Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа № 1

**Реляционная СУБД PostgreSQL**

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверил: Марков А.В.

Чебоксары, 2024

Цель работы: Знакомство с СУБД PostgreSQL. Установка необходимого ПО. Создание тестовой базы. Изучение основных приципов работы с БД.

Теоретические сведения.

База данных (БД) – это структурированный набор данных, организованный для удобного хранения, поиска, обновления и управления.

Основные понятия:

* Таблицы – основные структуры хранения данных (например, таблица users, orders).
* Строки (записи) – отдельные элементы данных в таблице.
* Столбцы (поля) – атрибуты данных (например, id, name, email).
* Первичный ключ (Primary Key, PK) – уникальный идентификатор записи (например, id).
* Внешний ключ (Foreign Key, FK) – ссылка на первичный ключ другой таблицы (обеспечивает связь между таблицами).

Типы баз данных:

* Реляционные (SQL) – данные хранятся в таблицах с четкой структурой (PostgreSQL, MySQL, Oracle).
* Нереляционные (NoSQL) – данные хранятся в виде документов, ключ-значение, графов (MongoDB, Redis).

PostgreSQL (сокращенно Postgres) – это реляционная система управления базами данных (СУБД) с открытым исходным кодом.

DBeaver – это универсальный графический клиент для работы с базами данных. Поддерживает PostgreSQL, MySQL, Oracle, SQLite и многие другие СУБД.

Выполнение работы.

* 1. Создание БД в PostgreSQL

Запустим pgAdmin и создаем новое подключение и базу данных.

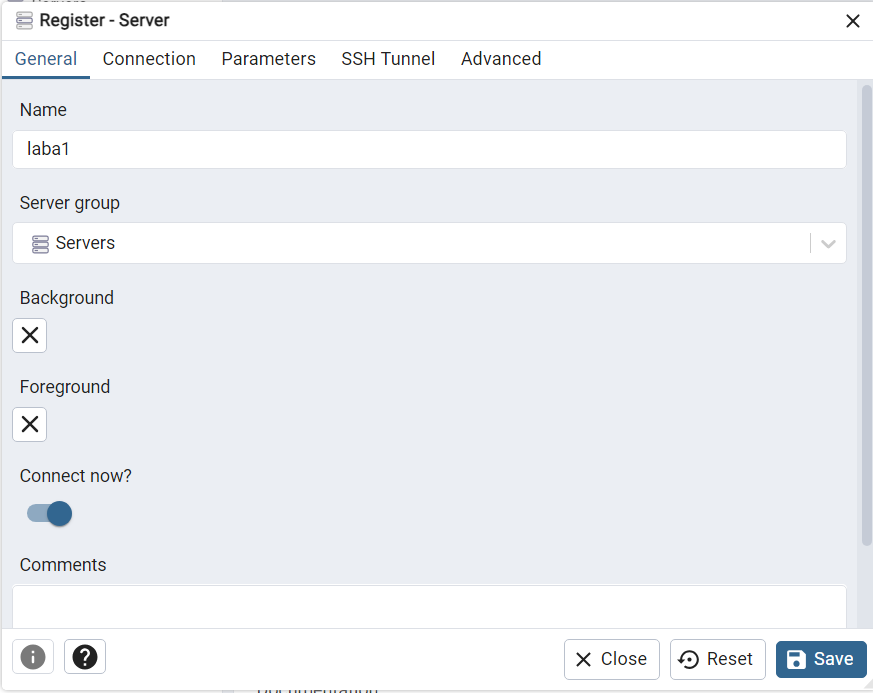


Рис.1. Создание нового сервера

Настраиваем соединение.

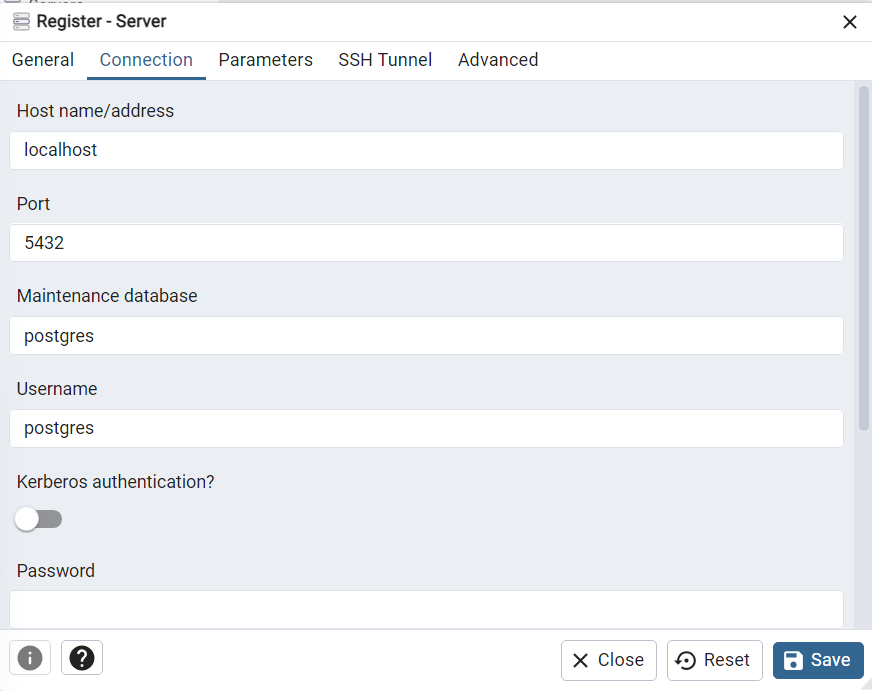


Рис.2. Настройка соединения новой бд.

Добавим пользователя

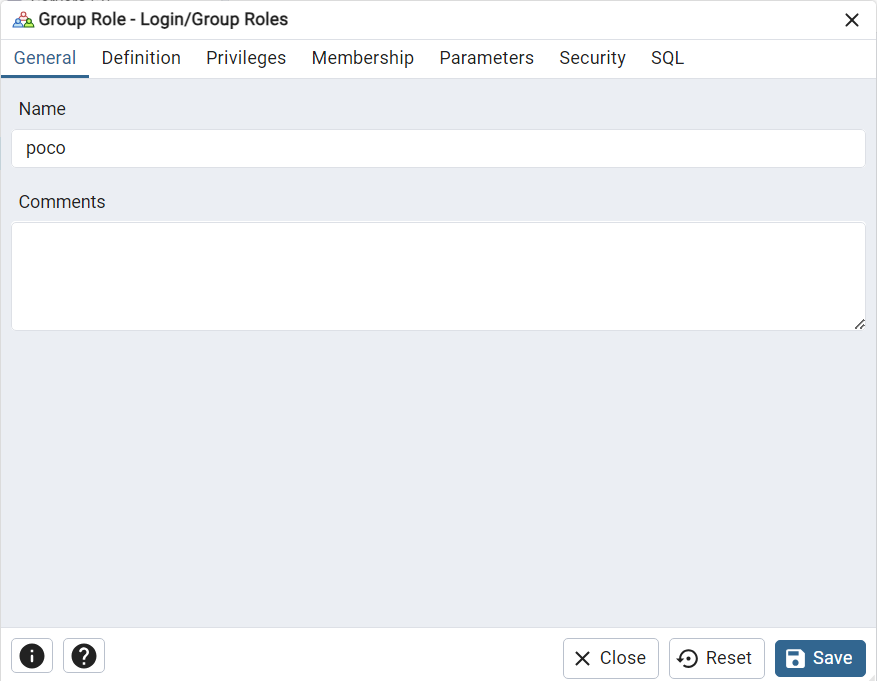


Рис.3. Добавление нового пользователя.

Придумаем пароль для нового пользователя

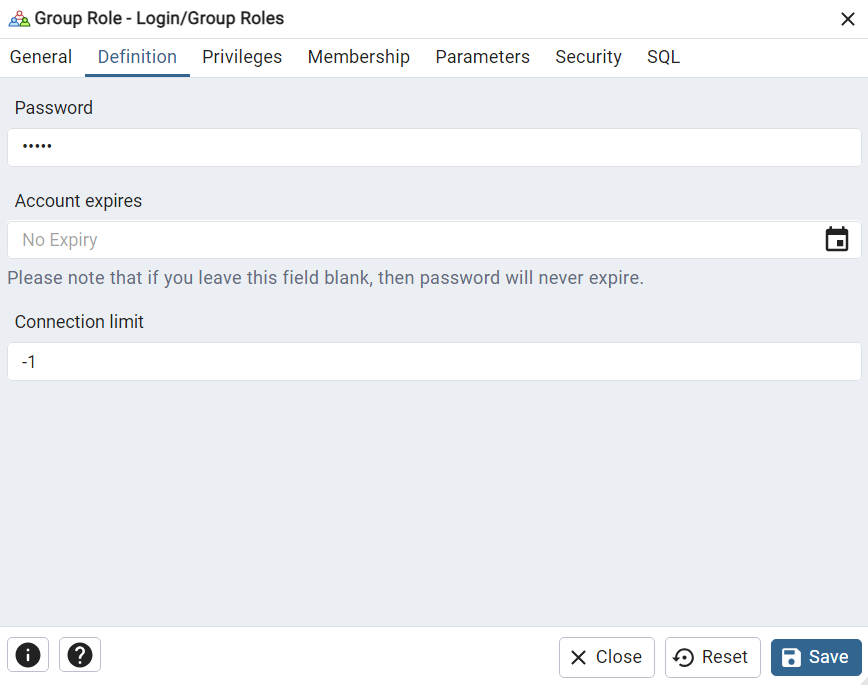


Рис.4. Пароль для пользователя

Создадим базу данных

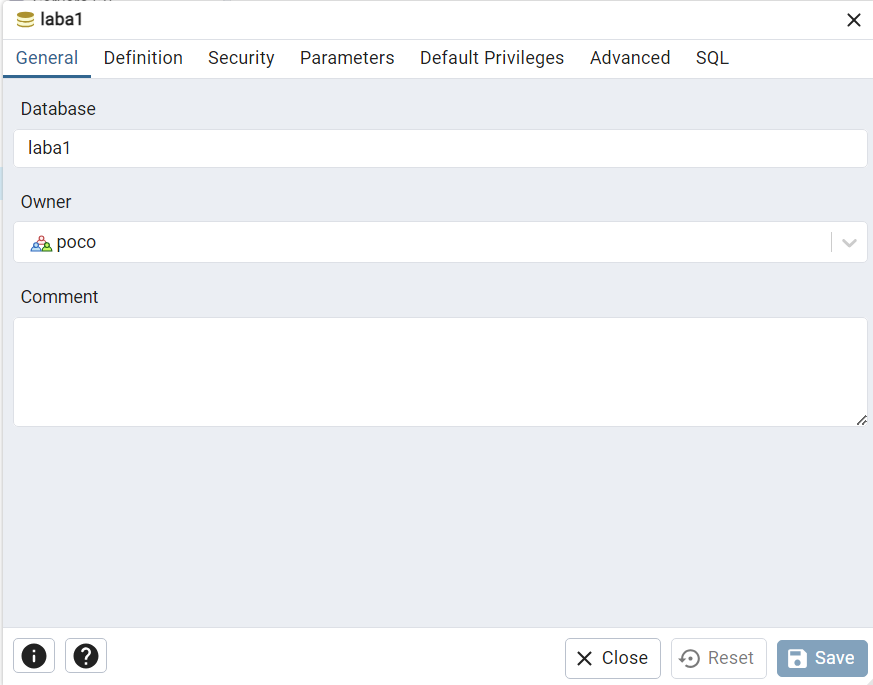


Рис.5. Создание базы данных.

* 1. Создание подключения с помощью Dbeaver

Создаем новое подключение к базе данных

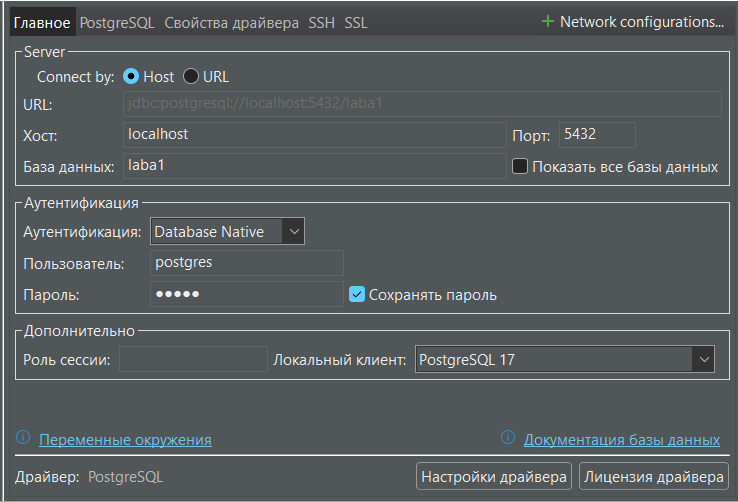


Рис.6. Создание подключения в Dbeaver.

Настроим конфигурацию соединения

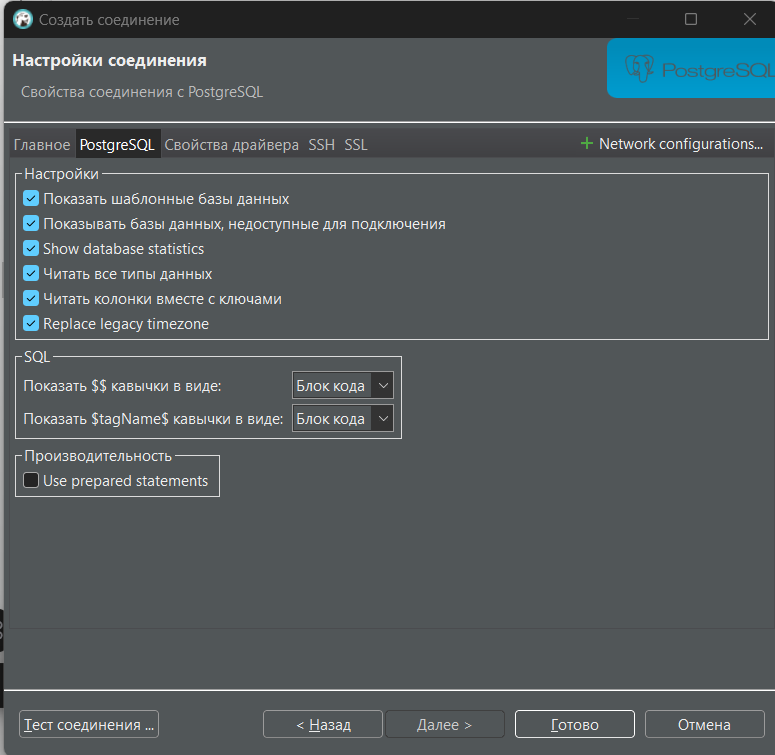


Рис.7. Настройка конфигурации.

Создадим простейший запрос для проверки подключения

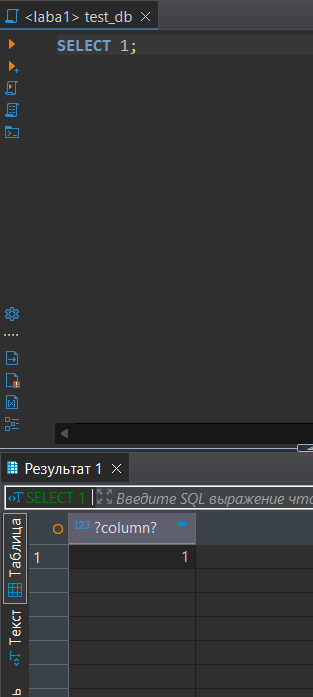


Рис.8. Создание и выполнение простейшего запроса.

Вывод есть, подключение работает.

* 1. Создание таблиц и простых SQL-запросов

Напишем скрипт для создания таблиц в БД.

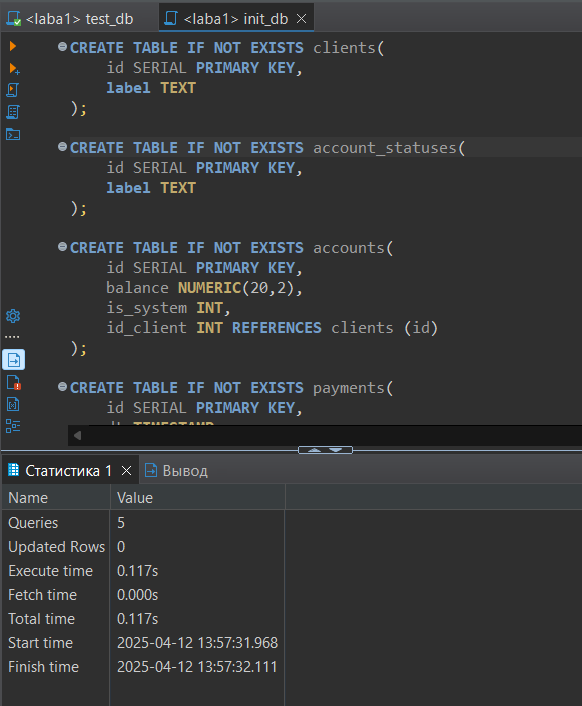


Рис.9. Скрипт создания таблиц.

Таблицы отображаются, запрос сработал.

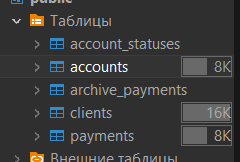


Рис.10. Результат выполнения скрипта.

Напишем запрос для вставки данных в таблицы

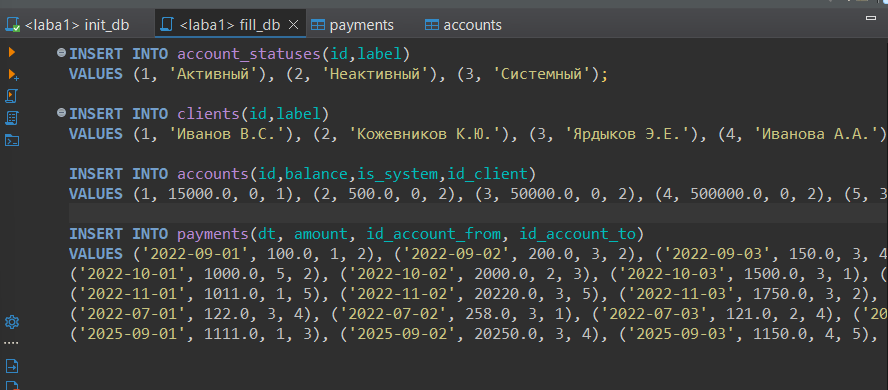


Рис.11. Скрипт для заполнения полей таблиц

Данные отображаются корректно.

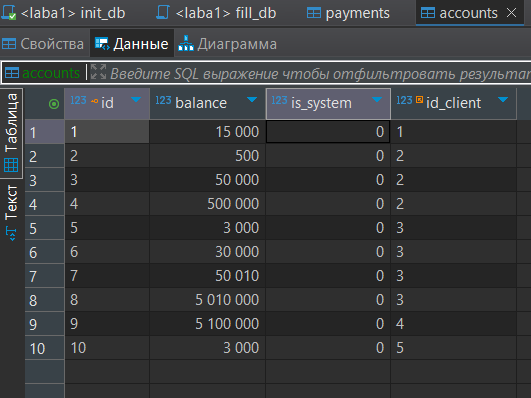


Рис.12. Отображение данных в полях.

Напишем запрос для удаления таблиц

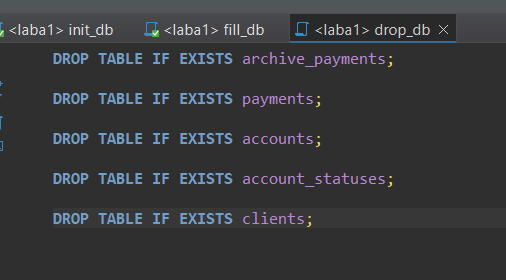


Рис.13. Скрипт удаления таблиц.

Таблиц нет, запрос так же выполняется корректно.

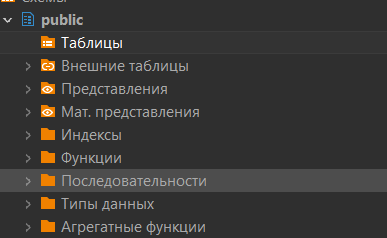


Рис.14. Результат выполнения скрипта удаления таблиц.

Весть цикл работает корректно, таблицы создаются, данные добавляются а после таблицы удалаются.

* 1. Изменение схемы данных

Напишем запрос на изменение таблиц

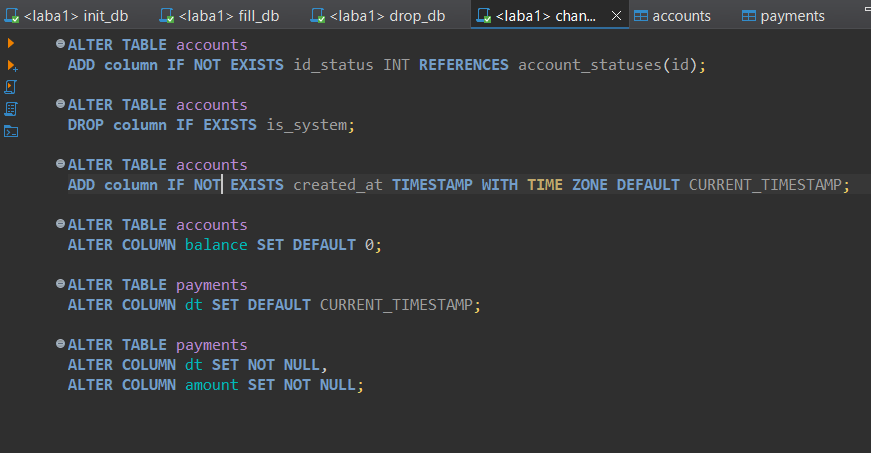
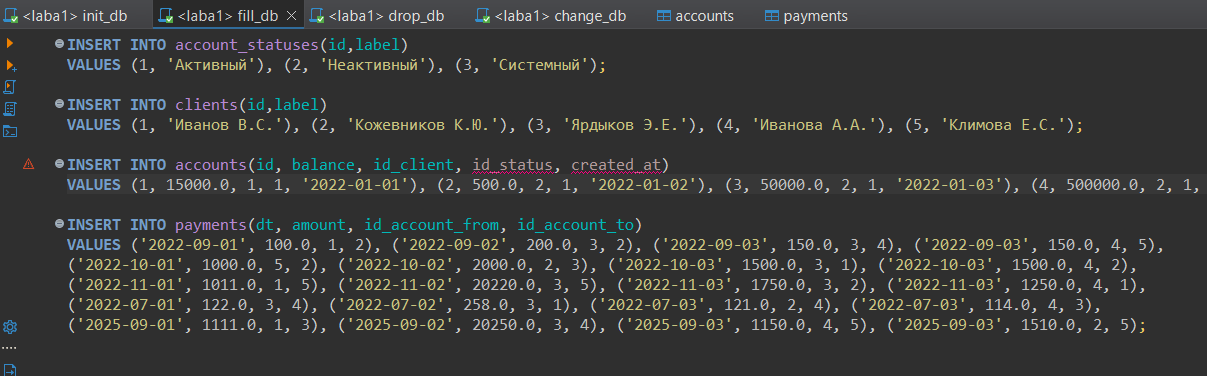


Рис.15. Скрипт добавления и удаления некоторых атрибутов.

Изменим запрос на вставку данных для обновленных атрибутов.



Запрос выполнился успешно, атрибуты изменились и добавились.

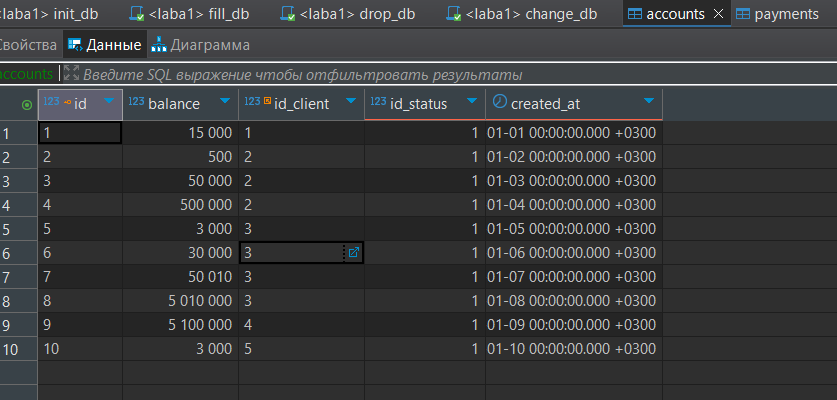


Рис.16. Результат выполнения запроса на изменение.

Создадим скрипт с Select-запросами.

1. список клиентов, являющихся плательщиками хотя бы по одному платежу

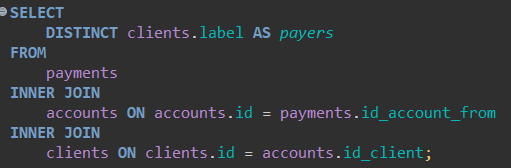


Рис.17. Запрос на выдачу клиентов-платильщиков.

1. список клиентов, являющихся получателями хотя бы по одному платежу

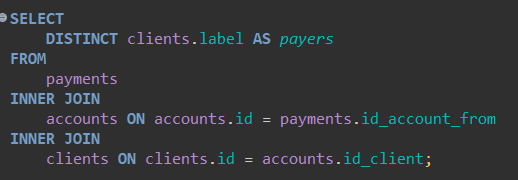


Рис.18. Запрос на выдачу клиентов-получателей.

1. id платежа с самой большой суммой

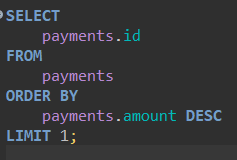


Рис.19. Запрос на выдачу платежа с самой большой суммой.

1. id трёх последних по дате платежей

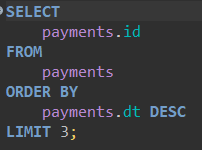


Рис.20. Запрос на выдачу id трех последних платежей.

1. общая сумма всех проведённых платежей

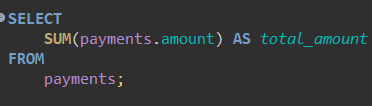


Рис.21. Запрос на выдачу общей суммы платежей.

1. количество исходящих платежей по каждому счёту

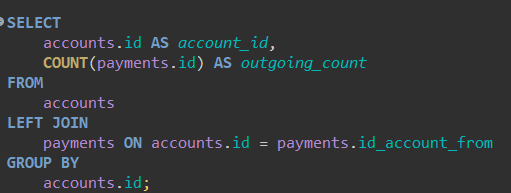


Рис.22. Запрос на выдачу кол-ва исходных платежей.

1. количество входящих платежей по каждому счёту

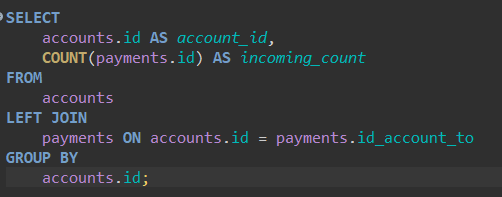


Рис.23. Запрос на выдачу кол-ва входящих платежей.

1. все платежи с указанием даты, суммы, id счёта получателя, названия клиента-получателя, id счёта плательщика, названия клиента-плательщика, упорядоченные по убыванию даты.

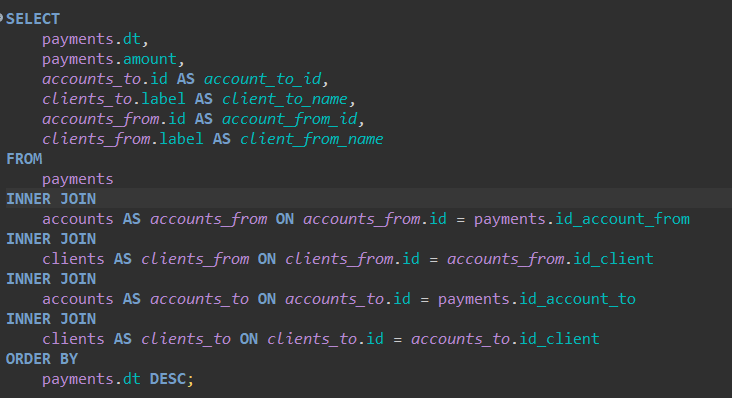


Рис.24. Запрос на выдачу всех платежей с полной информацией.

Полученная диаграмма из Dbeaver

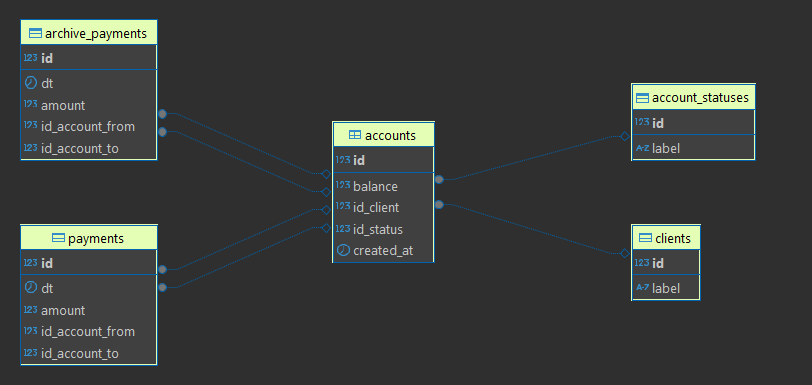


Рис.25. ER-диаграмма.

Для стиля кода выбрано руководство <https://www.sqlstyle.guide/ru/#синтаксис-запросов>.

Основные пункты стиля:

* **Идентификаторы и имена**. Осмысленные и в едином стиле.
* **Пробелы и отступы**. Логично расставленные для лучшей читаемости кода.
* **Дата и время**. Соответствующие стандарту [ISO 8601](https://ru.wikipedia.org/wiki/ISO_8601): YYYY-MM-DDTHH:MM:SS.SSSSS.
* **Функции SQL**. Стандартные вместо специфичных (определяемых поставщиком) с целью лучшей переносимости.
* **Код**. Лаконичный и без излишеств, как например: ненужные кавычки или скобки или неуместное использование оператора WHERE.
* **Комментарии**. Предпочтительно в [стиле C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)#.D0.9A.D0.BE.D0.BC.D0.BC.D0.B5.D0.BD.D1.82.D0.B0.D1.80.D0.B8.D0.B8) — /\* (начало) и \*/ (конец). Либо -- перед комментарием, тогда окончанием будет новая строка.

**Согласно стилю кода, был изменен скрипт fill\_db.sql.**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис.30. Измененный скрипт под стиль кода.

Вывод: ознакомился с СУБД PostgreSQL. Установил необходимого ПО. Создал тестовую базу. Изучил основные приципы работы с БД.