

Лабораторная работа № 6 по теме: «Работа с БД в СУБД PostgreSQL: создание БД, создание, удаление и заполнение таблиц»

Цель: знакомство с языком SQL, изучение способов создания и заполнения БД

Ход работы:

1. Оформите титульный лист
2. Выполните задания (по вариантам) и оформите отчет
3. Сдайте лабораторную работу преподавателю (на следующей паре)

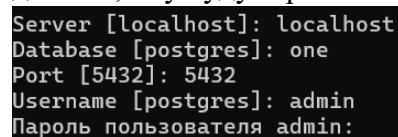
Отчет по лабораторной работе должен содержать (выполняется в MS Word):

1. ФИ, группа в верхнем колонтитуле на каждой странице, кроме первой
2. Первая страница – титульный лист
3. Каждая страница пронумерована, кроме первой
4. Выполненные упражнения и задания начинать со второй страницы документа (каждое упражнение и задание подписано, при необходимости присутствуют скриншоты) **P.S. Скриншоты делаются ВСЕГО экрана компьютера, а не только окна программы или строки кода!!!**

Ход работы.

Упражнение 1. Работа с интерактивным терминалом для администрирования СУБД PostgreSQL, создание роли, БД и таблицы

1. Запустите psql (SQL Shell). В открывшемся окне требуется указать соответствующие параметры (добавьте скриншот):
 - a. Server (адрес сервера) – localhost или 127.0.0.1;
 - b. Database (название базы данных) – postgres;
 - c. Port (номер порта) – 5432;
 - d. Username (имя пользователя) – postgres;
 - e. Password (пароль пользователя) – пароль, указанный в процессе установки СУБД PostgreSQL
2. После запуска psql будет открыта командная оболочка, в которой можно выполнять SQL запросы, также данная оболочка имеет набор встроенных команд. Введите help и получите справку.
3. Посмотрите справку по операторам SQL и по командам psql (добавьте скриншоты). Если столкнулись с ситуацией, что кириллица отображается некорректно, то введите команду **\! chcp 1251**
4. Создание новой роли. Если вы производили установку по инструкции, то к этому моменту в вашей СУБД есть только одна роль – postgres. Рекомендуется не использовать данную роль для работы со своими базами данных, а создавать для каждой базы новую роль (или несколько при необходимости). Для создания новой роли воспользуйтесь командой (не забудьте заменить username на желаемое имя пользователя, а password – на пароль для этого пользователя: **create user username with password 'password';** (Имя указывается без кавычек, а пароль – в одинарных кавычках). Создайте новую роль (скриншот).
5. Создание базы данных. Находясь в режиме командной строки psql, создать базу данных можно командой create database и указав название базы данных. Например, чтобы создать БД, выполните команду: **create database ONE;** (скриншот)
6. Для того, чтобы посмотреть все существующие БД необходимо выполнить команду **\L**
7. Назначение прав. Созданной ранее роли нужно назначить права на базу данных. В большинстве проектов, где у будет использоваться всего один пользователь базы данных, ему будут требоваться полные права. Выдать их можно следующим образом: **grant all privileges on database testing_db to username;** где testing_db –



```
Server [localhost]: localhost
Database [postgres]: one
Port [5432]: 5432
Username [postgres]: admin
Пароль пользователя admin:
```

название базы данных, выбранное на шаге 5, а username - имя пользователя, заданное на шаге 4(скриншот).

8. Выйдите из psql и войдите заново, используя новые параметры для входа. Теперь вы работаете с БД one;
9. Создадим новую таблицу: `CREATE TABLE job (id_p int primary key, job varchar(50));`
10. Произведем добавление записи в таблицу job. Для этого необходимо выполнить команду: **Insert INTO job (id_p, job) VALUES ('51', 'Doctor');**
11. Проверьте добавленную запись командой Select: **Select * from job;** (добавить скриншот выполнения, * - все записи)
12. Для выполнения задания необходимо гораздо больше записей, что приведет к определенной проблеме, так как выполнение отдельных команд вставки единственной записи (и так 50 раз) не является рациональным решением. Воспользуемся открытым ресурсом: для генерации <https://www.mockaroo.com/> 50 записей, подходящих под структуру созданной таблицы

The screenshot shows the Mockaroo website interface. At the top, there's a navigation bar with links: SCHEMAS, DATASETS, APIS, DATABASES (marked with a 'NEW' badge), SCENARIOS, and PROJECTS. Below the navigation bar, there's a dark-themed configuration area. It has a table with columns: Field Name, Type, and Options. The table contains two rows: 'id_p' with Type 'Row Number' and 'job' with Type 'Job Title'. Below the table, there are buttons: '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. At the bottom of the configuration area, there are input fields for '# Rows' (set to 50), 'Format' (set to SQL), 'Table Name' (set to person), and a checkbox for 'include CREATE TABLE statement'. Below this, there's a section titled 'Mock your back-end API and start coding your UI today.' with a paragraph of text and a 'Why is test data important?' section. At the very bottom, there are three buttons: 'GENERATE DATA', 'PREVIEW', and 'SAVE AS...'. The 'GENERATE DATA' button is highlighted with a green border.

13. Сгенерируйте файл и сохраните его себе в формате .sql
14. После этого в консоли psql необходимо вызвать команду \i FILE. Команда выполняет инструкции SQL, содержащиеся в файле FILE. В данном случае необходимо выполнить следующую команду: `\i d:/job.sql` (укажите свое расположение скачанного файла)
15. Осуществим выборку первых десяти записей из таблицы job, предварительно упорядочив их по полю job: **select * from job limit 10;** (добавить скриншот выполнения, * - все записи)

16. Также осуществим выборку 10 записей, но изменим немного запрос: **select * from job order by job limit 10;** (сделайте скриншот). В чем разница между 2мя запросами? Для чего предназначен **order by** (напишите ответ)?
17. Выполните команды **\d, \dt, \ds и \d job;** добавьте скриншоты и опишите, для чего предназначены данные команды.
18. Создадим новую таблицу: **CREATE TABLE person (id BIGSERIAL PRIMARY KEY, FIO VARCHAR(50) NOT NULL, gender VARCHAR(8) NOT NULL, bdate DATE, FK_job INT, foreign key(FK_job) references job(id_p));** (**P.S. Bigserial** — это тип данных в базе данных PostgreSQL, используемый для автоматической генерации уникальных идентификаторов, , **foreign key** (....) **references** (...) устанавливает связь с таблице job через внешний ключ).
19. Произведем добавление записи в таблицу person. Для этого необходимо выполнить команду: **Insert INTO person (first_name, last_name, gender, bdate, FK_job)Values ('ivan', 'petrov', 'male', '2000-12-12', '5');**
20. Проверьте добавленную запись командой **Select: Select * from person;** (добавить скриншот выполнения).
21. Добавьте еще 10 записей в таблицу person любым способом и выведите их на экран (скриншот).
22. Выполните операцию проекции по столбцу last_name, отсортируйте по возрастанию (вместо * использовать название столбца, скриншот)
23. Выполните операцию проекции по столбцу first_name, отсортируйте по убыванию (как сделать по убыванию найдите самостоятельно, скриншот)
24. Теперь выведем всех людей, сопоставляя их с профессиями (связь со второй таблицей через внешний ключ): **select * from person inner join job on person.fk_job = job.id_p;** (скриншот)
25. Выведите всех людей (имя, фамилию) профессии 'doctor' (или любой другой из вашей базы, чтобы выводилась хотя бы одна запись):
select person.first_name, person.last_name, job.job from person inner join job on person.fk_job = job.id_p where job='Doctor'; (скриншот)
26. Удалим созданные таблицы и БД (скриншот):
drop table person; (удаление таблицы)
drop database one; (удаление БД идет под тем пользователем (ролью), под которым создавался)

Задание 1. (Каждый шаг сопровождать скриншотами и словесным описанием сделанного)

Создайте новую БД.

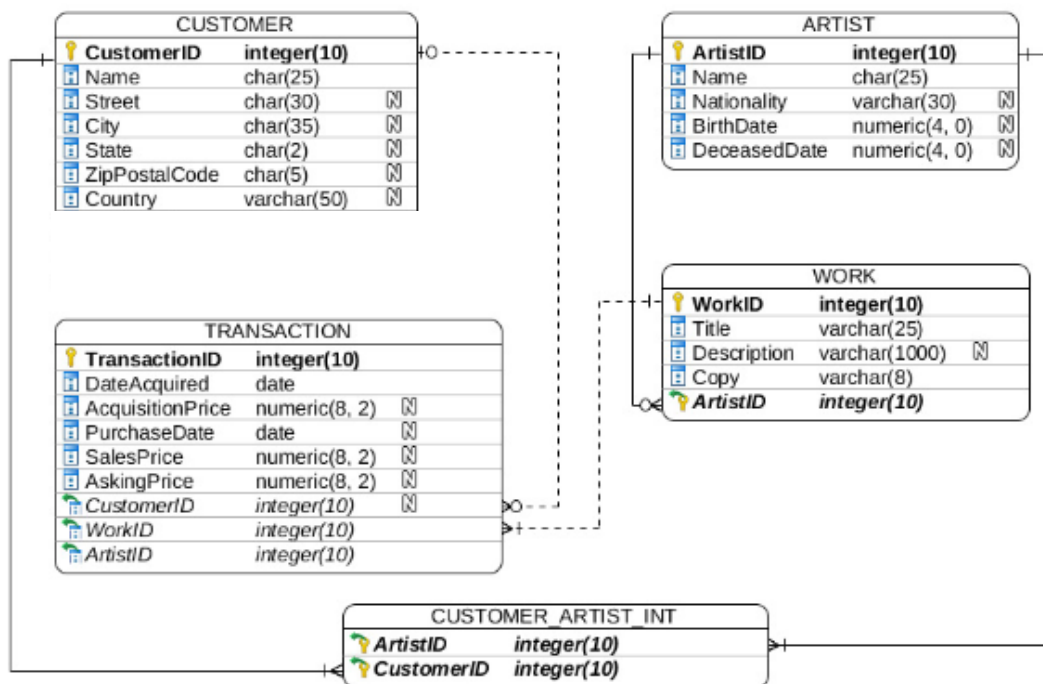
Создайте таблицы, используя следующую физическую модель (после сдачи работы не удалять данную базу)

P.S. Не забываем при создании добавлять ограничения первичного ключа, внешнего ключа, not null и unique при необходимости.

Заполните таблицы не менее, чем 30 записями любыми способами (скриншоты заполненных таблиц приложить (расположены в алфавитном порядке)).

Выполните следующие запросы:

1. Вывести Имена и национальности всех художников
2. Вывести имена всех художников определенной национальности
3. Вывести Имя всех покупателей с датами покупки
4. Вывести название всех работ(картин) и их стоимость



Критерии оценки:

| | |
|-------------------|---|
| Зачтено | Все упражнения и задания лабораторной работы выполнены самостоятельно, студент может объяснить, как он выполнил то или иное действие; отчет содержит все необходимые сведения, студент уверенно отвечает на вопросы по теории |
| Не зачтено | Лабораторная работа выполнена не самостоятельно (студент не может ответить ни на один вопрос преподавателя), или выполнена не до конца, или не выполнена совсем |