**Overview of the JDBC PostgreSQL Implementation - TP1**

**MARIE THERESE HANNA 243104**

GITHUB LINK: <https://github.com/marietheresehanna/AD>

**Project Overview**

In this project, we implemented a **JDBC (Java Database Connectivity) connection** between a **Java standalone application** and a **PostgreSQL 16 database**. The objective was to establish a connection, create a table, insert sample data, retrieve data using SQL queries, and validate the connection.

The project workflow included the following steps:

**Step 1: Setting Up PostgreSQL**

* Installed **PostgreSQL 16** and verified that the server is running.
* Created a new database named **tp1** using **pgAdmin**.
* Verified the connection by running SELECT version(); in the **Query Tool**.

Screenshot: pgAdmin dashboard confirming the PostgreSQL installation and database creation.

**Step 2: Creating and Populating the Database**

* Created a **users** table in the database using the following SQL query:
* CREATE TABLE users (
* id SERIAL PRIMARY KEY,
* name VARCHAR(50),
* email VARCHAR(100)
* );
* Inserted sample data:
* INSERT INTO users (name, email) VALUES
* ('Alice', 'alice@example.com'),
* ('Bob', 'bob@example.com'),
* ('Charlie', 'charlie@example.com');
* Verified the data by running:
* SELECT \* FROM users;

Screenshot: Query Tool displaying the users table with inserted data.

**Step 3: Implementing Java JDBC Connection**

* Downloaded and added the **PostgreSQL JDBC Driver (postgresql-42.7.5.jar)** to the project.
* Wrote a **Java program (PostgresJDBC.java)** to connect to the database and retrieve data:
* import java.sql.Connection;
* import java.sql.DriverManager;
* import java.sql.ResultSet;
* import java.sql.SQLException;
* import java.sql.Statement;
* public class PostgresJDBC {
* private static final String URL = "jdbc:postgresql://localhost:5432/tp1";
* private static final String USER = "postgres";
* private static final String PASSWORD = "your\_password";
* public static void main(String[] args) {
* try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
* Statement stmt = conn.createStatement();
* ResultSet rs = stmt.executeQuery("SELECT \* FROM users")) {
* System.out.println("Connected to PostgreSQL successfully!");
* while (rs.next()) {
* System.out.println("ID: " + rs.getInt("id") +
* ", Name: " + rs.getString("name") +
* ", Email: " + rs.getString("email"));
* }
* } catch (SQLException e) {
* e.printStackTrace();
* }
* }
* }
* Compiled and executed the Java program, successfully retrieving and displaying the data.

Screenshot: Terminal showing the Java program output with retrieved user data.

**Step 4: Uploading the Project to GitHub**

* Initialized a Git repository in the project folder.
* Added all project files (PostgresJDBC.java, postgresql-42.7.5.jar, README.md).
* Committed and pushed the project to a **GitHub repository**.

Screenshot: GitHub repository showing all uploaded project files.

**Final Verification and Submission**

* Verified the **PostgreSQL connection** by running SELECT version(); in pgAdmin.
* Successfully retrieved data in the **Java program output**.
* Uploaded the project to **GitHub** for submission.

Screenshot: Final pgAdmin verification query.

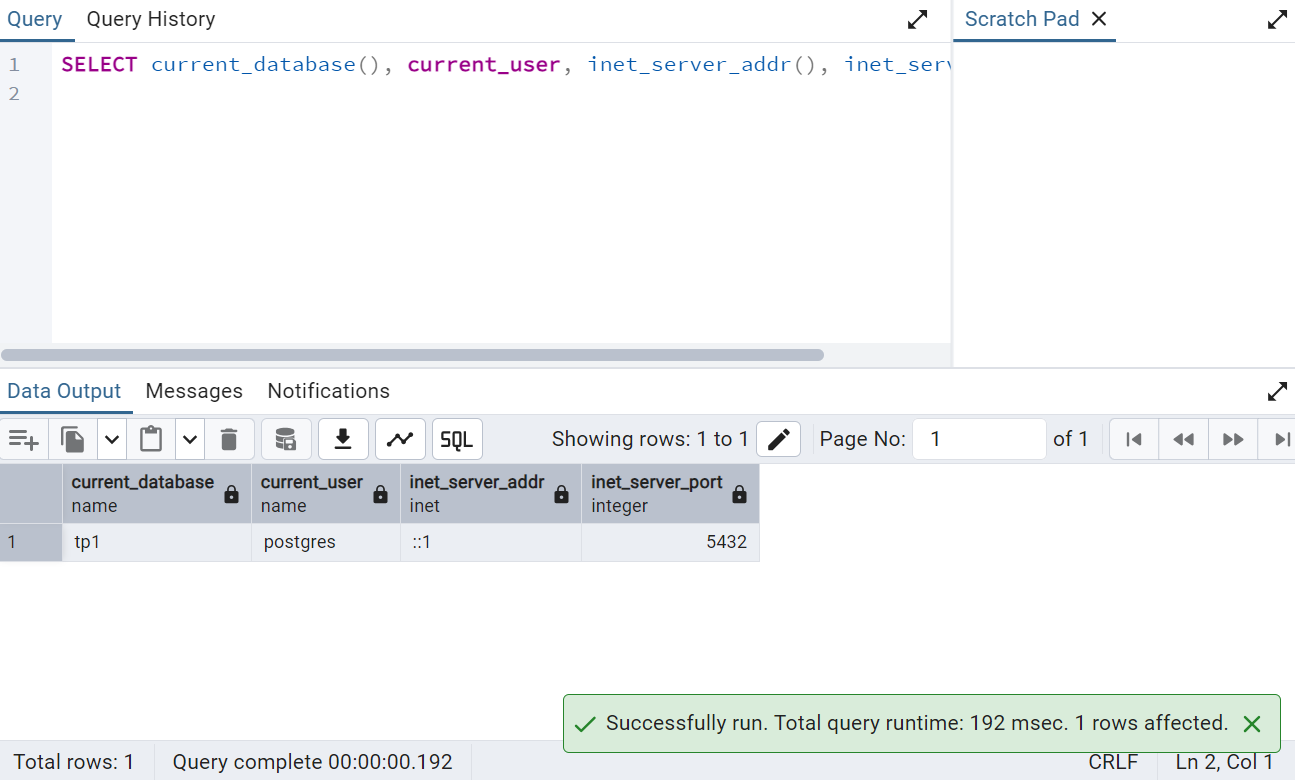
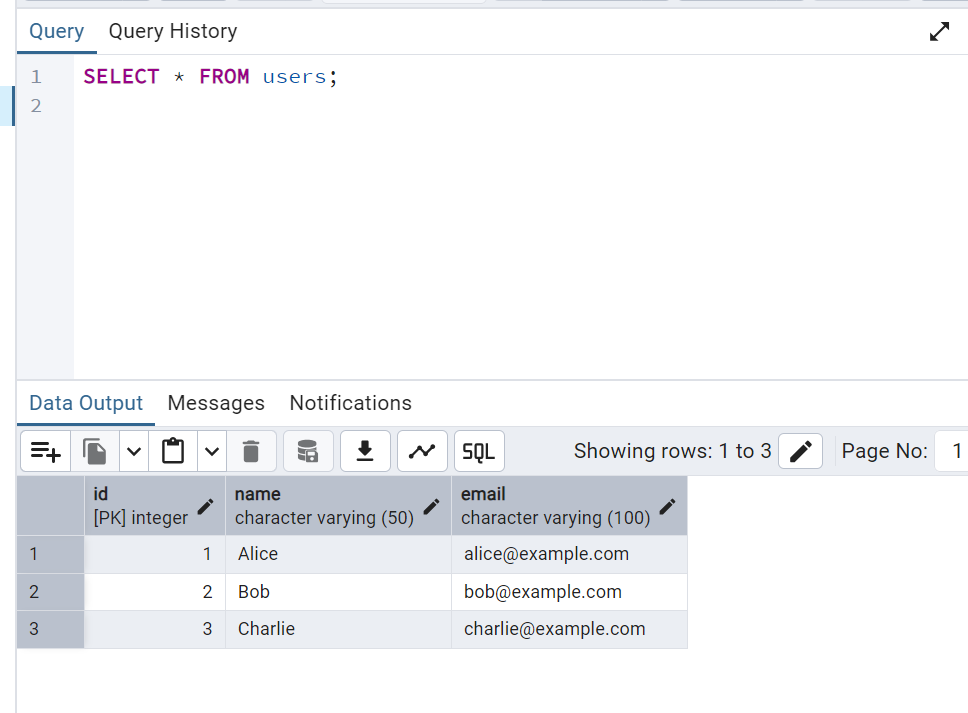
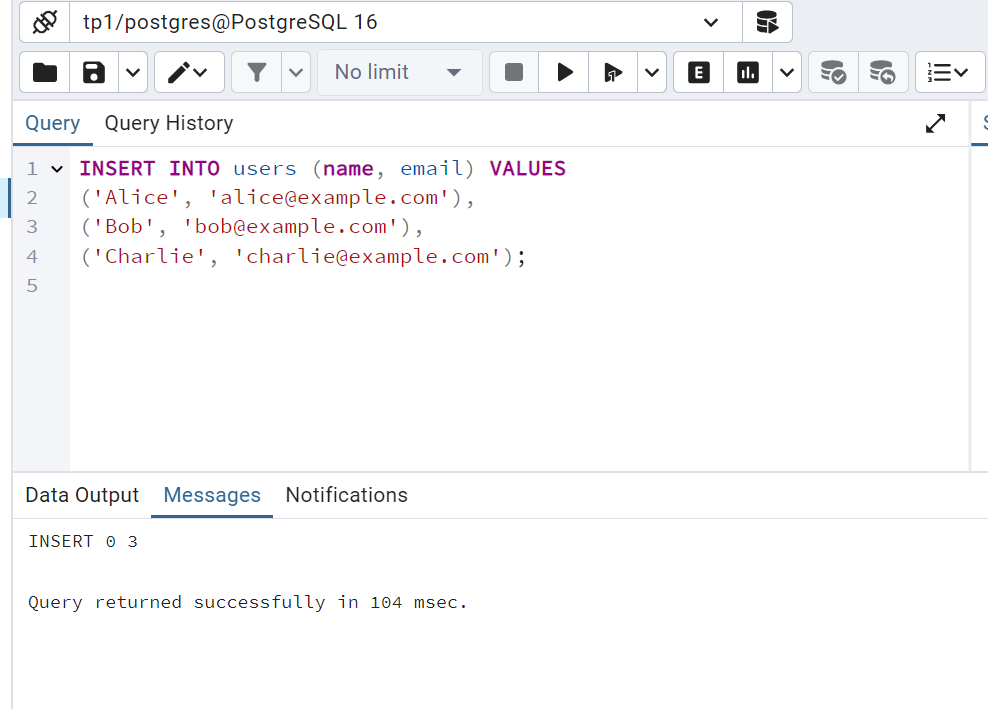
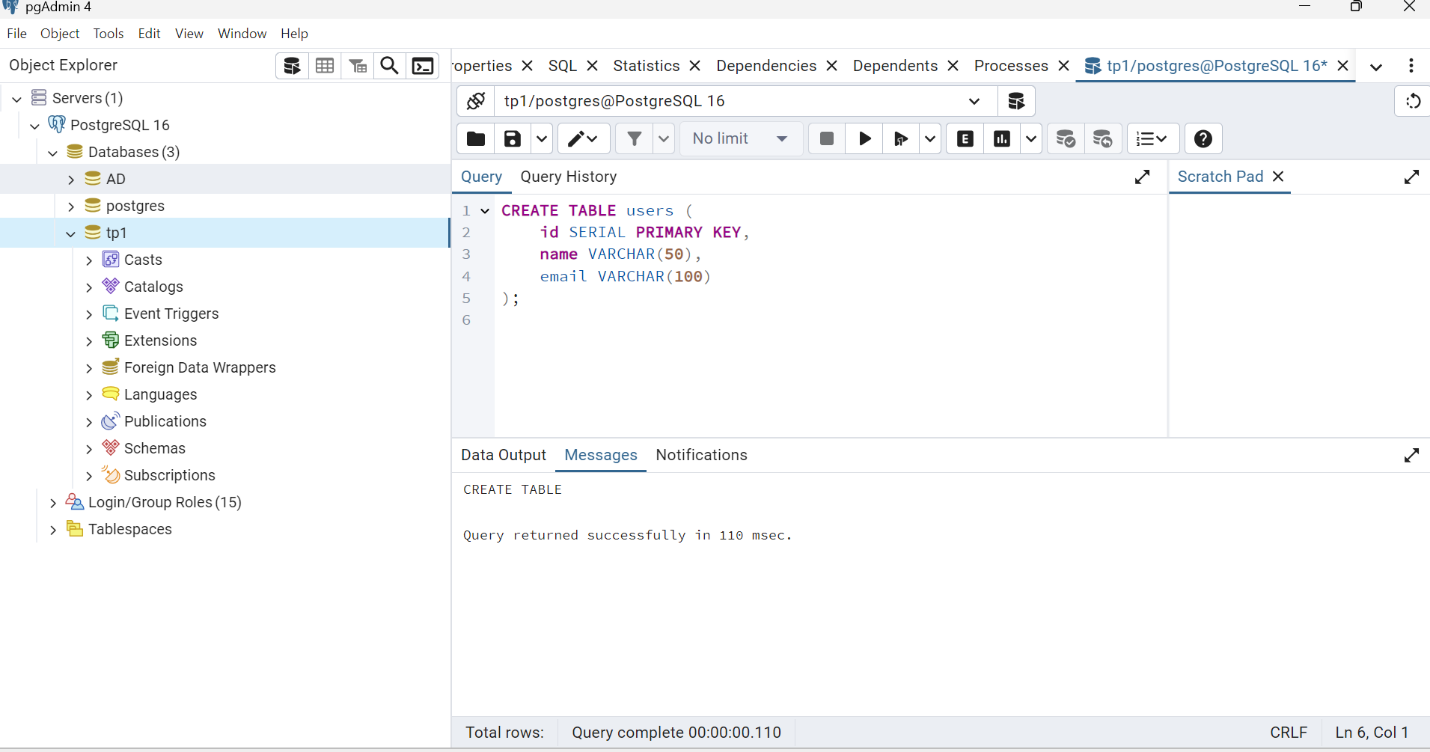
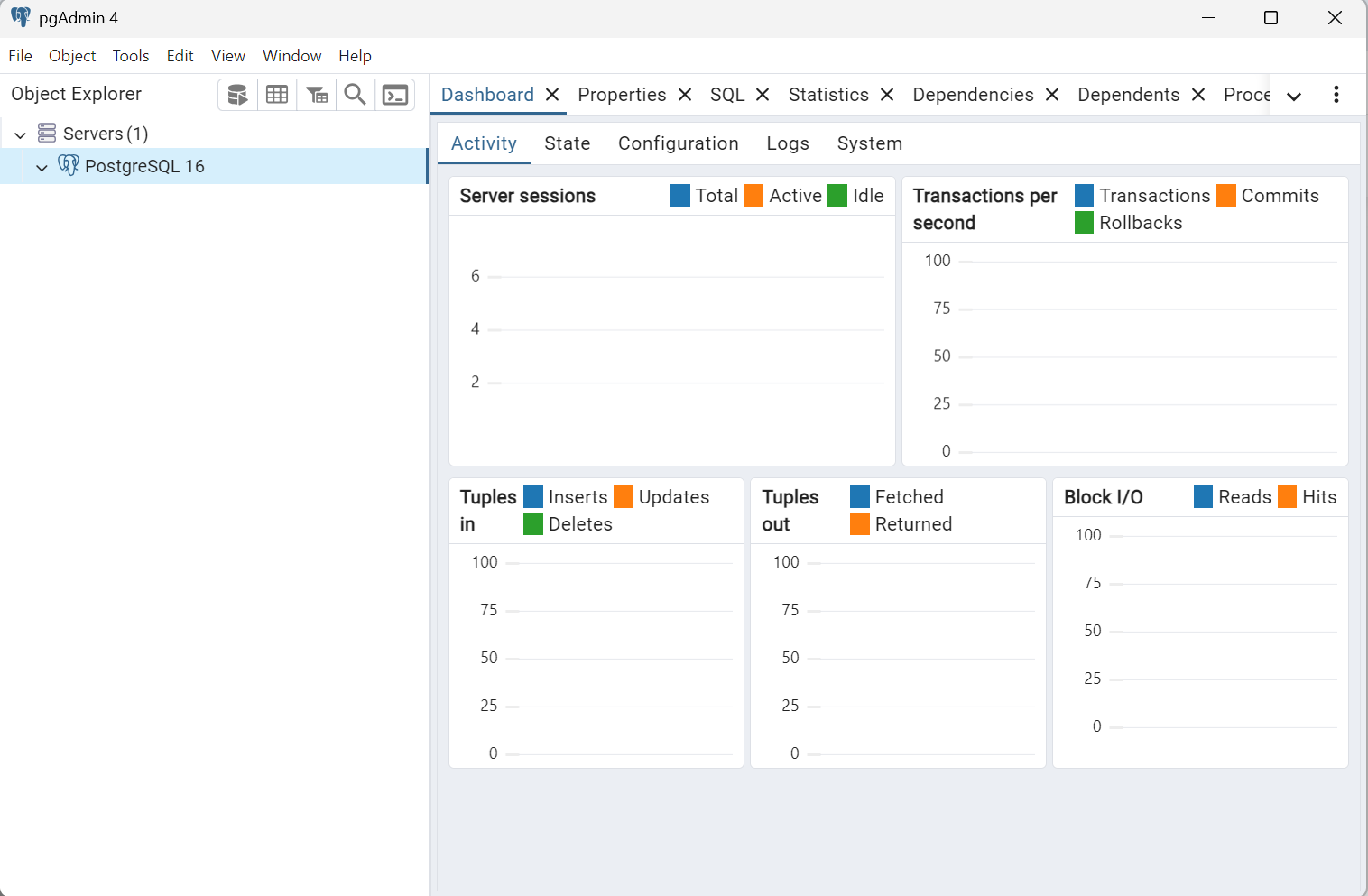
**Conclusion**

This project successfully demonstrated the implementation of **JDBC in Java** to interact with a **PostgreSQL database**. The key takeaways included:

* Establishing a **database connection** in Java.
* Performing **SQL queries** using JDBC.
* Managing a **PostgreSQL database** using **pgAdmin**.
* Using **GitHub** for version control.

**GitHub Repository:** [Insert your GitHub repo link here]

This concludes the successful implementation of the JDBC PostgreSQL connection.

A screenshot of a computer

Description automatically generated