ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет по лабораторной работе 3**

**по дисциплине: «Базы данных**

**Тема работы «Ознакомление с основами PostgreSQL. Изучение базовых операций по работе с базой данных»**

**“Обслуживание работы конференции”**

студента очного отделения

2 курса 12001801 группы

Капустина Виктора Сергеевича

Проверил(а):

Петров Денис Васильевич

Белгород 2020

**Лабораторная работа №3**

**Цель работы:**

1. Изучить базовые операции по работе с базой данных.

2. Изучить синтаксис команд.

3. Приобрести навыки создания базы данных, создания, заполнения и модификации таблиц в PostgreSQL.

**PostgreSQL** — свободная объектно-реляционная система управления базами данных.   
**В основные команды psql входят такие как:**

* copyright
* помощь с командами SQL
* помощь с командами psql
* завершение точкой-с-запятой для выполнения запроса
* выйти из команды

Команда **\copyright** – показывает условия распространения (правовые вопросы) между пользователем и компанией.

**\h** – помощь в командах SQL. Достаточно ввести \h и нажать на клавишу Enter, чтобы узнать более подробно об этой команде.

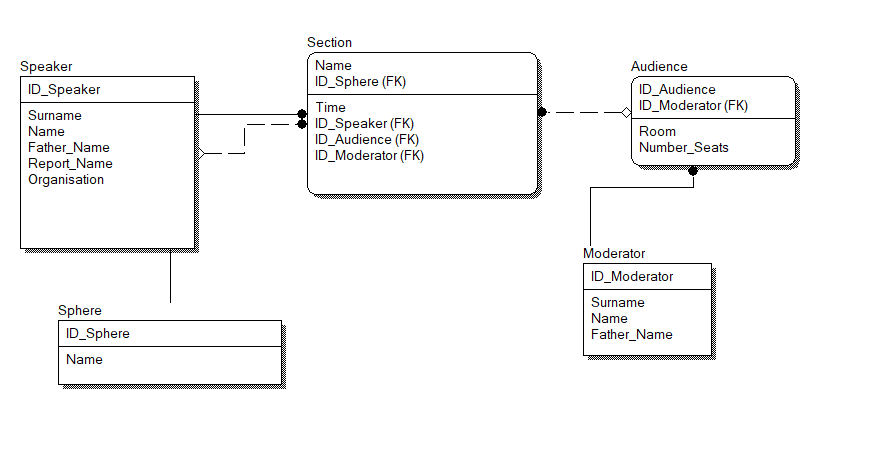
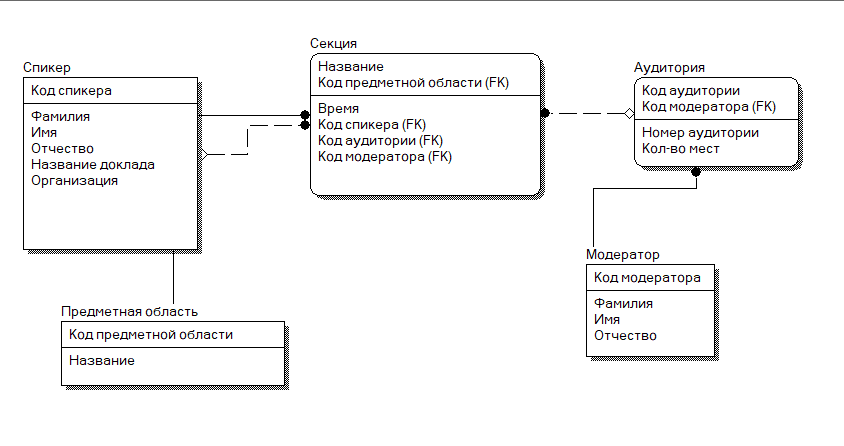
**\?** для помощи по командам psql.

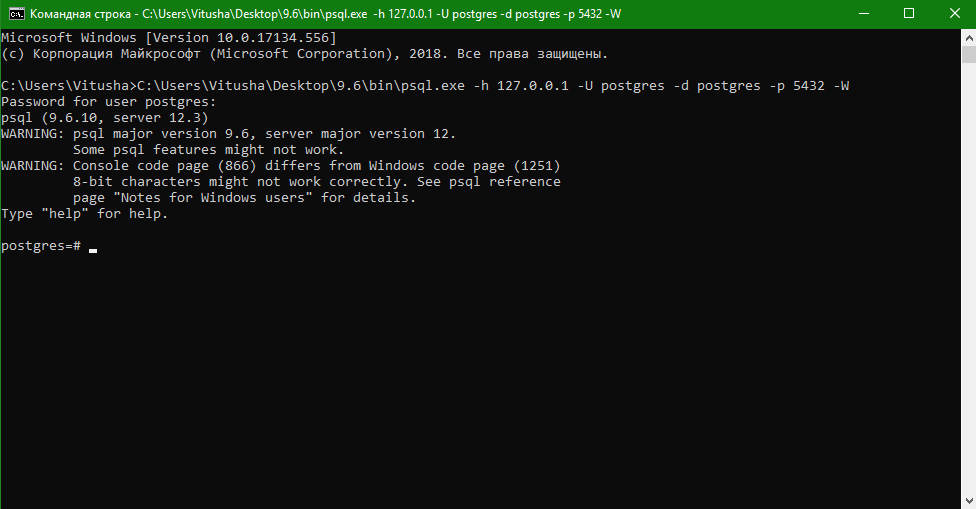
**\g** – это указание завершения с точкой-с-запятой для выполнения запроса. Иначе говоря, если хотим выполнить запрос, то нужно добавить точку-с-запятой (;) или \g в конце запроса.

**\q** – команда выхода из командной строки.

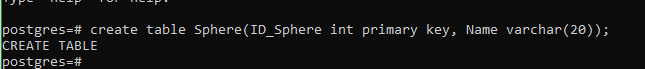
**PostgreSQL имеет следующие ограничения:**   

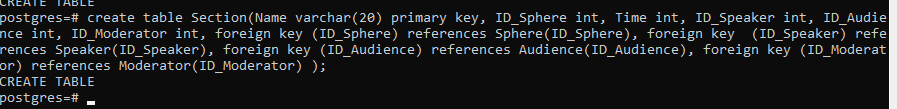

**Ход работы**

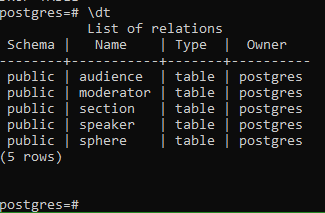
**Физическая ER модель**  
**Логическая ER модель**

Подключение к БД происходит через psql.exe , запускаемой в cmd.exe.   
  
  
Подключение к бд

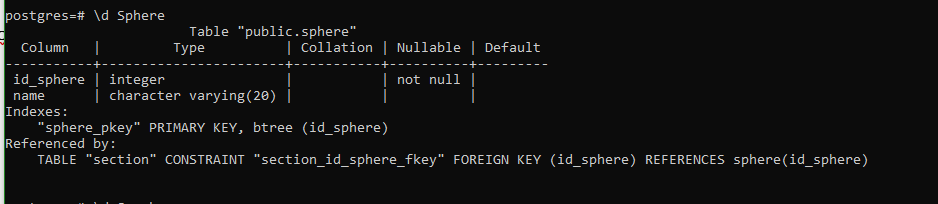
**Работа с таблицами.**

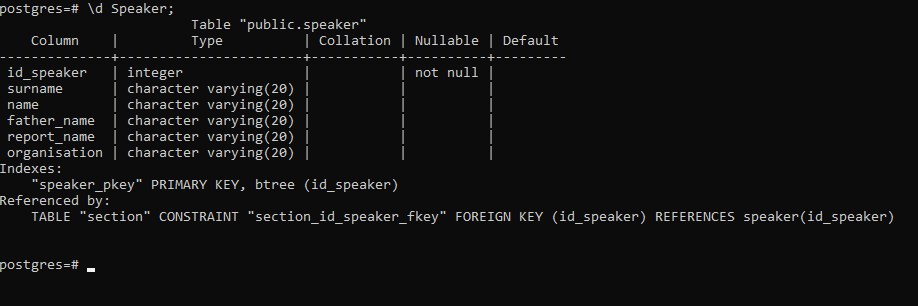
Для создания таблицы используется команда CREATE TABLE название\_таблицы(названия\_колонки тип\_данных (размер)).  


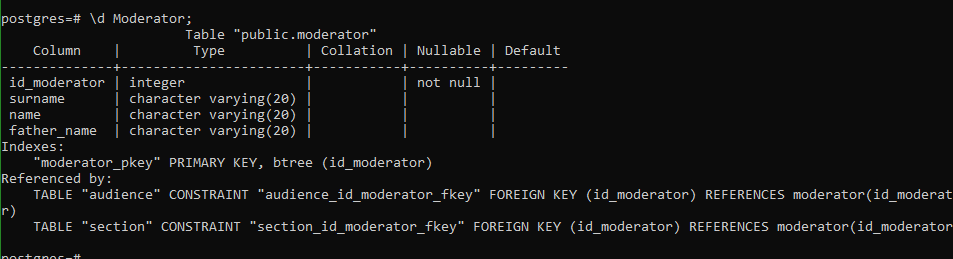
Создание таблицы Предметная область  
Создание таблицы Аудитория  
  
Создание таблицы Модератор  
  
Создание таблицы спикер  
  
Создание таблицы Секция

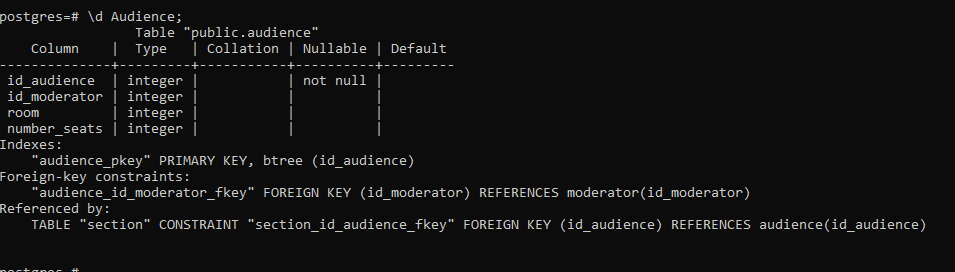
Для того, чтобы просмотреть список таблицы в базе данных, нужно использовать команду \dt.   


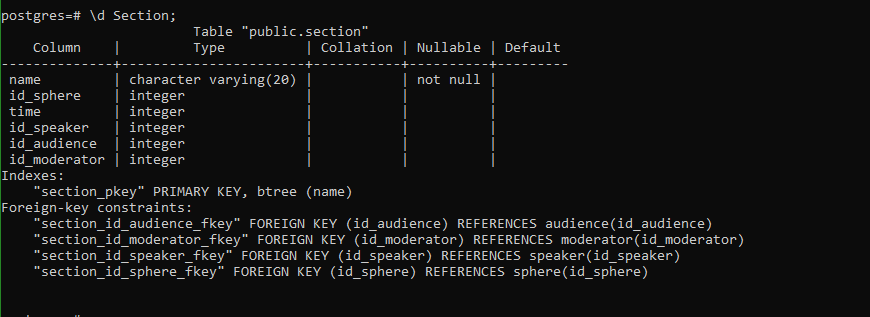
**Просмотр структуры таблицы.**

Для того, чтобы просмотреть, какие поля присутствуют в таблице, используется команда \d имя\_таблицы.   
  


Структура Таблицы Предметная область  
  
Структура Таблицы Спикер



Структура Таблицы Модератор  


Структура Таблицы Аудитория  
  
Структура таблицы Секция

**Команды для заполнения таблиц.**

· Для заполнения таблицы Moderator были использованы следующие команды:

Insert into Moderator values(1, ‘Денис’ ,’Сидоров’ , ‘Дмитириевич’ );

Insert into Moderator values(2, ‘Андрей’ ,’Иванов’ ,’Сергеевич’ );

Insert into Moderator values(3, ‘Валерий’ ,’Петров’ ,’Иванович’ );

Insert into Moderator values(4, ‘Дмитрий’, ‘Холопов’, ‘Олегович’);

Insert into Moderator values(4, ‘Дмитрий’ ,’Шишкин’, ‘Олегович’);

Insert into Moderator values(5, ‘Олег’ ,’Домов’ ,’Борисович’ );

Insert into Moderator values(6, ‘Александр’ ,’Печкин’ ,’Алекссевич’ );

Insert into Moderator values(7, ‘Алексей’ ,’Федосеев’ , ‘Иванович’);

Insert into Moderator values(8, ‘Борис’ ,’Филиппов’ ,’Андреевич’ );

Insert into Moderator values(9, ‘Иван’,’Капустин’ , ‘Денисович’);

Insert into Moderator values(10, ‘Андрей’ ,’Калкин’ ,’Дмитриевич’ );

· Для заполнения таблицы Audience были использованы следующие команды:

Insert into Audience values(1, 3 ,117 , 32 );

Insert into Audience values(2, 2 ,118 , 32 );

Insert into Audience values(3, 4 ,119 , 30 );

Insert into Audience values(4, 7 ,121 , 32 );

Insert into Audience values(5, 9 ,210 , 64 );

Insert into Audience values(6, 10 ,211 , 63 );

Insert into Audience values(7, 8 ,220 , 60 );

Insert into Audience values(8, 1 ,214 , 65 );

Insert into Audience values(9, 5 ,301 , 54 );

Insert into Audience values(10, 6 ,303 , 54 );

· Для заполнения таблицы Speaker были использованы следующие команды:

Insert into Speaker values(1, ‘Просин’ ,’Роман’ , ‘Андреевич’,’Code review’,’Райффайзенбанк’ );

Insert into Speaker values(2, ‘Колодяжный’ ,’ Павел’ , ‘Сергеевич’,’Акции и бренды’,’Альфа-банк’ );

Insert into Speaker values(3, ‘Баяндин ’ ,’Владимир’ , ‘Андреевич’,’Customer development’,’Skyeng’ );

Insert into Speaker values(4, ‘Гришенко’ ,’Егор’ , ‘Максимович’,’трассировка’,’Insolar’ );

Insert into Speaker values(5, ‘Панченко’ ,’Владимир’ , ‘Олегович’,’ статический анализ’,’Acumatica’ );

Insert into Speaker values(6, ‘Молчанов’ ,’Николай’ , ‘Андреевич’,’Мутационный анализ’,’JUG Ru Group’ );

Insert into Speaker values(7, ‘Скрыган’ ,’Кирилл’ , ‘Сергеевич’,’Platform Wars’,’JetBrains’ );

Insert into Speaker values(8, ‘Повар’ ,’Яков’ , ‘Денисович’,’Events sourcing’,’ PT’ );

Insert into Speaker values(9, ‘Дятлов’ ,’Андрей’ , ‘Андреевич’,’ references types’,’JetBrains’ );

Insert into Speaker values(10, ‘Богатов’ ,’Егор’ , ‘Дмитриевич’,’.NET5: runtimes’,’Microsoft’ );

· Для заполнения таблицы Sphere были использованы следующие команды:

Insert into Sphere values(1, ‘Компьютерные науки’);

Insert into Sphere values(2, ‘Биология’);

Insert into Sphere values(3, ‘Математика’);

Insert into Sphere values(4, ‘Бизнес’);

Insert into Sphere values(5, ‘Гуманитарные науки’);

**·**  Для заполнения таблицы Section были использованы следующие команды:

Insert into Section values(‘Райффайзенбанк’, 4 ,45 , 1, 10,1);

Insert into Section values(‘Альфа-банк’, 4 ,45 , 2, 9, 2);

Insert into Section values(‘SkyEng’, 4 ,45, 3, 8,3);

Insert into Section values(‘запросs’, 1 ,90 ,4, 7,4 );

Insert into Section values(‘статического анализ’, 3 ,90 ,5, 6,5);

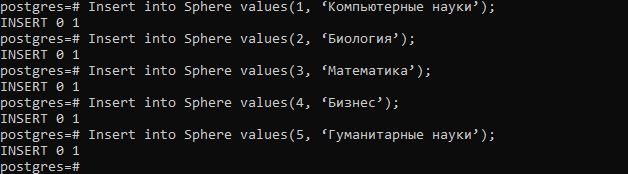
Insert into Section values(’Мутационный анализ’, 2 ,45 ,6, 5,6);

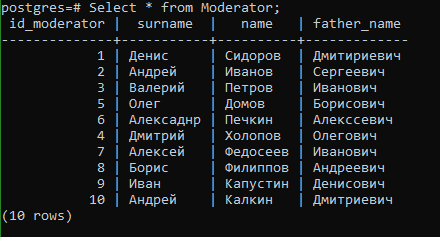
Insert into Section values(‘Platform\_wars’, 1 ,90 ,7, 4,7);

Insert into Section values(‘Event Sourcing’, 1,90 ,8, 3,8);

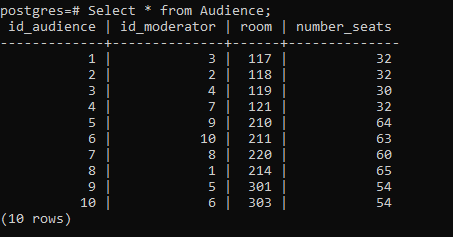
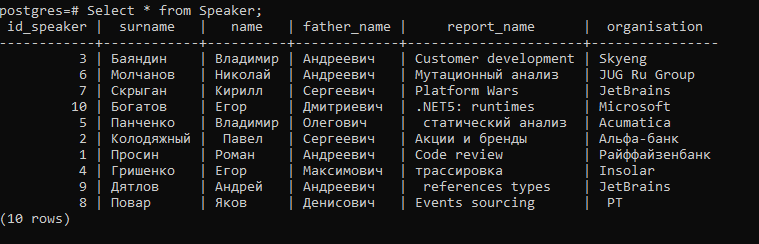
Insert into Section values(‘references types’, 3 ,90 ,9, 2,9);

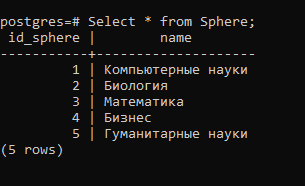
Insert into Section values(‘NET5’, 1 ,90 ,10, 10 );

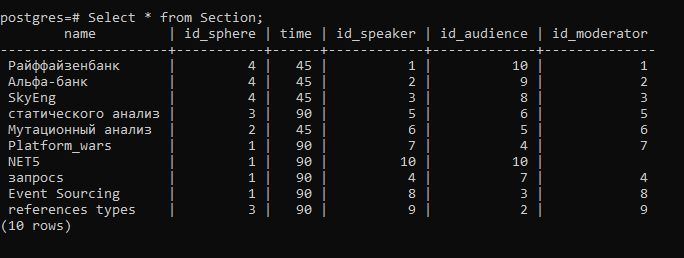
Пример заполнения таблицы  


Для просмотра данных таблицы используется команда SELECT  


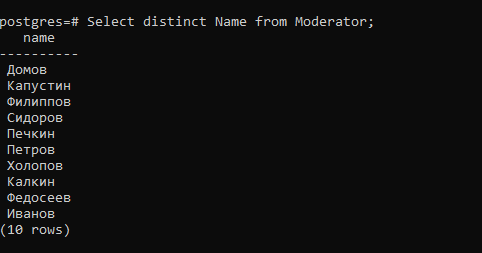
Просмотр таблицы Модератор

  
Просмотр таблицы Аудитория  
  
Просмотр таблицы Модератор Спикер

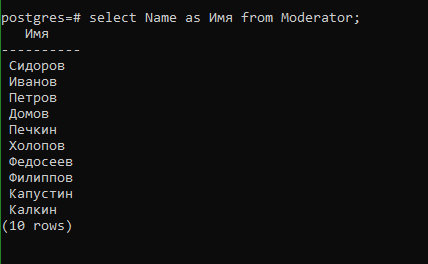
  
Просмотр таблицы Предметная область

  
Просмотр таблицы Секция

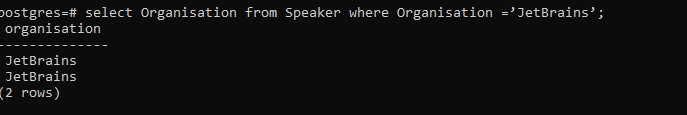
**Select distinct название\_колонки from название\_таблицы –** определяет поле или выражение, значения которого должны входить в итоговый набор не более одного раза.

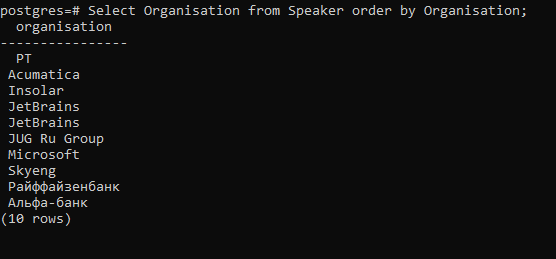


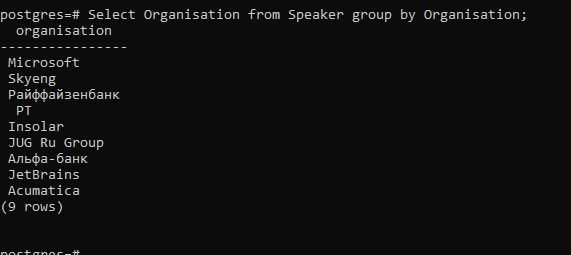
**Select колонка\_таблицы AS название from название\_таблицы –** используется для динамического переименования колонки таблицы.

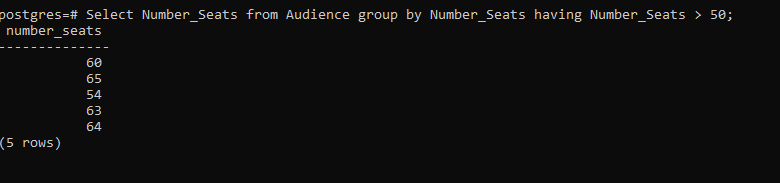


**Select название\_колонки from название\_таблицы where условие –** выбирает определённый элементы, удовлетворяющие условию.

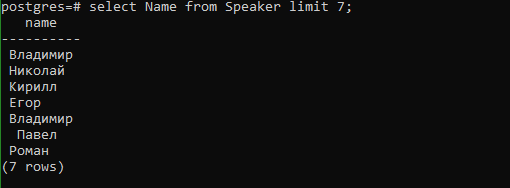


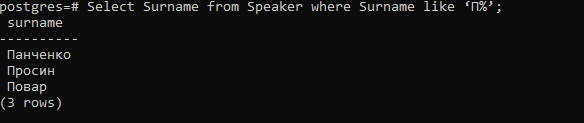
**Select название\_колонки from навзвание\_таблицы order by условие –** выбирает данные из таблицы и сортирует их.  


**Select название\_колонки from название\_таблицы group by критерий –** выводит данные и обеспечивает их группировку по заданному критерию.  


**Select название\_колонки from название таблицы having критерий –** проверяет условие на уровне целых групп, а не отдельных записей.  


**Delete from название\_таблицы –** удаляет записи из таблицы.  


**Select название\_колонки from название\_таблицы limit число –** выводит ограниченное количество записей.  


**Select название\_колонки from название\_таблицы where название\_колонки like шаблон –** ищет заданный шаблон в столбце.  


**Select название\_колонки from название\_таблицы JOIN … -** используется для соединения таблиц.  
