МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра вычислительной техники



**Лабораторная работа №1**

**по дисциплине «Базы данных»**

**ОЗНАКОМЛЕНИЕ С ОСНОВАМИ POSTGRESQL**

|  |  |
| --- | --- |
| Студент | Ванин К.Е. |
| Вариант | 4 |
| Группа | АВТ-819 |
| Преподаватель | Бычков М.И. |
| Дата сдачи | 27.09.2020 |

Новосибирск

2020 г.

1. **Цель**

Изучить базовые операции по работе с базой данных. Изучить синтаксис команд. Приобрести навыки создания баз данных, создания, заполнения и модификации таблиц в PostgreSQL. Приобрести навыки работы с транзакциями, с ограничениями.

1. **Постановка задачи**

*Общая часть:* Создать и заполнить базу данных своего варианта в PostgreSQL. Таблицы (минимум по 10 записей в каждой) связать между собой полями идентификаторов. С помощью команд интерактивного терминала *psql* просмотреть структуру базы данных, структуру таблиц, просмотреть данные в них, изменить структуру таблиц, добавить столбцы, добавить данные, создать столбцы с пользовательскими типами данных. Ознакомиться с теоретическими сведениями о возможностях создания пользователей баз данных, использования транзакций в PostgreSQL. Создать нового пользователя и зайти под его именем. Создать и заполнить базу данных своего варианта. Таблицы (минимум по 10 записей в каждой) связать между собой полями идентификаторов. С помощью команд интерактивного терминала *psql* просмотреть структуру базы данных, структуру таблиц, просмотреть данные в них. Ознакомиться с теоретическими сведениями о возможностях создания ограничений в PostgreSQL. Наложить ограничения согласно своему варианту. Проверить работоспособность ограничений путем добавления в таблицы данных, удовлетворяющих и не удовлетворяющих условиям ограничений. Создать транзакционный блок, в котором производится добавление в таблицы произвольных полей. Не завершая транзакции параллельно запустить еще одно окно терминала, подключиться к базе и попробовать добавить и удалить записи в таблицы. Сделать откат транзакций, просмотреть структуру таблиц.

*Вариант 4*: Создать и заполнить базу данных для обработки данных по работе книжной лавки, состоящую из четырех таблиц. Первая таблица должна содержать поля: идентификатор\_жанра, наименование жанра, ‘. Вторая: идентификатор\_книги, название книги, автор книги и другие поля при необходимости. Третья: идентификатор\_издательства, наименование издателя, адрес издательства и другие поля при необходимости. Четвертая: идентификатор\_страны, наименование страны. На основании созданных таблиц создать таблицу, содержащую, например, поля: название книги, автор книги, жанр, издательство, количество страниц. Ограничить заполнение поля «жанр» следующими наименованиями: детектив, роман, повесть, рассказ, пьеса, новелла. На поле «стоимость» наложить ограничение таким образом, чтобы нельзя было ввести цену ниже 50 р.

1. **Выполнение работы**

-- Создаем пользователя vanin и БД dblab1

Server [localhost]:

Database [postgres]:

Port [5432]:

Username [postgres]:

Пароль пользователя postgres:

psql (10.14)

ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной

страницы Windows (1251).

8-битовые (русские) символы могут отображаться некорректно.

Подробнее об этом смотрите документацию psql, раздел

"Notes for Windows users".

Введите "help", чтобы получить справку.

postgres=# \! chcp 1251

postgres=# create user vanin createdb password '12345';

CREATE ROLE

postgres=# create database dblab1;

CREATE DATABASE

postgres=#

-- Выходим, заходим заново под созданным пользователем

Server [localhost]:

Database [postgres]: dblab1

Port [5432]:

Username [postgres]: vanin

Пароль пользователя lab1:

psql (10.14)

ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной

страницы Windows (1251).

8-битовые (русские) символы могут отображаться некорректно.

Подробнее об этом смотрите документацию psql, раздел

"Notes for Windows users".

Введите "help", чтобы получить справку.

dblab1=> \! chcp 1251

Текущая кодовая страница: 1251

dblab1=>

-- Создадим перечисляемый тип для поля " gg " чтобы ограничить название жанра

dblab1=> create type gg as enum ('detective', 'novel', 'tale', 'story', 'play', 'romance');

CREATE TYPE

-- Создаем первую таблицу genre

-- Сразу устанавливаем ограничение: price

dblab1=> create table genre(genre\_id int, genre\_name gg, type\_of\_publication char(50), number\_of\_page int, price int check(price>=50), test char(20));

CREATE TABLE

-- Посмотрим, что получилось

dblab1=> select \* from genre;

genre\_id | genre\_name | type\_of\_publication | number\_of\_page | price | test

----------+------------+---------------------+----------------+-------+------

(0 строк)

-- Удалим поле test

dblab1=> alter table genre drop test;

ALTER TABLE

-- Посмотрим, что получилось

dblab1=> select \* from genre;

genre\_id | genre\_name | type\_of\_publication | number\_of\_page | price

----------+------------+---------------------+----------------+-------

(0 строк)

-- Заполним таблицу

dblab1=> insert into genre values (1, 'romance', 'Education', 350, '1002');

INSERT 0 1

dblab1=> insert into genre values (2, 'story', 'Publishing Center "Academy"', 249, '1350');

INSERT 0 1

dblab1=> insert into genre values (3, 'romance', 'Publishing house "Eksmo"', 120, '750');

INSERT 0 1

dblab1=> insert into genre values (4, 'story', 'Education', 420, '2456');

INSERT 0 1

dblab1=> insert into genre values (5, 'play', 'Publishing Center "Academy"', 367, '1125');

INSERT 0 1

dblab1=> insert into genre values (6, 'novel', 'Publishing group "AST"', 560, '2500');

INSERT 0 1

dblab1=> insert into genre values (7, 'detective', 'Publishing group "AST"', 135, '980');

INSERT 0 1

dblab1=> insert into genre values (8, 'play', 'Education', 450, '1500');

INSERT 0 1

dblab1=> insert into genre values (9, 'novel', 'Publishing Center "Academy"', 670, '5600');

INSERT 0 1

dblab1=> insert into genre values ('10', 'tale', 'Publishing house "Eksmo"', 320, '2760');

INSERT 0 1

-- Проверка созданного типа genre\_name на ввод некорректного значения

dblab1=> insert into genre values (2,'sdcgvgs', 'best', '20', '50');

ОШИБКА: неверное значение для перечисления gg: "sdcgvgs"

СТРОКА 1: insert into genre values (2,'sdcgvgs', 'best', '20', '50');

-- Проверка на выполнение заданного условия на цену: условие работает

dblab1=> ОШИБКА: новая строка в отношении "genre" нарушает ограничение-проверку "genre\_price\_check"

ПОДРОБНОСТИ: Ошибочная строка содержит (1, detective, best, 20, 2).

-- Посмотрим, что получилось

dblab1=> select \* from genre;

genre\_id | genre\_name | type\_of\_publication | number\_of\_page | price

----------+------------+--------------------------------+----------------+-------

1 | romance | Education | 350 | 1002

2 | story | Publishing Center "Academy" | 249 | 1350

3 | romance | Publishing house "Eksmo" | 120 | 750

4 | story | Education | 420 | 2456

5 | play | Publishing Center "Academy" | 367 | 1125

6 | novel | Publishing group "AST" | 560 | 2500

7 | detective | Publishing group "AST" | 135 | 980

8 | play | Education | 450 | 1500

9 | novel | Publishing Center "Academy" | 670 | 5600

10 | tale | Publishing house "Eksmo" | 320 | 2760

(10 строк)

-- Создаем вторую таблицу books2

dblab1=> create table books2(book\_id int, book\_name char(50), after\_book’s char(50));

CREATE TABLE

-- Посмотрим, что получилось

dblab1=> select \*from books2;

book\_id | book\_name | after\_book’s

---------+-----------+--------------

(0 строк)

-- Заполним таблицу

dblab1=> insert into books2 values (1, 'Harry Potter', 'Ivanov');

INSERT 0 1

dblab1=> insert into books2 values (2, 'Airport', 'Arthur Hailey');

INSERT 0 1

dblab1=> insert into books2 values (3, ' The Adventures of Sherlock Holmes', 'Ivanov');

INSERT 0 1

dblab1=> insert into books2 values (4, 'Three Men in a Boat', 'J.K. Rowling');

INSERT 0 1

dblab1=> insert into books2 values (5, 'Jane Eyre', 'Ivanov');

INSERT 0 1

dblab1=> insert into books2 values (6, 'Bridget Jones', 'Helen Fielding');

INSERT 0 1

dblab1=> insert into books2 values (7, 'To Kill a Mockingbird', 'Harper Lee');

INSERT 0 1

dblab1=> insert into books2 values (8, 'Hitchhiker Guide to the Galaxy', 'Douglas Adams');

INSERT 0 1

dblab1=> insert into books2 values (9, 'White Fang', 'Jack London');

INSERT 0 1

dblab1=> insert into books2 values (10, 'Martin Eden', 'Jack London');

INSERT 0 1

^

-- Посмотрим, что получилось

dblab1=> select \* from books2;

book\_id | book\_name | after\_book’s

---------+----------------------------------------------------+----------------------------------------------------

1 | Harry Potter | Ivanov

2 | Airport | Arthur Hailey

3 | The Adventures of Sherlock Holmes | Ivanov

4 | Three Men in a Boat | J.K. Rowling

5 | Jane Eyre | Ivanov

6 | Bridget Jones | Helen Fielding

7 | To Kill a Mockingbird | Harper Lee

8 | Hitchhiker Guide to the Galaxy | Douglas Adams

9 | White Fang | Jack London

10 | Martin Eden | Jack London

(10 строк)

-- Создаем третью таблицу books3

dblab1=> create table books3(publishing\_id int, publishing\_name char(50), publisher\_address\_streets char(50), publisher\_address\_city char(50));

CREATE TABLE

-- Посмотрим, что получилось

dblab1=> select \* from books3;

publishing\_id | publishing\_name | publisher\_address\_streets | publisher\_address\_city

---------------+-----------------+---------------------------+------------------------

(0 строк)

-- Заполним таблицу

dblab1=> insert into books3 values (1, '"REFL-book". Kiev. "ISA". 1994', 'Champs Elysees 36', 'St. Petersburg');

INSERT 0 1

dblab1=> insert into books3 values (2, '"The science". St. Petersburg branch', 'Wall Street 152', 'London');

INSERT 0 1

dblab1=> insert into books3 values (3, '"Centerpolygraph"', 'Abby Road 451', 'Moscow');

INSERT 0 1

dblab1=> insert into books3 values (4, '"Olma - Media Group"', 'Lombard Street 55', 'London');

INSERT 0 1

dblab1=> insert into books3 values (5, '"Centerpolygraph"', 'Abby Road 451', 'St. Petersburg');

INSERT 0 1

dblab1=> insert into books3 values (6, '"The science". St. Petersburg branch', 'Wall Street 152', 'New York');

INSERT 0 1

dblab1=> insert into books3 values (7, '"REFL-book". Kiev. "ISA". 1994', 'Champs Elysees 36', 'Novosibirsk');

INSERT 0 1

dblab1=> insert into books3 values (8, '"Olma - Media Group"', 'Lombard-Street 55', 'Tokyo');

INSERT 0 1

dblab1=> insert into books3 values (9, ' "Flamingo"', ' Youth 378', 'Tokyo');

INSERT 0 1

dblab1=> insert into books3 values (10, '"Juventa"', ' Sadovaya 13', 'New York');

INSERT 0 1

-- Посмотрим, что получилось

dblab1=> select \*from books3;

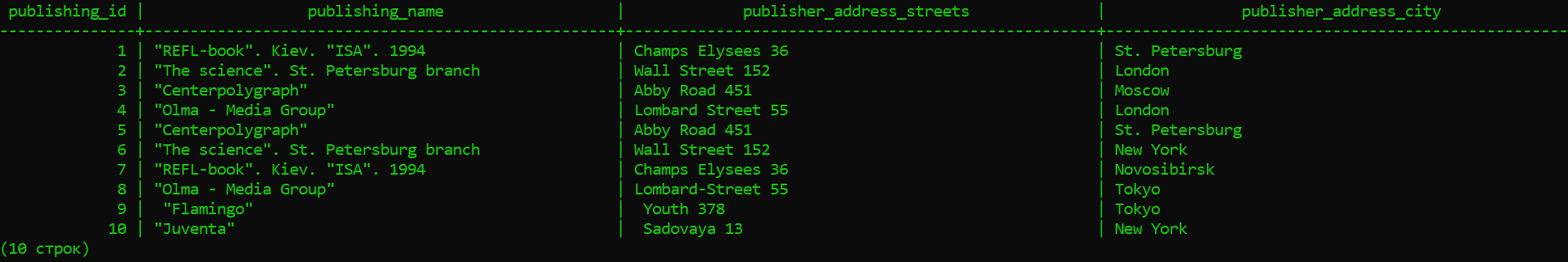


Рис. 1

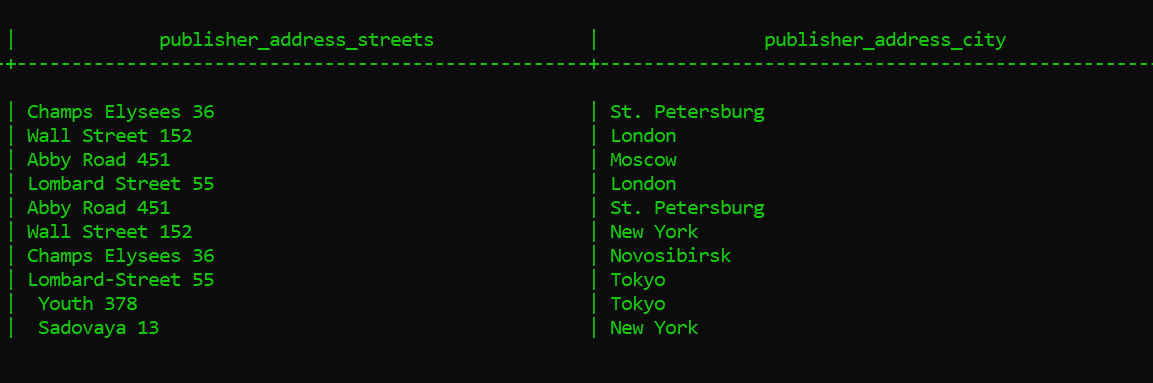


Рис. 2

-- Создаем четвертую таблицу country

dblab1=> create table country (country\_id int, country\_name char(30));

CREATE TABLE

-- Посмотрим, что получилось

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------

(0 строк)

-- Заполним таблицу

dblab1=> insert into country values (1, 'Russia');

INSERT 0 1

dblab1=> insert into country values (2, 'USA');

INSERT 0 1

dblab1=> insert into country values (3, 'Russia');

INSERT 0 1

dblab1=> insert into country values (4, 'Japan');

INSERT 0 1

dblab1=> insert into country values (5, 'Russia');

INSERT 0 1

dblab1=> insert into country values (6, 'Czech');

INSERT 0 1

dblab1=> insert into country values (7, 'Russia');

INSERT 0 1

dblab1=> insert into country values (8, 'Slovakia');

INSERT 0 1

dblab1=> insert into country values (9, 'Czech');

INSERT 0 1

dblab1=> insert into country values (10, 'Slovakia');

INSERT 0 1

-- Посмотрим, что получилось

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

(10 строк)

-- теперь нужно объединить таблицы

-- Создаем таблицу result, где объединим заданные ранее таблицы по id

dblab1=> create table result as select genre\_id, book\_id, book\_name, after\_book’s, publishing\_id, publishing\_name, publisher\_address\_streets, country\_id, country\_name from genre, books2, books3, country where genre\_id= book\_id and publishing\_id =country\_id and genre\_id =country\_id;

SELECT 10

-- Посмотрим, что получилось

dblab1=> select \* from result;

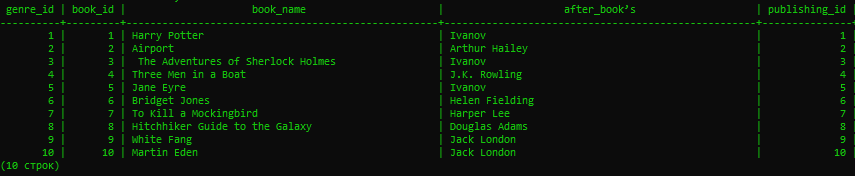


Рис. 3

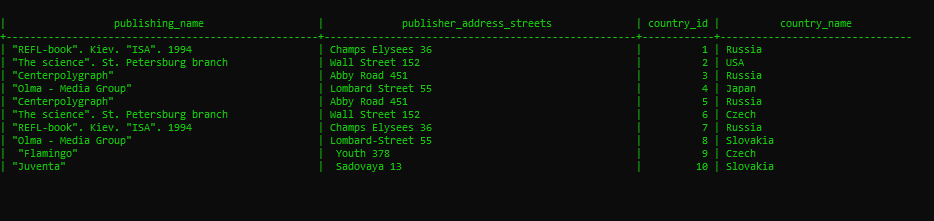


Рис. 4

-- Создадим транзакционный блок

dblab1=> begin;

BEGIN

-- Вставим новое поле в таблицу country

dblab1=> insert into country values (11, 'Italy');

INSERT 0 1

-- Посмотрим, что получилось

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

11 | Italy

(11 строк)

-- Откатим изменения

dblab1=> rollback;

ROLLBACK

-- Посмотрим, что получилось

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

(10 строк)

-- Создадим еще один транзакционный блок

dblab1=> begin;

BEGIN

-- Открываем еще одну копию шелла, входим в БД и добавляем поле

Server [localhost]:

Database [postgres]: dblab1

Port [5432]:

Username [postgres]: vanin

Пароль пользователя vanin:

psql (10.14)

ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной

страницы Windows (1251).

8-битовые (русские) символы могут отображаться некорректно.

Подробнее об этом смотрите документацию psql, раздел

"Notes for Windows users".

Введите "help", чтобы получить справку.

dblab1=> \! chcp 1251

Текущая кодовая страница: 1251

dblab1=> insert into country values (11, 'Italy');

INSERT 0 1

-- Возвращаемся в первую копию шелла и смотрим, что получилось

dblab1=> select \* from country;

ОШИБКА: текущая транзакция прервана, команды до конца блока транзакции игнорируются

-- Закрываем второй шелл и повторяем действие, происходит такая же ошибка что и описана выше.

--После закрытия первого шелла и повторной выполненной команды, получаем

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

11 | Italy

(11 строк)

-- Удалим ненужную запись по id

dblab1=> delete from country where country\_id=11;

DELETE 1

-- Посмотрим, что получилось

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

(10 строк)

-- Создадим еще один транзакционный блок

dblab1=> begin;

BEGIN

-- Создадим точку сохранения

dblab1=> savepoint save;

SAVEPOINT

--Вставим в таблицу новое поле

dblab1=> insert into country values (11, 'Italy');

INSERT 0 1

-- Откатимся к точке сохранения

dblab1=> rollback to save;

ROLLBACK

-- Посмотрим, что получилось

dblab1=> select \* from country;

country\_id | country\_name

------------+--------------------------------

1 | Russia

2 | USA

3 | Russia

4 | Japan

5 | Russia

6 | Czech

7 | Russia

8 | Slovakia

9 | Czech

10 | Slovakia

(10 строк)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Вывод**

В процессе лабораторной работы были получены навыки при создании пользователей, баз данных, таблиц баз данных. Были изучены основные команды и правила их записи для работы с базами данными. Был создан пользователь, наделённый правами, база данных, содержащая определённые таблицы и поля, и наложены ограничения на эти поля. Над таблицами были произведены операции вставки, удаления, просмотра, объединения и прочие. В результате была получена база данных, обладающая требованиями, предъявляемые в варианте задания.