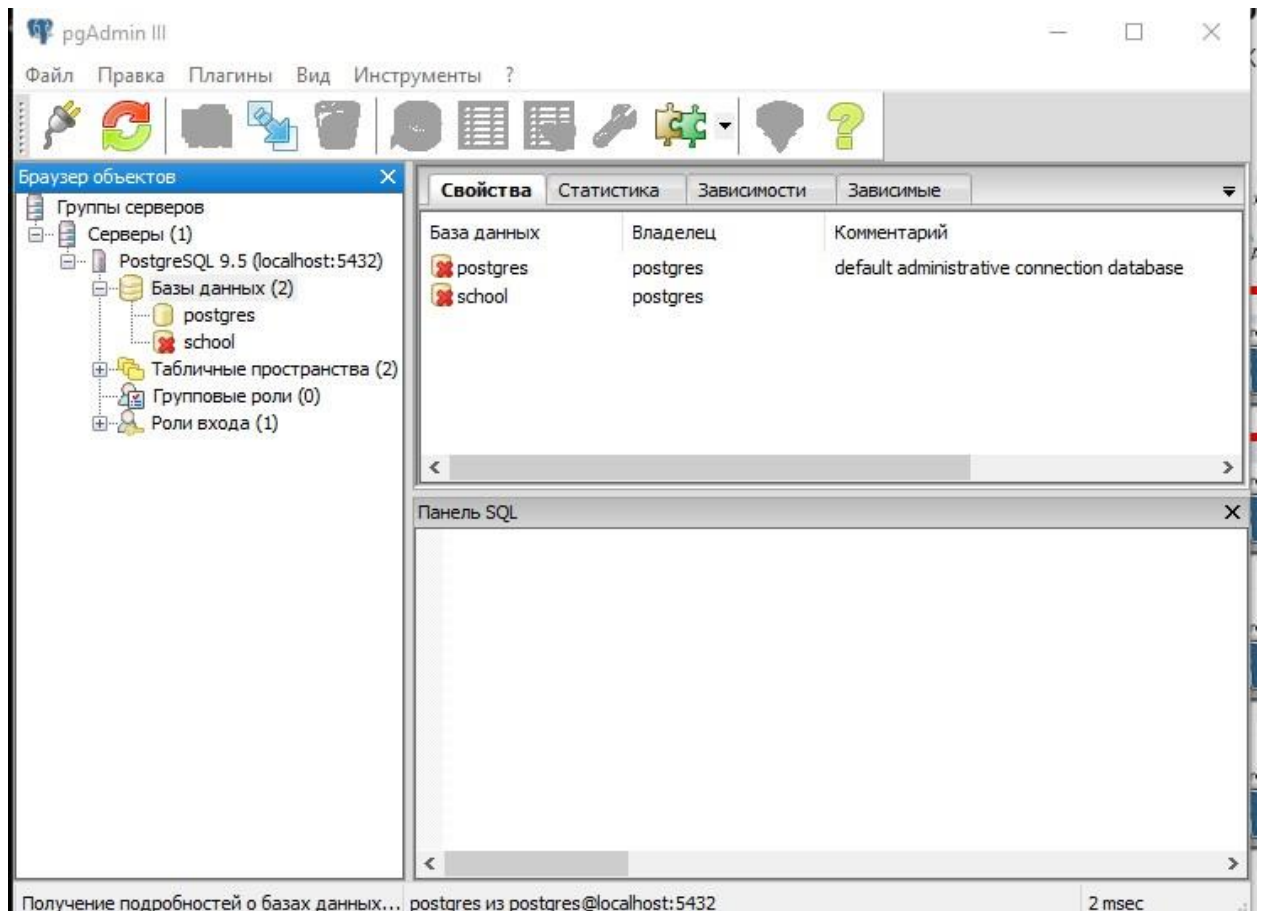


3) Для создания новой базы данных используется оператор CREATE DATABASE. Создадим CREATE DATABASE school. Для того, чтобы подключиться к ней, используем команду \c school (\connect school). Проверяем корректность создание базы данных через PgAdmin.

```
postgres=#
postgres=# CREATE DATABASE school;
CREATE DATABASE
postgres=# \l
```

База данных	Хозяин	Кодировка	LC_COLLATE	LC_CTYPE	Соединение
postgres	postgres	UTF8	Russian_Russia.1251	Russian_Russia.1251	
school	postgres	UTF8	Russian_Russia.1251	Russian_Russia.1251	
template0	postgres	UTF8	Russian_Russia.1251	Russian_Russia.1251	=c/postgres +
template1	postgres	UTF8	Russian_Russia.1251	Russian_Russia.1251	postgres=CtC/postgres +

```
postgres=# CREATE DATABASE school;
ОШИБКА: база данных "school" уже существует
postgres=# clear
postgres=# \c school
Вы подключены к базе данных "school" как пользователь "postgres".
school=# /conninfo
school=# \conninfo
Вы подключены к базе данных "school" как пользователь "postgres" (сервер "localhost", порт "5432").
school=#
```



4)Для создания таблиц базы данных используется оператор CREATE TABLE. Создадим таблицы: Lesson (Урок\Предмет), Teacher (Учитель), Student (Ученик) и Gradebook (Журнал успеваемости).

При создании таблицы Lesson укажем первичный ключ – id_lesson SERIAL PRIMARY KEY, название предмета с типом данных – name_lesson varchar(30), ФИО учителя, который ведет предмет – fuull_name varchar(45), номер класса, у которого идет предмет – id_class varchar(3), начало предмета – start_l_time TIME, конец предмета end_l_time TIME.

При создании таблицы Teacher укажем первичный ключ – id_teacher SERIAL (тип данных, используемый для создания уникального идентификатора для каждого нового объекта) PRIMARY KEY, ФИО – full_name varchar(60), телефон – phone varchar(15),преподавательский стаж – experience integer , преподаваемый урок\предмет – id_lesson int REFERENCES Lesson (id_lesson).

При создании таблицы Student укажем первичный ключ – id_student SERIAL PRIMARY KEY, ФИО – fio_student varchar(60), класс – class varchar(3), телефон – phone varchar(15), дата рождения student_l_date DATE.

При создании таблицы Gradebook укажем первичный ключ – id_gradebook SERIAL PRIMARY KEY, ФИО ученика – fio_student int REFERENCES Student (id_student), предмет – name_lesson int REFERENCES Lesson (id_lesson), ФИО преподавателя full_name int REFERENCES Teacher(id_teacher), последняя оценка–grade integer , средний балл – grade_medium integer.

```
school=# CREATE TABLE Lesson(  
school(# id_lesson SERIAL PRIMARY KEY,  
school(# name_lesson varchar(30) NULL,  
school(# fuull_name varchar(45) NULL,  
school(# id_class varchar(3) NULL,  
school(# start_l_time TIME,  
school(# end_l_time TIME  
school(# );  
CREATE TABLE
```

```

school=# CREATE TABLE Teacher(
school(# id_teacher SERIAL PRIMARY KEY,
school(# full_name varchar(60) NULL,
school(# phone varchar(15) NULL,
school(# experience integer NULL,
school(# id_lesson int REFERENCES Lesson (id_lesson)
school(# );
CREATE TABLE

```

```

school=# CREATE TABLE Student(
school(# id_student SERIAL PRIMARY KEY,
school(# fio_student varchar(60) NULL,
school(# class varchar(3) NULL,
school(# phone varchar(15) NULL,
school(# student_l_date DATE
school(# );
CREATE TABLE

```

```

school=# CREATE TABLE Gradebook(
school(# id_gradebook SERIAL PRIMARY KEY,
school(# fio_student int REFERENCES Student (id_student),
school(# name_lesson int REFERENCES Lesson (id_lesson),
school(# full_name int REFERENCES Teacher(id_teacher),
school(# grade integer NULL,
school(# grade_medium integer NULL
school(# );
CREATE TABLE
school=#

```

После выполнения данного запроса можно получить список созданных таблиц и связей с помощью команды \d. Результат работы команды имеет вид:

```

school=# \d

```

Схема	Таблица	Типы полей	Ссылки
public	gradebook	integer, varchar, integer, integer	
public	gradebook_id_gradebook_seq	integer	gradebook (id_gradebook)
public	lesson	integer, varchar, integer, integer	
public	lesson_id_lesson_seq	integer	lesson (id_lesson)
public	student	integer, varchar, varchar, varchar, date	
public	student_id_student_seq	integer	student (id_student)
public	teacher	integer, varchar, varchar, integer	
public	teacher_id_teacher_seq	integer	teacher (id_teacher)

(8 строк)

Список созданных индексов можно получить с помощью команды \di. Результат представлен в таблице:

```

school=# \di

```

Схема	Индекс	Таблица	Тип индекса
public	gradebook_pkey	gradebook	primary
public	lesson_pkey	lesson	primary
public	student_pkey	student	primary
public	teacher_pkey	teacher	primary

(4 строк)

Для вставки данных в таблицы служит оператор INSERT. Для того, чтобы вывести заполненную таблицу используем оператор SELECT – SELECT * FROM название_таблицы.

```
school=# INSERT INTO Lesson(
school(# name_lesson, fuull_name, id_class, start_l_time, end_l_time)
school-# VALUES ('Математика','Петров Н.А.','11А','08:00:00','08:45:00');
INSERT 0 1
school=# INSERT INTO Lesson(
school(# name_lesson, fuull_name, id_class, start_l_time, end_l_time)
school-# VALUES ('Физка','Климов Н.А.','11Б','08:00:00','08:45:00');
INSERT 0 1
school=# INSERT INTO Lesson(
school(# name_lesson, fuull_name, id_class, start_l_time, end_l_time)
school-# VALUES ('Химия','Карасев Д.Д.','10Б','08:00:00','08:45:00');
INSERT 0 1
school=# INSERT INTO Lesson(
school(# name_lesson, fuull_name, id_class, start_l_time, end_l_time)
school-# VALUES ('ИЗО','Гагарин Д.Д.','10А','08:45:00','09:40:00');
INSERT 0 1
school=# INSERT INTO Lesson(
school(# name_lesson, fuull_name, id_class, start_l_time, end_l_time)
school-# VALUES ('ПСО','Гаврилов В.С.','9А','08:45:00','09:40:00');
INSERT 0 1
school=# SELECT * FROM Lesson;
```

id_lesson	name_lesson	fuull_name	id_class	start_l_time	end_l_time
1	Математика	Петров Н.А.	11А	08:00:00	08:45:00
2	Физка	Климов Н.А.	11Б	08:00:00	08:45:00
3	Химия	Карасев Д.Д.	10Б	08:00:00	08:45:00
4	ИЗО	Гагарин Д.Д.	10А	08:45:00	09:40:00
5	ПСО	Гаврилов В.С.	9А	08:45:00	09:40:00

(5 ёёёюь)

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# VALUES ('Петров Н.А.', ' +79548478474 ', '12', 'Математика');
INSERT 0 1
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# ('Климов Н.А.', ' +79646478474 ', '17', 'Физика');
ОШИБКА: ошибка синтаксиса (примерное положение: "'?<ЁҀR҃ ?..'"')
СТРОКА 3: ('?<ЁҀR҃ ?..', ' +79646478474 ', '17', "'ЁЅЁЁ '');
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# ('Климов Н.А.', ' +79646478474 ', '17', 'Физика');
ОШИБКА: ошибка синтаксиса (примерное положение: "'?<ЁҀR҃ ?..'"')
СТРОКА 3: ('?<ЁҀR҃ ?..', ' +79646478474 ', '17', "'ЁЅЁЁ '');
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# VALUES ('Климов Н.А.', ' +79646478474 ', '17', 'Физика');
INSERT 0 1
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# VALUES ('Карасев Д.Д.', ' +75454554554 ', '21', 'Химия');
INSERT 0 1
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# VALUES ('Гагарин Д.Д.', ' +75454524455 ', '17', 'ИЗО');
INSERT 0 1
```

```
school=# INSERT INTO Teacher(
school(# full_name, phone, experience, id_lessonn)
school=# VALUES ('Гаврилов В.С.', ' +78478478395 ', '7', 'ПСО');
INSERT 0 1
```

```
school=# SELECT * FROM Teacher;
```

id_teacher	full_name	phone	experience	id_lessonn
1	Петров Н.А.	+79548478474	12	Математика
2	Климов Н.А.	+79646478474	17	Физика
3	Карасев Д.Д.	+75454554554	21	Химия
4	Гагарин Д.Д.	+75454524455	17	ИЗО
5	Гаврилов В.С.	+78478478395	7	ПСО

(5 ёёЁюь)

```

school=# INSERT INTO Student(
school(# fio_student, class, phone, student_l_date)
school=# VALUES ('Петрашов Н.А.', '11А', ' +79435878877 ', '2001-11-26');
INSERT 0 1
school=# INSERT INTO Student(
school(# fio_student, class, phone, student_l_date)
school=# VALUES ('Стутко В.Д.', '11А', ' +79435874356 ', '2001-06-17');
INSERT 0 1
school=# INSERT INTO Student(
school(# fio_student, class, phone, student_l_date)
school=# VALUES ('Сароян С.А.', '11А', ' +79764564356 ', '2001-06-24');
INSERT 0 1
school=# INSERT INTO Student(
school(# fio_student, class, phone, student_l_date)
school=# VALUES ('Аничкин Д.Ю.', '10А', ' +79564567788 ', '2002-01-26');
INSERT 0 1
school=# INSERT INTO Student(
school(# fio_student, class, phone, student_l_date)
school=# VALUES ('Бульвар Д.П.', '9А', ' +79764545555 ', '2003-12-26');
INSERT 0 1
school=# SELECT * FROM Student;

```

id_student	fio_student	class	phone	student_l_date
1	Петрашов Н.А	11А	+79435878877	2001-11-26
2	Стутко В.Д.	11А	+79435874356	2001-06-17
3	Сароян С.А	11А	+79764564356	2001-06-24
4	Аничкин Д.Ю	10А	+79564567788	2002-01-26
5	Бульвар Д.П.	9А	+79764545555	2003-12-26

(5 ẽẽёюъ)

school=#

```

school=# INSERT INTO Gradebook(
school(# fio_studentt, name_lessonn, full_nameee, grade, grade_medium)
school=# VALUES ('Петрашов Н.А.', 'Математика', 'Петров Н.А.', 4, 4);
INSERT 0 1
school=# INSERT INTO Gradebook(
school(# fio_studentt, name_lessonn, full_nameee, grade, grade_medium)
school=# VALUES ('Стутко В.Д.', 'Физика', 'Климов Н.А.', 4, 4);
INSERT 0 1
school=# INSERT INTO Gradebook(
school(# fio_studentt, name_lessonn, full_nameee, grade, grade_medium)
school=# VALUES ('Сароян С.А.', 'Химия', 'Карасев Д.Д.', 3, 3);
INSERT 0 1
school=# INSERT INTO Gradebook(
school(# fio_studentt, name_lessonn, full_nameee, grade, grade_medium)
school=# VALUES ('Аничкин Д.Ю.', 'ПСО', 'Гаврилов В.С.', 3, 3);
INSERT 0 1
school=# INSERT INTO Gradebook(
school(# fio_studentt, name_lessonn, full_nameee, grade, grade_medium)
school=# VALUES ('Бульвар Д.П.', 'ИЗО', 'Гагарин Д.Д.', 5, 5);
INSERT 0 1
school=# SELECT * FROM Gradebook;

```

id_gradebook	fio_studentt	name_lessonn	full_nameee	grade	grade_medium
1	Петрашов Н.А	Математика	Петров Н.А.	4	4
2	Стутко В.Д	Физика	Климов Н.А.	4	4
3	Сароян С.А.	Химия	Карасев Д.Д.	3	3
4	Аничкин Д.Ю.	ПСО	Гаврилов В.С.	3	3
5	Бульвар Д.П.	ИЗО	Гагарин Д.Д.	5	5

(5 ẽẽёюъ)

Например, чтобы вывести название предмета, ФИО учителя и время начала занятия из таблицы Lesson и отсортировать их по времени начала занятия, нужно написать запрос следующего вида:

```
SELECT name_lesson, full_name, start_l_time FROM Lesson ORDER BY start_l_time;
```

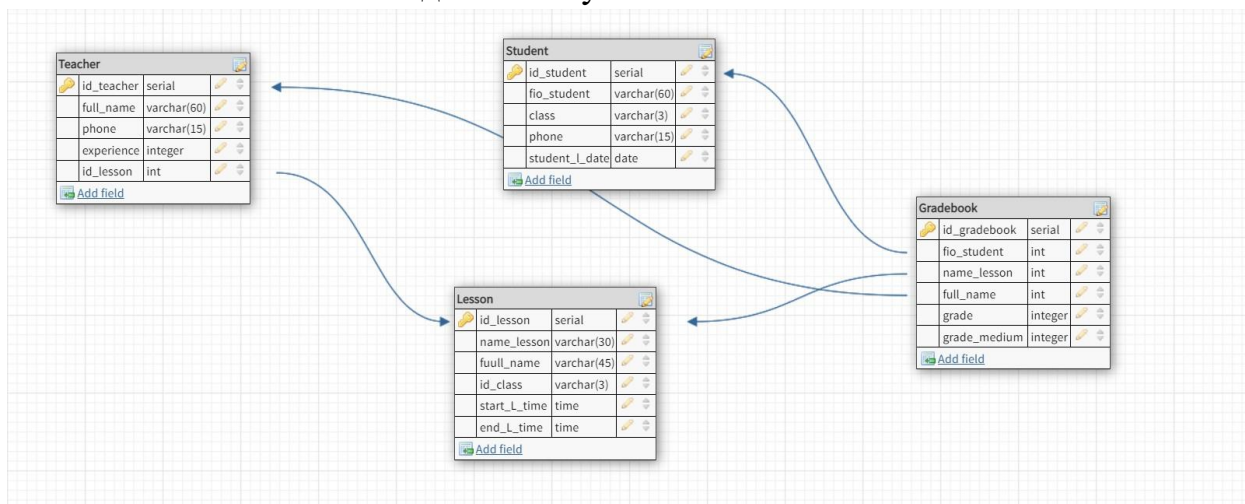
При этом результаты будут отсортированы в порядке возрастания времени начала занятия. Если нужно отсортировать их в порядке убывания, нужно добавить ключевое слово DESC после названия столбца:

```
SELECT name_lesson, full_name, start_l_time FROM Lesson ORDER BY start_l_time DESC;
```

Таким образом, результаты будут отсортированы в порядке убывания времени начала занятия.

```
school=# SELECT name_lesson, fuull_name, start_l_time FROM Lesson ORDER BY start_l_time DESC;
name_lesson | fuull_name | start_l_time
-----+-----+-----
ИЗО          | Гагарин Д.Д. | 08:45:00
ПСО          | Гаврилов В.С. | 08:45:00
Математика   | Петров Н.А.   | 08:00:00
Физка        | Климов Н.А.   | 08:00:00
Химия        | Карасев Д.Д.  | 08:00:00
(5 ÆÆËÿ)
```

Создаем схему



SQL представления

```
school=# CREATE VIEW FIOY AS
school=# SELECT full_name AS FIOYY
school=# FROM Teacher
school=# ORDER BY full_name;
CREATE VIEW
school=# SELECT *FROM FIOY;
      fioyy
-----
Гагарин Д.Д
Гаврилов В.С.
Карасев Д.Д.
Климов Н.А.
Петров Н.А.
(5 строк)
```

Создает представление (VIEW) с именем FIOY которое содержит одну колонку FIOYY. Эта колонка является результатом выборки столбца full_name из таблицы Teacher.

Представление FIOY будет отображать имена всех учителей, отсортированных по алфавиту по возрастанию. Результаты при выводе таблицы будут получены, если бы происходил вывод только одного столбца SELECT full_name FROM Teacher;

Использование представлений для скрытия столбцов и строк

```
school=# CREATE VIEW class AS
school=# SELECT name_lesson, fuull_name, id_class
school=# FROM Lesson;
CREATE VIEW
school=# SELECT * FROM class;
name_lesson | fuull_name | id_class
-----+-----+-----
Математика | Петров Н.А. | 11А
Физка      | Климов Н.А. | 115
Химия      | Карасев Д.Д. | 105
ИЗО        | Гагарин Д.Д. | 10А
ПСО        | Гаврилов В.С. | 9А
(5 строк)
```

код создает представление (view) в базе данных, которое называется "class". Представление содержит данные о клиентах, взятых из таблицы "Lesson".

Представление возвращает три столбца: "Name_lesson" (название предмета), "fuull_name" (имя учителя) и "id_class" (номер класса), которые выбираются из таблицы "Lesson".

Использование представлений для отображения вычисляемых столбцов

```
school=# CREATE VIEW STUD AS
school=# SELECT class,
school=# ('(' || class || ')') || Phone AS Phone
school=# FROM STUDENT;
CREATE VIEW
school=# SELECT FROM STUD;
--
(5 ÷÷÷÷÷)

school=# SELECT * FROM STUD;
 class |      phone
-----+-----
 11A   | (11A) +79435878877
 11A   | (11A) +79435874356
 11A   | (11A) +79764564356
 10A   | (10A) +79564567788
 9A    | (9A) +79764545555
(5 ÷÷÷÷÷)
```

код создает представление (view) в базе данных, которое называется "Stud". Представление отображает два столбца из таблицы "Student" - "class" и "Phone".

Использование представлений для скрытия сложного синтаксиса

```
school=# CREATE VIEW GradebookData AS
school=# SELECT s.fio_student AS StudentName,
school=# l.name_lesson AS LessonName,
school=# t.full_name AS TeacherName,
school=# g.grade AS Grade,
school=# g.grade_medium AS GradeMedium
school=# FROM Gradebook g
school=# JOIN Lesson l ON g.name_lessonn = l.name_lesson
school=# JOIN Teacher t ON g.full_nameee = t.full_name
school=# JOIN Student s ON g.fio_studentt = s.fio_student;
CREATE VIEW
school=# SELECT * FROM GradebookData;
 studentname | lessonname | teachername | grade | grademedium
-----+-----+-----+-----+-----
(0 ÷÷÷÷÷)

school=# SELECT FROM GradebookData;
--
(0 ÷÷÷÷÷)

school=# SELECT * FROM GradebookData;
 studentname | lessonname | teachername | grade | grademedium
-----+-----+-----+-----+-----
(0 ÷÷÷÷÷)

school=# SELECT StudentName, LessonName, TeacherName, Grade FROM GradebookData;
 studentname | lessonname | teachername | grade
-----+-----+-----+-----
(0 ÷÷÷÷÷)
```

Отображает данные из таблицы "Gradebook" и связанных таблиц "Lesson", "Teacher" и "Student". Он выбирает имена столбцов для отображения данных, чтобы сделать их более понятными. При использовании этого представления можно получить информацию о студентах, учителях, уроках и оценках, которые они получили, с помощью одного запроса.

Для получения количества строк в таблице "Lesson" можно использовать следующий код:

```
school=# SELECT COUNT(*) INTO rowcount
school=# FROM Lesson;
SELECT 1
```

Данный запрос поможет выделить данные из таблиц "Lesson", "Teacher" и "Gradebook" и получить общую информацию о количестве оценок, средних оценках и т.д. для каждой уникальной комбинации учителя, названия урока, оценки и среднего балла.

```
school=# SELECT
school=# Teacher.full_name AS TeacherName,
school=# Lesson.name_lesson AS LessonName,
school=# Gradebook.grade AS Grade,
school=# Gradebook.grade_medium AS GradeMedium,
school=# COUNT(Gradebook.id_gradebook) AS TotalGrades
school=# FROM
school=# Gradebook
school=# JOIN Lesson ON Gradebook.name_lessonn = Lesson.name_lesson
school=# JOIN Teacher ON Teacher.id_teacher = Lesson.id_lesson
school=# GROUP BY
school=# Teacher.full_name,
school=# Lesson.name_lesson,
school=# Gradebook.grade,
school=# Gradebook.grade_medium;
```

teachername	lessonname	grade	grademedium	totalgrades
Карасев Д.Д.	Химия	3	3	1
Гагарин Д.Д	ИЗО	5	5	1
Гаврилов В.С.	ПСО	3	3	1
Петров Н.А.	Математика	4	4	1

(4 ёёёёш)

```
school=#
```