|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №1**

по дисциплине «Технологии обработки транзакций клиент-серверных приложений»

**Студент группы** ИКБО-20-21 Сидоров С.Д.

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

(подпись студента)

**Руководитель практической работы** преподаватель Маличенко С.В.

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

(подпись руководителя)

Работа представлена « » 2024 г.

Москва 2023

**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

Подготовить рабочее место, где предполагается выполнение практических работ, установив СУБД PostgreSQL. Можно использовать СУБД в изолированной среде (докер контейнере). Для начала нужно создать базу данных в СУБД в соответствии с предложенной схемой базы данных. Заполнить таблицы данными, не менее 5 строк на каждую таблицу.

**РЕФЕРАТ**

Отчёт 14 с., 3 рис., 3 источн.

POSTGRESQL

Объектом разработки является база данных.

Цель работы – разработка и заполнение базы данных.

В ходе работы была разработана и заполнена данным база данных.

Результатом является заполненная база данных.

Report 14 p., 3 fig., 3 sources.

The object of development is the database.

The goal of the work is the development and population of the database.

During the work, the database was developed and populated with data.

The result is the populated database.

**СОДЕРЖАНИЕ**

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ 9](#_Toc158807900)

[1.1 Создание базы данных 9](#_Toc158807901)

[1.2 Заполнение базы данных 9](#_Toc158807902)

[ЗАКЛЮЧЕНИЕ 10](#_Toc158807903)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 11](#_Toc158807904)

[ПРИЛОЖЕНИЕ А 12](#_Toc158807905)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

БД – база данных;

СУБД – система управления базами данных.

**ВВЕДЕНИЕ**

В современном мире организации сталкиваются с необходимостью эффективного управления большим объемом данных, которые являются ключевым активом для принятия стратегических решений. В этом контексте базы данных играют решающую роль, обеспечивая хранение, организацию и быстрый доступ к информации.

Целью данного проекта является разработка базы данных с использованием системы управления базами данных PostgreSQL. PostgreSQL - это мощная и надежная объектно-реляционная система управления базами данных, которая предоставляет широкий спектр функций для создания и управления данными.

В данной работе будет рассмотрена процедура разработки базы данных, начиная с проектирования ее структуры и определения сущностей, заканчивая заполнением данных и обеспечением безопасности.

1. **РАЗРАБОТКА БАЗЫ ДАННЫХ**

## Создание базы данных

Для работы с базой данных использовалась СУБД PostgreSQL. Результат создания базы данных представлен на рисунке 1.



Рисунок 1 – Результат создания БД

После создания базы данных и переключение на неё были созданы таблицы. Пример создания таблицы представлен на рисунке 2.

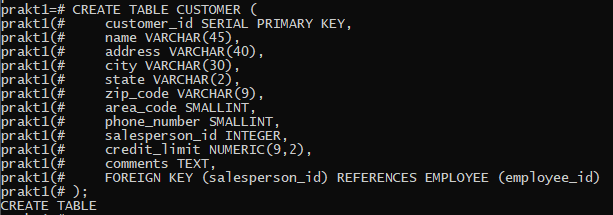


Рисунок 2 – Пример создания таблицы

## Заполнение базы данных

После создания базы данных, её таблицы были заполнены записями. Пример заполнения таблиц представлен на рисунке 3.

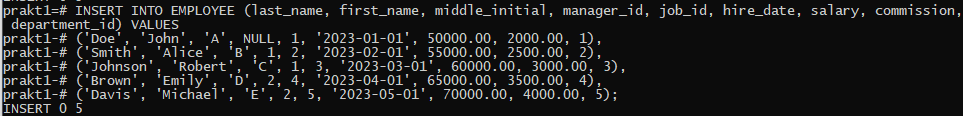


Рисунок 3 – Пример заполнения таблицы

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы была разработана и заполнена база данных с использованием СУБД PostgreSQL.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PostgreSQL: Официальная документация [Электронный ресурс] – URL: https://www.postgresql.org/docs/ (дата обращения: 14.02.2024).
2. PostgreSQL Tutorial for Beginners [Электронный ресурс] – URL: https://www.tutorialspoint.com/postgresql/index.htm (дата обращения: 14.02.2024).
3. PostgreSQL: Википедия [Электронный ресурс] – URL: https://en.wikipedia.org/wiki/PostgreSQL (дата обращения: 14.02.2024).
4. PostgreSQL: Primary Key Constraint. [Электронный ресурс] – URL: https://www.tutorialsteacher.com/postgresql/foreign-key (дата обращения: 14.02.2024).
5. PostgreSQL Foreign Key Constraint. [Электронный ресурс] – URL: https://www.tutorialsteacher.com/postgresql/foreign-key (дата обращения: 17.12.2023)

# ПРИЛОЖЕНИЕ А

**ФРАГМЕНТЫ КОДА РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ**

Листинг 1 – Фрагмент кода создания таблиц

CREATE TABLE LOCATION (

location\_id SERIAL PRIMARY KEY,

regional\_group VARCHAR(20)

);

CREATE TABLE DEPARTMENT (

department\_id SERIAL PRIMARY KEY,

name VARCHAR(14),

location\_id INTEGER,

FOREIGN KEY (location\_id) REFERENCES LOCATION (location\_id)

);

CREATE TABLE JOB (

job\_id SERIAL PRIMARY KEY,

function VARCHAR(30)

);

CREATE TABLE EMPLOYEE (

employee\_id SERIAL PRIMARY KEY,

last\_name VARCHAR(15),

first\_name VARCHAR(15),

middle\_initial VARCHAR(1),

manager\_id INTEGER,

CONSTRAINT ref\_emp\_manager FOREIGN KEY (manager\_id) REFERENCES EMPLOYEE(employee\_id),

job\_id INTEGER,

hire\_date DATE,

salary NUMERIC(7,2),

commission NUMERIC(7,2),

department\_id INTEGER,

FOREIGN KEY (job\_id) REFERENCES JOB (job\_id),

FOREIGN KEY (department\_id) REFERENCES DEPARTMENT (department\_id)

);

Продолжение листинга 1

CREATE TABLE CUSTOMER (

    customer\_id SERIAL PRIMARY KEY,

    name VARCHAR(45),

    address VARCHAR(40),

    city VARCHAR(30),

    state VARCHAR(2),

    zip\_code VARCHAR(9),

    area\_code SMALLINT,

    phone\_number SMALLINT,

    salesperson\_id INTEGER,

    credit\_limit NUMERIC(9,2),

    comments TEXT,

    FOREIGN KEY (salesperson\_id) REFERENCES EMPLOYEE (employee\_id)

);

CREATE TABLE SALES\_ORDER (

    order\_id SERIAL PRIMARY KEY,

    order\_date DATE,

    customer\_id INTEGER,

    ship\_date DATE,

    total NUMERIC(8,2),

    FOREIGN KEY (customer\_id) REFERENCES CUSTOMER (customer\_id)

);

CREATE TABLE PRODUCT (

    product\_id SERIAL PRIMARY KEY,

    description VARCHAR(30)

);

Продолжение листинга 1

CREATE TABLE ITEM (

item\_id SERIAL,

order\_id INTEGER,

product\_id INTEGER,

actual\_price NUMERIC(8,2),

quantity INTEGER,

total NUMERIC(8,2),

PRIMARY KEY (item\_id, order\_id),

FOREIGN KEY (order\_id) REFERENCES SALES\_ORDER (order\_id),

FOREIGN KEY (product\_id) REFERENCES PRODUCT (product\_id)

);

CREATE TABLE PRICE (

product\_id INTEGER,

start\_date DATE,

list\_price NUMERIC(8,2),

min\_price NUMERIC(8,2),

end\_date DATE,

PRIMARY KEY (product\_id, start\_date),

FOREIGN KEY (product\_id) REFERENCES PRODUCT (product\_id)

);

Листинг 2 – Фрагмент кода заполнения базы данных

INSERT INTO LOCATION (regional\_group) VALUES

('Area A'),

('Area B'),

('Area C'),

('Area D'),

('Area E');

INSERT INTO DEPARTMENT (name, location\_id) VALUES

('Department 1', 1),

('Department 2', 2),

('Department 3', 3),

('Department 4', 4),

('Department 5', 5);

Продолжение листинга 2

INSERT INTO JOB (function) VALUES

('Function 1'),

('Function 2'),

('Function 3'),

('Function 4'),

('Function 5');

INSERT INTO EMPLOYEE (last\_name, first\_name, middle\_initial, manager\_id, job\_id, hire\_date, salary, commission, department\_id) VALUES

('Doe', 'John', 'A', NULL, 1, '2023-01-01', 50000.00, 2000.00, 1),

('Smith', 'Alice', 'B', 1, 2, '2023-02-01', 55000.00, 2500.00, 2),

('Johnson', 'Robert', 'C', 1, 3, '2023-03-01', 60000.00, 3000.00, 3),

('Brown', 'Emily', 'D', 2, 4, '2023-04-01', 65000.00, 3500.00, 4),

('Davis', 'Michael', 'E', 2, 5, '2023-05-01', 70000.00, 4000.00, 5);

INSERT INTO CUSTOMER (name, address, city, state, zip\_code, area\_code, phone\_number, salesperson\_id, credit\_limit) VALUES

('Customer 1', '123 Main St', 'City A', 'CA', '12345', 123, 4567890, 1, 10000.00),

('Customer 2', '456 Elm St', 'City B', 'NY', '54321', 456, 7890123, 2, 15000.00),

('Customer 3', '789 Oak St', 'City C', 'TX', '67890', 789, 1234567, 3, 20000.00),

('Customer 4', '101 Pine St', 'City D', 'FL', '09876', 234, 5678901, 4, 25000.00),

('Customer 5', '202 Maple St', 'City E', 'WA', '56789', 345, 6789012, 5, 30000.00);

Продолжение листинга 2

INSERT INTO SALES\_ORDER (order\_date, customer\_id, ship\_date, total) VALUES

('2023-01-01', 1, '2023-01-05', 1000.00),

('2023-02-01', 2, '2023-02-05', 1500.00),

('2023-03-01', 3, '2023-03-05', 2000.00),

('2023-04-01', 4, '2023-04-05', 2500.00),

('2023-05-01', 5, '2023-05-05', 3000.00);

INSERT INTO PRODUCT (description) VALUES

('Product 1'),

('Product 2'),

('Product 3'),

('Product 4'),

('Product 5');

INSERT INTO ITEM (order\_id, product\_id, actual\_price, quantity, total) VALUES

(1, 1, 10.00, 1, 10.00),

(2, 2, 20.00, 2, 40.00),

(3, 3, 30.00, 3, 90.00),

(4, 4, 40.00, 4, 160.00),

(5, 5, 50.00, 5, 250.00);

INSERT INTO PRICE (product\_id, start\_date, list\_price, min\_price, end\_date) VALUES

(1, '2024-01-01', 10.00, 8.00, '2024-12-31'),

(2, '2024-01-01', 20.00, 16.00, '2024-12-31'),

(3, '2024-01-01', 30.00, 24.00, '2024-12-31'),

(4, '2024-01-01', 40.00, 32.00, '2024-12-31'),

(5, '2024-01-01', 50.00, 40.00, '2024-12-31');