

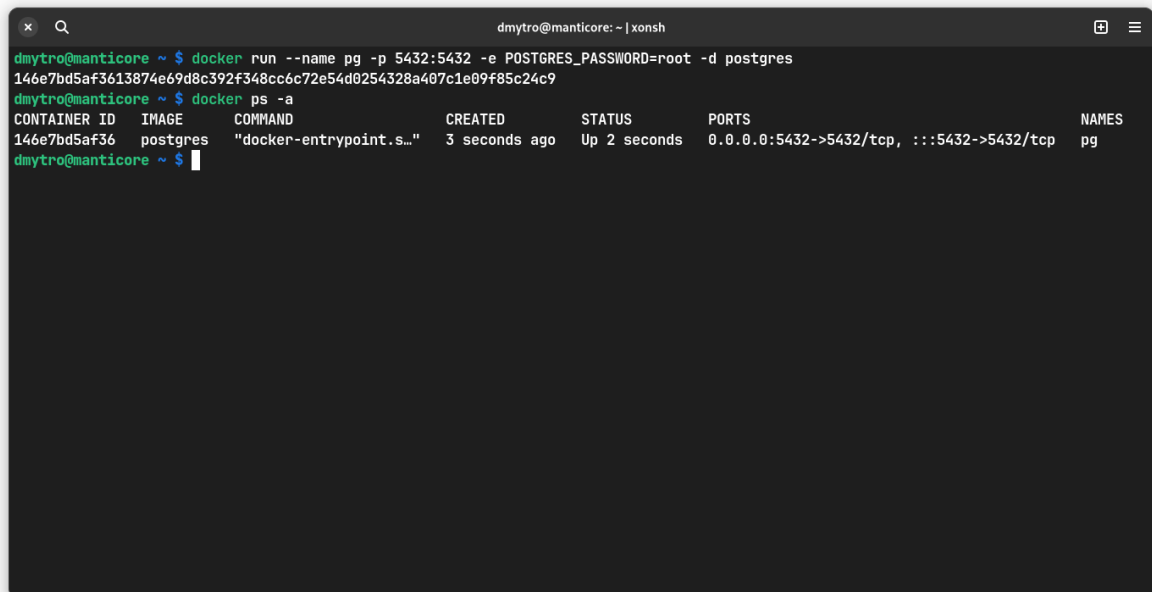
Лабораторна робота 7-8

Завдання роботи

Завдання лабораторної роботи виконав на 74%

Встановіть PostgreSQL (<https://www.postgresql.org/download/>) .

Для швидкого розгортання бази даних для виконання практичної роботи я використав Docker:



```
dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ docker run --name pg -p 5432:5432 -e POSTGRES_PASSWORD=root -d postgres
146e7bd5af3613874e69d8c392f348cc6c72e54d0254328a407c1e09f85c24c9
dmytro@manticore ~ $ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
146e7bd5af36   postgres "docker-entrypoint.s..." 3 seconds ago  Up 2 seconds  0.0.0.0:5432->5432/tcp, :::5432->5432/tcp  pg
dmytro@manticore ~ $
```

Приєднайтесь до СКБД із допомогою консольного клієнта та створіть нову БД.

У командному рядку:

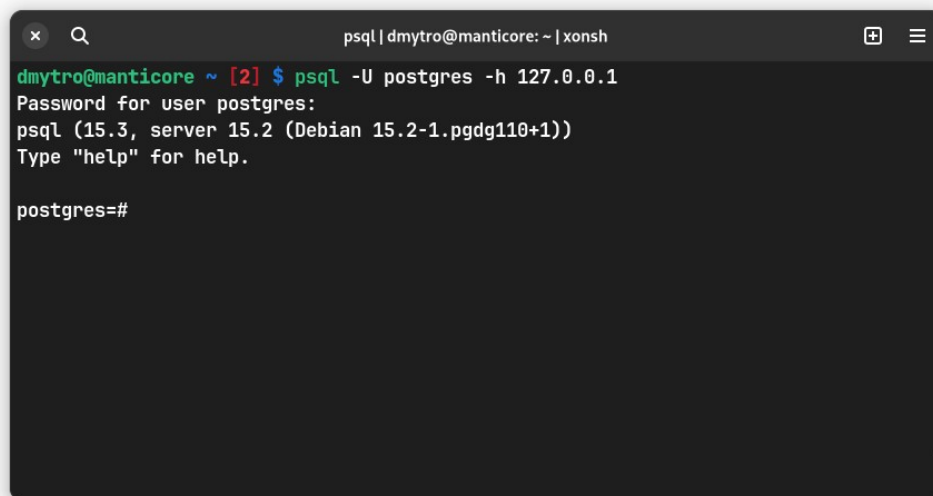
psql.exe -U userName

у psql:

CREATE DATABASE student_KB81_01_production

\q

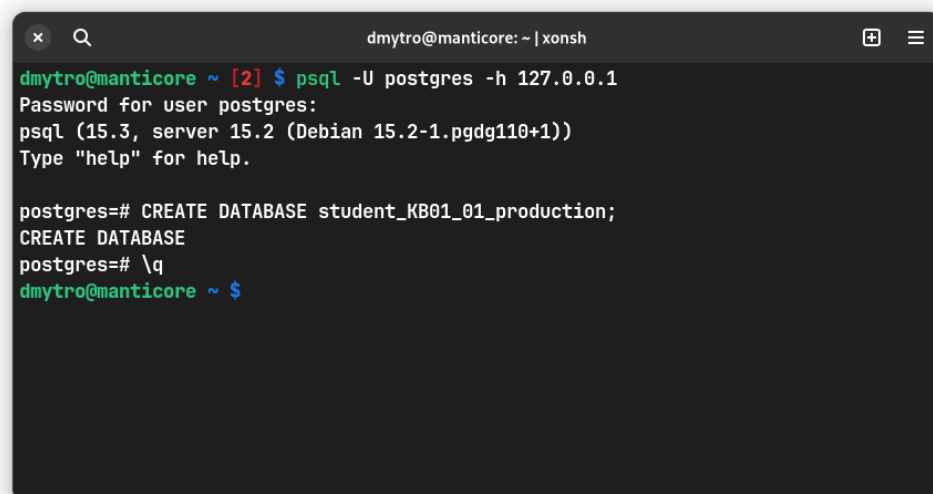
Підключення до бази даних:

A terminal window titled 'psql | dmytro@manticore: ~ | xonsh'. The prompt is 'dmytro@manticore ~ [2] \$'. The user enters 'psql -U postgres -h 127.0.0.1'. The terminal shows the password prompt, the psql version and server information, and the prompt 'postgres=#'.

```
dmytro@manticore ~ [2] $ psql -U postgres -h 127.0.0.1
Password for user postgres:
psql (15.3, server 15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

postgres=#
```

Створення бази даних та вихід:

A terminal window titled 'dmytro@manticore: ~ | xonsh'. The prompt is 'dmytro@manticore ~ [2] \$'. The user enters 'psql -U postgres -h 127.0.0.1'. The terminal shows the password prompt, the psql version and server information, and the prompt 'postgres=#'. The user enters 'CREATE DATABASE student_KB01_01_production;', 'CREATE DATABASE', and '\q'. The terminal shows the prompt 'dmytro@manticore ~ \$' at the end.

```
dmytro@manticore ~ [2] $ psql -U postgres -h 127.0.0.1
Password for user postgres:
psql (15.3, server 15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

postgres=# CREATE DATABASE student_KB01_01_production;
CREATE DATABASE
postgres=# \q
dmytro@manticore ~ $
```

Модифікуйте сценарій створення таблиць Вашої курсової роботи, так щоб його можна було застосувати у PostgreSQL. У більшості випадків все спрацює, якщо ви прибереже із сценарію все зайве (параметри створення таблиць тощо). Невеликі складнощі можуть виникнути із типами даних, але всі основні типи даних – підтримуються.

Завантажте сценарій до вашої БД

Завантаження сценарію:

[illegible]

Приєднайтесь до бази та перевірте, що все пройшло вдало (“\q” – для виходу):

Перевірка записів:

```
dmytro@manticore: ~/Study/DB/Semester_2/Lab_3 | xonsh
dmytro@manticore ~/Study/DB/Semester_2/Lab_3 main $ psql -U postgres -h 127.0.0.1 -d student_KB01_01_production
Password for user postgres:
psql (15.3, server 15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

student_KB01_01_production=# select * from "Cars";
 car_id |      name      | number_plate | color  | rent_rate
-----+-----+-----+-----+-----
      1 | Lanos          | AA 7777 AA  | black  |         75
      2 | VW Passat      | AA 7778 AA  | red    |        275
      3 | ZAZ            | AA 7779 AA  | green  |         45
      4 | Skoda Octavia  | AA 7780 AA  | blue   |         23
      5 | Ford Transit   | AA 7781 AA  | black  |         67
      6 | Renault Logan  | AA 7782 AA  | purple |         94
      7 | BMW M3         | AA 7783 AA  | yellow |        234
      8 | Chevrolet Aveo | AA 7784 AA  | green  |         47
      9 | GMC            | AA 7785 AA  | green  |         32
     10 | Nissan Skyline GT R32 | AA 7786 AA  | black  |         19
     11 | Honda Civic    | AA 7787 AA  | red    |         93
(11 rows)

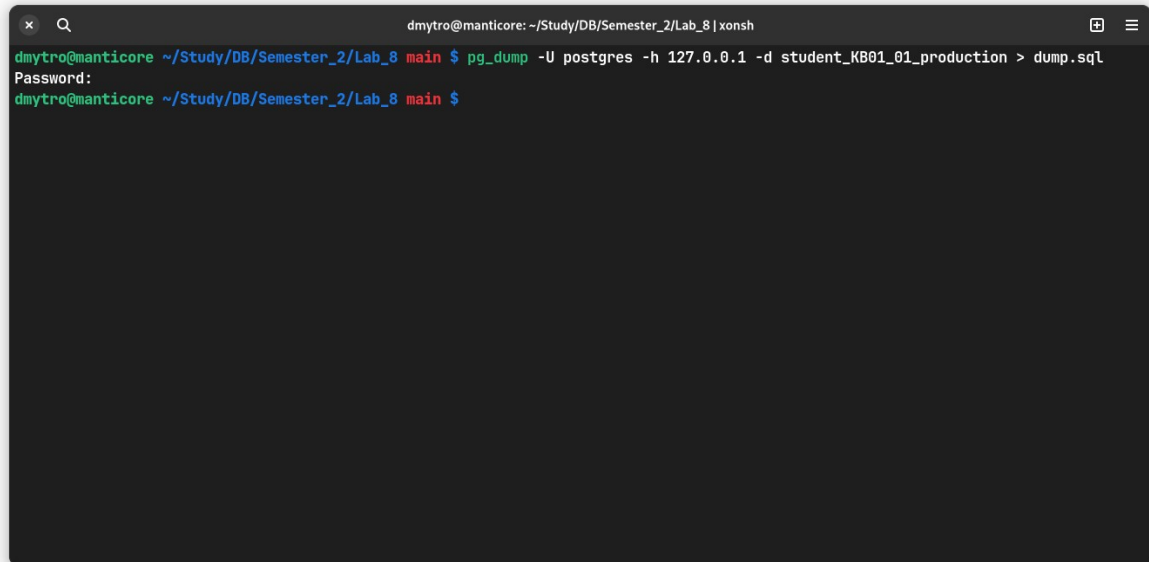
student_KB01_01_production=#
\q
dmytro@manticore ~/Study/DB/Semester_2/Lab_3 main $
```

60%

Зробіть резервну копію БД із використанням `pg_dump` (деталі використання `pg_dump --help`, опції аналогічні `psql`, але треба вказати куди зробити резервну копію)

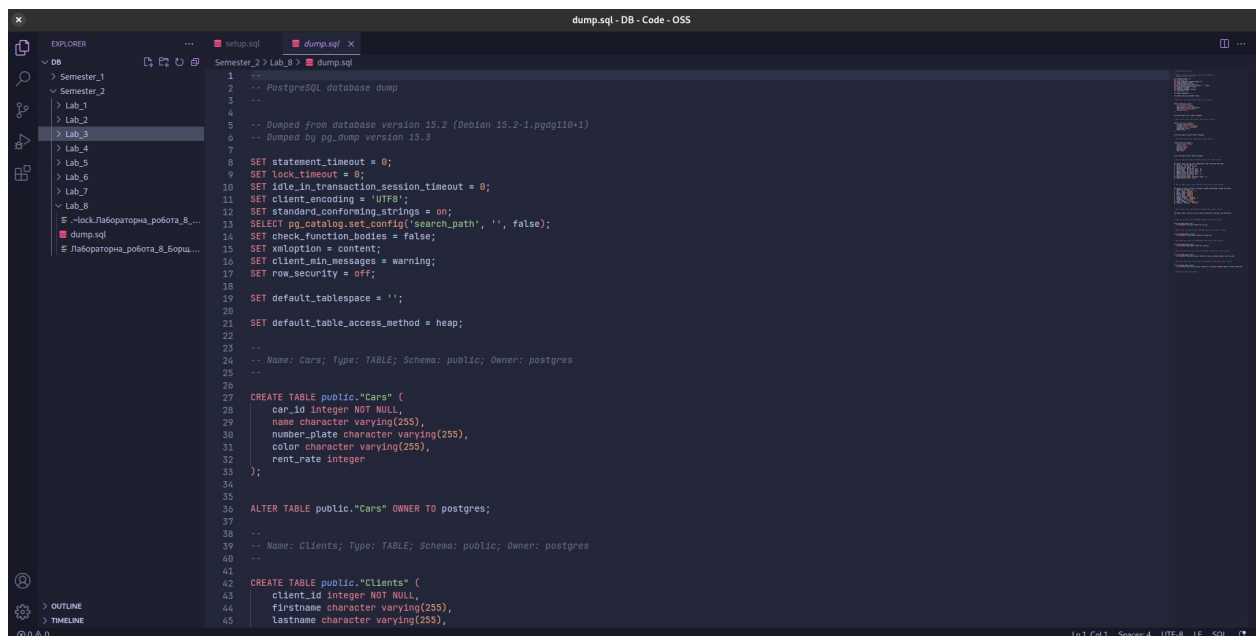
Наведіть результат у звіті.

Створення дампу:



```
dmytro@manticore: ~/Study/DB/Semester_2/Lab_8 | xonsh
dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main $ pg_dump -U postgres -h 127.0.0.1 -d student_KB01_01_production > dump.sql
Password:
dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main $
```

Відкривши файл можемо побачити, що це фактично готовий скрипт для того щоб відтворити аналогічну базу шляхом його виконання:



```
dump.sql - DB - OSS
1 --
2 -- PostgreSQL database dump
3 --
4 --
5 -- Dumped from database version 15.2 (Debian 15.2-1.pgdg110+1)
6 -- Dumped by pg_dump version 15.3
7
8 SET statement_timeout = 0;
9 SET lock_timeout = 0;
10 SET idle_in_transaction_session_timeout = 0;
11 SET client_encoding = 'UTF8';
12 SET standard_conforming_strings = on;
13 SELECT pg_catalog.set_config('search_path', '', false);
14 SET check_function_bodies = false;
15 SET xmloption = content;
16 SET client_min_messages = warning;
17 SET row_security = off;
18
19 SET default_tablespace = '';
20
21 SET default_table_access_method = heap;
22
23 --
24 -- Name: Cars; Type: TABLE; Schema: public; Owner: postgres
25 --
26
27 CREATE TABLE public."Cars" (
28     car_id integer NOT NULL,
29     name character varying(255),
30     number_plate character varying(255),
31     color character varying(255),
32     rent_rate integer
33 );
34
35 ALTER TABLE public."Cars" OWNER TO postgres;
36
37 --
38 -- Name: Clients; Type: TABLE; Schema: public; Owner: postgres
39 --
40
41 CREATE TABLE public."Clients" (
42     client_id integer NOT NULL,
43     firstname character varying(255),
44     lastname character varying(255),
45
```

64%

Дослідить різницю між варіантами опції `--format=c|d|t|p` утиліти `pg_dump`.

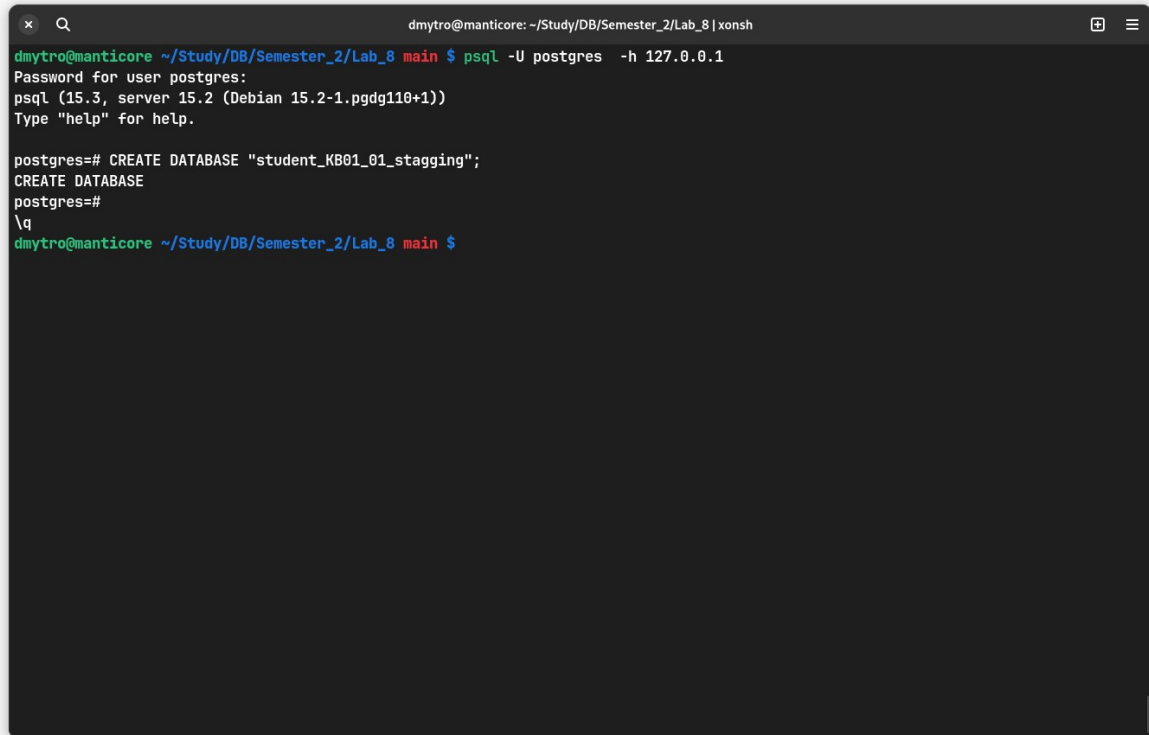
Вибирає формат виведення. Формат може бути одним із таких:

- `p, plain` - Вивести файл сценарію SQL у вигляді звичайного тексту (за замовчуванням);
- `c, custom` - Вивести архів спеціального формату, придатний для введення в `pg_restore`. Разом із форматом виведення каталогу, це найбільш гнучкий формат виведення, оскільки він дозволяє вручну вибирати та змінювати порядок архівованих елементів під час відновлення. Цей формат також стискається за замовчуванням;
- `d, directory` - Вивести архів у форматі каталогу, придатний для введення в `pg_restore`. Це створить каталог з одним файлом для кожної таблиці та великого об'єкта, що вивантажується, а також так званий файл змісту, який описує викинуті об'єкти в машиночитаному форматі, який може читати `pg_restore`. Архівом формату каталогу можна керувати стандартними інструментами Unix; наприклад, файли в нестисненому архіві можна стиснути за допомогою інструменту `gzip`. Цей формат стискається за замовчуванням, а також підтримує паралельні дампи;
- `t, tar` - Виведіть архів у форматі `tar`, придатний для введення в `pg_restore`. Формат `tar` сумісний із форматом каталогу: розпакування архіву у форматі `tar` створює дійсний архів у форматі каталогу. Однак формат `tar` не підтримує стиснення. Крім того, при використанні формату `tar` відносний порядок елементів таблиці даних не можна змінити під час відновлення.

74%

Створіть нову базу даних student_KB81_01_stagging з резервної копії бази student_KB81_01_production . (Підійде утіліта psql для тексових форматів, або pg_restore для бінарних).

Створення нової БД:

A terminal window titled 'dmytro@manticore: ~/Study/DB/Semester_2/Lab_8 | xonsh'. The user runs the command 'psql -U postgres -h 127.0.0.1'. The terminal shows the password prompt for the 'postgres' user, followed by the psql prompt. The user enters the SQL command 'CREATE DATABASE "student_KB81_01_stagging";'. The terminal displays the output 'CREATE DATABASE' and the psql prompt. The user then enters the backslash command '\q' to exit psql. The terminal returns to the shell prompt 'dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main \$'.

Відтворення бази даних з дампу:

```
dmytro@manticore: ~/Study/DB/Semester_2/Lab_8 | xonsh
dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main $ psql -U postgres -h 127.0.0.1 -d student_KB01_01_stagging < dump.sql
Password for user postgres:
SET
SET
SET
SET
SET
set_config
-----
(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
COPY 11
COPY 11
COPY 0
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main $
```

Перевірка наявності таблиць і наповнення:

```
psql | dmytro@manticore: ~/Study/DB/Semester_2/Lab_8 | xonsh
dmytro@manticore ~/Study/DB/Semester_2/Lab_8 main $ psql -U postgres -h 127.0.0.1 -d student_KB01_01_stagging
Password for user postgres:
psql (15.3, server 15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

student_KB01_01_stagging=# \dt
      List of relations
Schema | Name   | Type  | Owner
-----+-----+-----+-----
public | Cars   | table | postgres
public | Clients| table | postgres
public | Rents  | table | postgres
(3 rows)

student_KB01_01_stagging=# select * from "Cars";
 car_id | name           | number_plate | color  | rent_rate
-----+-----+-----+-----+-----
      1 | Lanos          | AA 7777 AA   | black  |         75
      2 | VW Passat      | AA 7778 AA   | red    |        275
      3 | ZAZ            | AA 7779 AA   | green  |         45
      4 | Skoda Octavia  | AA 7780 AA   | blue   |         23
      5 | Ford Transit   | AA 7781 AA   | black  |         67
      6 | Renault Logan  | AA 7782 AA   | purple |         94
      7 | BMW M3         | AA 7783 AA   | yellow |        234
      8 | Chevrolet Aveo | AA 7784 AA   | green  |         47
      9 | GMC            | AA 7785 AA   | green  |         32
     10 | Nissan Skyline GT R32 | AA 7786 AA   | black  |         19
     11 | Honda Civic    | AA 7787 AA   | red    |         93
(11 rows)

student_KB01_01_stagging=#
```