Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Новосибирский государственный технический университет»

Кафедра вычислительной техники

**Отчет**

По лабораторной работе №1

По дисциплине: «Базы данных»

Ознакомление с основами работы в среде PostgreSQL

Вариант 10

Выполнили: Павлюк А.С. Преподаватель: Харюткина С. А.

Группа: АВТ-008

Факультет: АВТ

Новосибирск, 2022

Оглавление

[Цель работы 3](#_Toc112622182)

[Задание 3](#_Toc112622183)

[Структура базы данных 4](#_Toc112622184)

[Ход работы 4](#_Toc112622185)

[Вывод 8](#_Toc112622186)

[Приложение 9](#_Toc112622187)

Цель работы

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

Задание

Создать и заполнить базу данных своего варианта в PostgreSQL. Таблицы (минимум по 10 записей в каждой) связать между собой полями идентификаторов. Предусмотреть наличие связей типа: «один ко многим». 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. Предусмотреть наличие таблиц-справочников и таблиц, использующих справочники.

*Вариант 10:* создать и заполнить базу данных туристического агентства, состоящую из четырех таблиц. Первая таблица должна содержать поля: идентификатор заказа, дата заказа, номер заказа, идентификатор тура, идентификатор руководителя тура, количество участников и другие поля при необходимости. Вторая: идентификатор тура, вид тура (автобусный, железнодорожный, авиа), стоимость тура, дата начала тура, идентификатор города путешествия и другие поля при необходимости. Третья: идентификатор руководителя тура, фамилия руководителя, дата рождения и другие поля при необходимости. Четвертая: справочник городов. На основании созданных таблиц создать таблицу, содержащую, например, поля: дата заказа, фамилия руководителя тура, вид тура, наименование города путешествия, дата начала тура, стоимость тура.

Структура базы данных

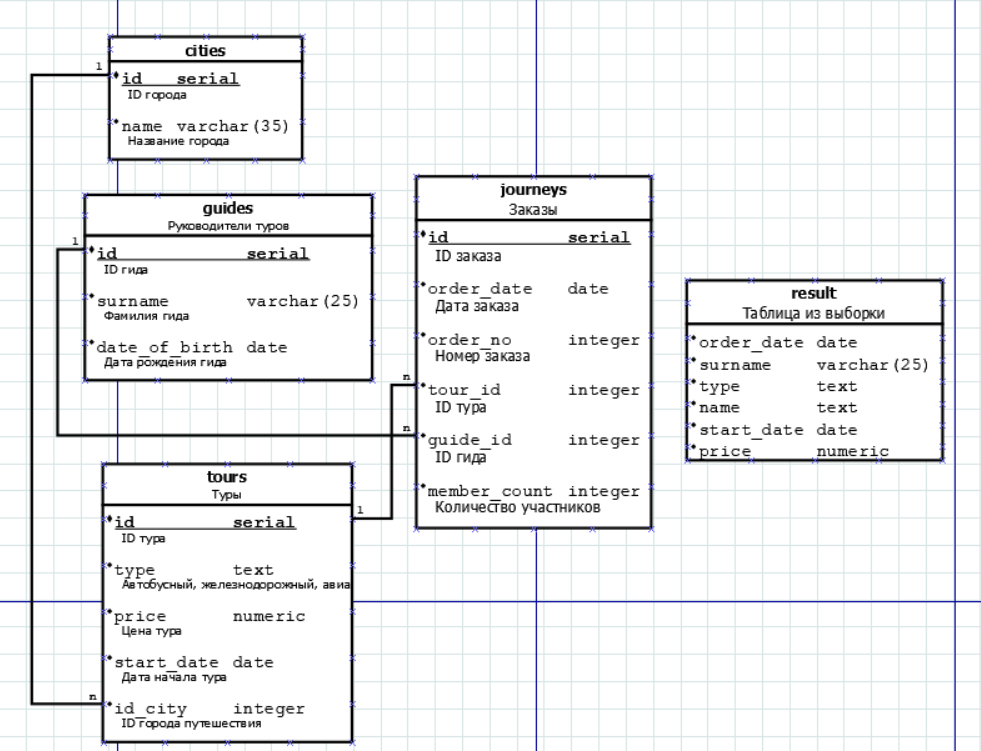


Рисунок 1. Структура базы данных

Ход работы

Для выполнения лабораторной работы была выбрана СУБД PostgreSQL, были использованы наборы операторов для **определения** (Data Definition Language) и **манипуляции** (Data Manipulation Language) данными.

DLL:

1. CREATE – создание объектов базы данных;
2. ALTER – изменение объектов базы данных;
3. DROP – удаление объектов базы данных.

DML:

1. SELECT – осуществляет выборку данных;
2. INSERT – добавляет новые данные;
3. UPDATE – изменяет существующие данные;
4. DELETE – удаляет данные.

Начнем выполнение задания с создания базы банных **tour\_agency** и подключения к ней.

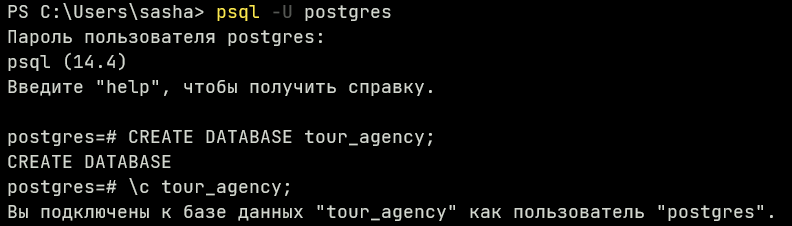


Рисунок 2. Создание базы данных

Далее создадим таблицу-справочник **cities**, которая содержит идентификаторы и названия городов.

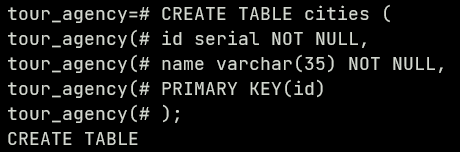


Рисунок 3. Создание таблицы-справочника городов

Тип данных **serial** – целое с автоувеличением.

**Primary key** – первичный ключ, значение должно быть уникальным и ненулевым.

Обновим таблицу cities, добавив в нее поле population – численность населения.

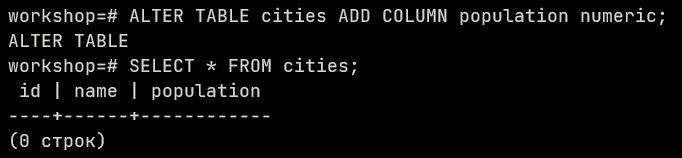


Рисунок 4. Добавление поля population

Приведем таблицу к первоначальному виду.

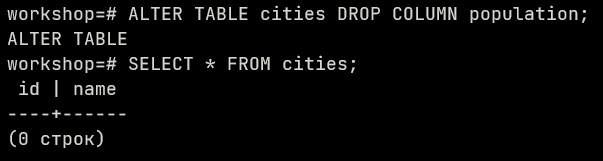


Рисунок 5. Удаление поля population

Далее создадим оставшиеся таблицы согласно варианту.

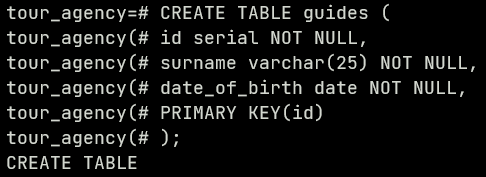


Рисунок 6. Создание таблицы руководителей туров

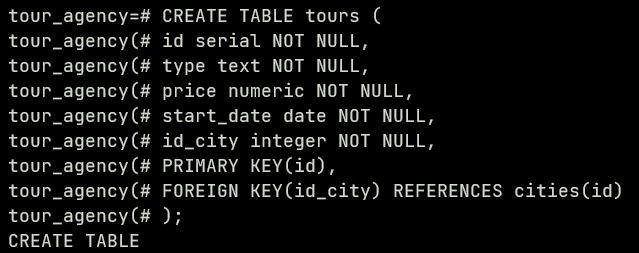


Рисунок 7. Создание таблицы туров

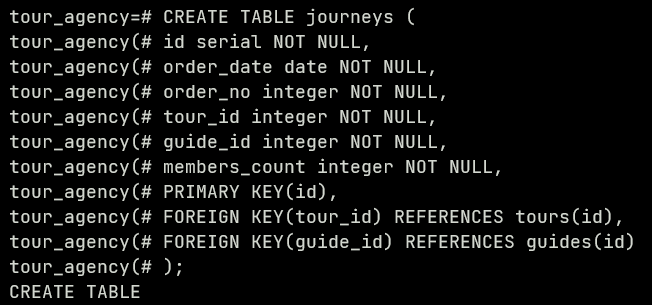


Рисунок 8. Создание таблицы заказов в туристическом агентстве

В данной БД присутствуют связи «Один ко Многим» - один гид на несколько заказов, один город на несколько туров. Для этого применяются внешние ключи (Foreign key) и ключевое слово **references**.

Заполним таблицы произвольными значениями. Операцию добавления записей в таблицу покажем на примере таблицы-справочника городов.

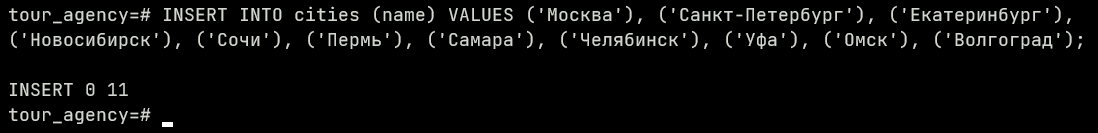


Рисунок 9. Вставка значений в таблицу

На основании созданных таблиц создадим таблицу, содержащую поля: дата заказа, фамилия руководителя тура, вид тура, наименование города путешествия, дата начала тура, стоимость тура.

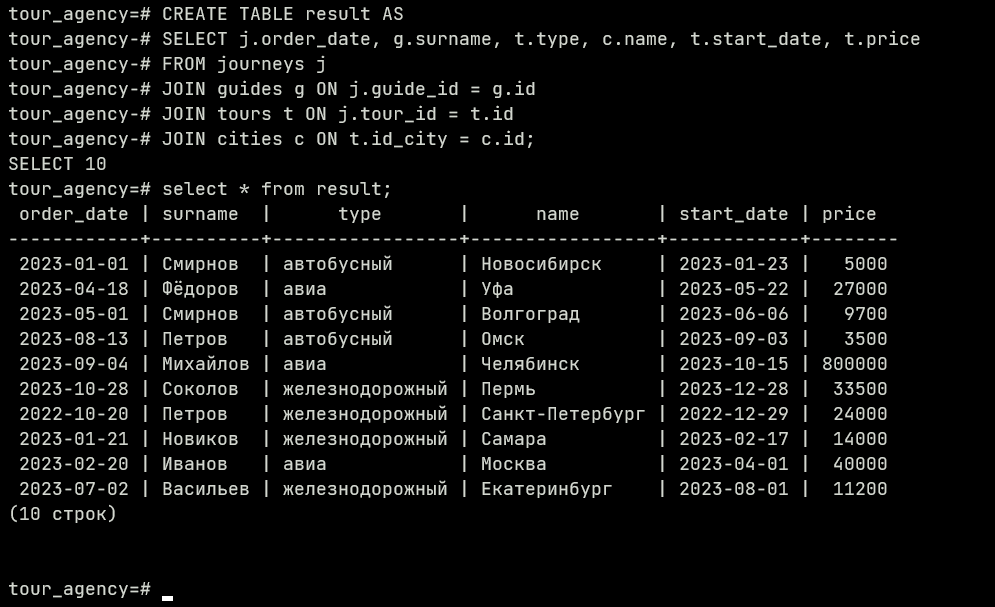


Рисунок 10. Выборка данных из таблиц

Вывод

В результате выполнения лабораторной работы были изучены базовые операции по работе с базой данных, синтаксис команд. Также были приобретены навыки создания баз данных, создания, заполнения и модификации таблиц в PostgreSQL. Была спроектирована база данных туристического агентства.

Приложение

**Создание**

CREATE DATABASE tour\_agency;

CREATE TABLE cities (

id serial NOT NULL,

name varchar(35) NOT NULL,

PRIMARY KEY(id)

);

CREATE TABLE guides (

id serial NOT NULL,

surname varchar(25) NOT NULL,

date\_of\_birth date NOT NULL,

PRIMARY KEY(id)

);

CREATE TABLE tours (

id serial NOT NULL,

type text NOT NULL,

price numeric NOT NULL,

start\_date date NOT NULL,

id\_city integer NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY(id\_city) REFERENCES cities(id)

);

CREATE TABLE journeys (

id serial NOT NULL,

order\_date date NOT NULL,

order\_no integer NOT NULL,

tour\_id integer NOT NULL,

guide\_id integer NOT NULL,

members\_count integer NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY(tour\_id) REFERENCES tours(id),

FOREIGN KEY(guide\_id) REFERENCES guides(id)

);

**Заполнение**

INSERT INTO journeys (order\_date, order\_no, tour\_id, guide\_id, members\_count) VALUES ('2022-10-20', 653244, 1, 6, 60);

INSERT INTO cities (name) VALUES ('Москва'), ('Санкт-Петербург'), ('Екатеринбург'), ('Новосибирск'), ('Сочи'), ('Пермь'), ('Самара'), ('Челябинск'), ('Уфа'), ('Омск'), ('Волгоград');

INSERT INTO tours (type, price, start\_date, id\_city) VALUES ('железнодорожный', 24000, '2022-12-29', 2);

INSERT INTO guides (surname, date\_of\_birth) VALUES ('Иванов', '1994-08-07');

**Выборка**

CREATE TABLE result AS

SELECT j.order\_date, g.surname, t.type, c.name, t.start\_date, t.price

FROM journeys j

JOIN guides g ON j.guide\_id = g.id

JOIN tours t ON j.tour\_id = t.id

JOIN cities c ON t.id\_city = c.id;