Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Tema N° 22:
Bases de Datos

Nombre: Jhonatan Benjamin Mamani Céspedes **CUI: 20232188**

Link de GitHub: https://github.com/JBenjamin01/fp2-24b

Ejercicio 1 : Indicar los pasos para conectar y usar Java con los SGBD MySQL y PostgreSQL:

- **Configurar el entorno de desarrollo:**

  Primero se deben tener los paquetes de java y su entorno de desarrollo, para esto se debería instalar el JDK si no está ya configurado.

  Se puede usar un IDE como IntelliJ IDEA, Eclipse o NetBeans, en mi caso uso el editor de texto VS Code por las extensiones y la facilidad de observar el proceso.

  También se deben instalar los sistemas de gestión de bases de datos desde sus fuentes originales, se configuran el usuario, contraseña y el puerto que usará el servidor local al usar la base de datos.

  Descargar MySQL: https://dev.mysql.com/downloads/mysql/

  Descargar PostgreSQL: https://www.postgresql.org/download/

- **Agregar el conector JDBC:**

  Se descarga el driver JDBC para MySQL o PostgreSQL:

  MySQL: https://dev.mysql.com/downloads/connector/j/

  PostgreSQL: https://jdbc.postgresql.org/

  Estos archivos .jar del driver se colocan en alguna ubicación cercana al proyecto o simplemente con una dirección conocida. Cuando se vaya a compilar el programa se debe especificar el classpath de la dependencia de esa biblioteca.

- **Crear la base de datos y las tablas:**

  Se diseña y crea una base de datos con tablas relacionadas en MySQL o PostgreSQL.

- **Escribir el código Java:**

  Finalmente se establece la conexión utilizando la clase DriverManager y las consultas que se harán en la base de datos.

  Específicamente, son las clases Statement o PreparedStatement las que sirven para ejecutar consultas SQL.

Ejercicio 2:

Escribir un ejemplo de uso con cada SGBD y realizar 2 consultas con una BD cualquiera (de al menos 2 tablas relacionadas)

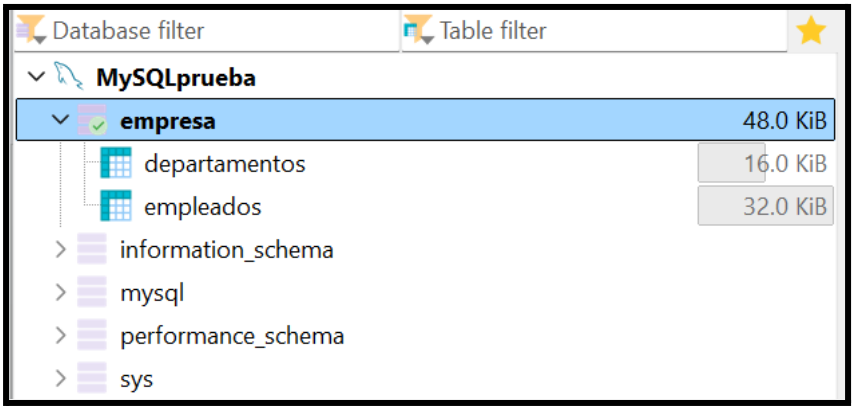
Uso de la base de datos en MySQL (Con ayuda de HeidiSQL para la interfaz)

Consultas para crear las tablas y algunos datos de prueba:

Host: localhost | Database: sys | Query*

```sql
CREATE DATABASE IF NOT EXISTS empresa;
USE empresa;

-- Crear la tabla departamentos
CREATE TABLE IF NOT EXISTS departamentos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100)
);

-- Crear la tabla empleados
CREATE TABLE IF NOT EXISTS empleados (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    edad INT,
    salario DECIMAL(10, 2),
    departamento_id INT,
    FOREIGN KEY (departamento_id) REFERENCES departamentos(id) ON DELETE CASCADE
);

-- Insertar datos en la tabla departamentos
INSERT INTO departamentos (nombre) VALUES
('Ventas'),
('Recursos Humanos'),
('TI');

-- Insertar datos en la tabla empleados
INSERT INTO empleados (nombre, edad, salario, departamento_id) VALUES
('Ana', 30, 4500.00, 1),
('Luis', 28, 3800.00, 2),
('Carlos', 35, 5000.00, 3);
```

Estructura de la base de datos ‘empresa’:



Clase EjemploMySQL.java:

```java
import java.sql.*;
public class EjemploMySQL {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3307/empresa";
        String usuario = "root";
        String contraseña = "jb301005";

        try (Connection conexion = DriverManager.getConnection(url, usuario, contraseña)) {
            System.out.println("Conexión exitosa a la base de datos 'empresa'.");

            String consulta1 = "SELECT e.nombre AS empleado, e.edad, e.salario, d.nombre AS departamento "
                             + "FROM empleados e "
                             + "JOIN departamentos d ON e.departamento_id = d.id";
            try (Statement stmt = conexion.createStatement();
                    ResultSet rs1 = stmt.executeQuery(consulta1)) {
                System.out.println("Consulta 1: Empleados y su departamento");
                while (rs1.next()) {
                    String empleado = rs1.getString("empleado");
                    int edad = rs1.getInt("edad");
                    double salario = rs1.getDouble("salario");
                    String departamento = rs1.getString("departamento");
                    System.out.println("Empleado: " + empleado + ", Edad: " + edad
                                    + ", Salario: " + salario + ", Departamento: " + departamento);
                }
            }

            String consulta2 = "SELECT d.nombre AS departamento, AVG(e.salario) AS salario_promedio "
                             + "FROM empleados e "
                             + "JOIN departamentos d ON e.departamento_id = d.id "
                             + "GROUP BY d.nombre";
            try (Statement stmt = conexion.createStatement();
                    ResultSet rs2 = stmt.executeQuery(consulta2)) {
                System.out.println("\nConsulta 2: Salario promedio por departamento");
                while (rs2.next()) {
                    String departamento = rs2.getString("departamento");
                    double salarioPromedio = rs2.getDouble("salario_promedio");
                    System.out.println("Departamento: " + departamento + ", Salario Promedio: " + salarioPromedio);
                }
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Consola:

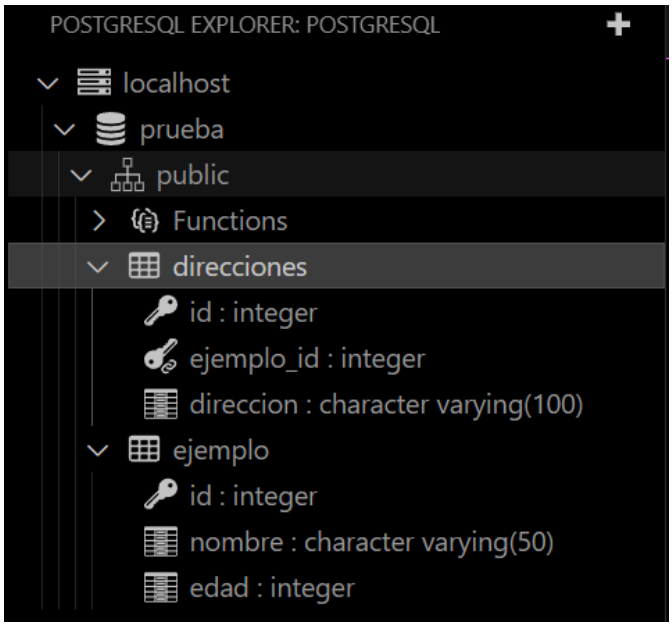
```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín\2nd
 1> javac -cp "mysql-connector-java-9.1.0.jar;." EjemploMySQL.java
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín\2nd
 1> java -cp "mysql-connector-java-9.1.0.jar;." EjemploMySQL
Conexión exitosa a la base de datos 'empresa'.
Consulta 1: Empleados y su departamento
Empleado: Ana, Edad: 30, Salario: 4500.0, Departamento: Ventas
Empleado: Luis, Edad: 28, Salario: 3800.0, Departamento: Recursos Humanos
Empleado: Carlos, Edad: 35, Salario: 5000.0, Departamento: TI

Consulta 2: Salario promedio por departamento
Departamento: Ventas, Salario Promedio: 4500.0
Departamento: Recursos Humanos, Salario Promedio: 3800.0
Departamento: TI, Salario Promedio: 5000.0
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín\2nd
```

Uso de la base de datos en PostgreSQL (Solo desde consola y con VS Code)

Consultas para crear las tablas y algunos datos de prueba:

```
PS C:\Users\jhona\Onpsql -U postgress\University\Universidad Nacional de San Agustín\2nd Year\Segundo Se
Password for user postgres:
psql (17.2)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# CREATE DATABASE prueba;
CREATE DATABASE
postgres=# \c prueba
You are now connected to database "prueba" as user "postgres".
prueba=#
prueba=# CREATE TABLE ejemplo (
prueba(#     id SERIAL PRIMARY KEY,
prueba(#     nombre VARCHAR(50),
prueba(#     edad INT
prueba(# );
CREATE TABLE
prueba=#
prueba=# CREATE TABLE direcciones (
prueba(#     id SERIAL PRIMARY KEY,
prueba(#     ejemplo_id INT REFERENCES ejemplo(id),
prueba(#     direccion VARCHAR(100)
prueba(# );
CREATE TABLE
prueba=#
prueba=# INSERT INTO ejemplo (nombre, edad) VALUES ('Maria', 35), ('Carlos', 40);
INSERT 0 2
prueba=# INSERT INTO direcciones (ejemplo_id, direccion) VALUES (1, 'Calle 123'), (2, 'Avenida 456');
INSERT 0 2
```

Estructura de la base de datos ‘prueba’:



Clase EjemploPostgreSQL.java:

```java
import java.sql.*;

public class EjemploPostgreSQL {
    public static void main(String[] args) {
        String url = "jdbc:postgresql://localhost:5432/prueba";
        String user = "postgres";
        String password = "301005";

        try {
            Class.forName("org.postgresql.Driver");

            Connection conn = DriverManager.getConnection(url, user, password);
            System.out.println("Conexión exitosa a PostgreSQL");

            String query1 = "SELECT * FROM ejemplo";
            try (Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(query1)) {
                while (rs.next()) {
                    System.out.println("ID: " + rs.getInt("id") + ", Nombre: " + rs.getString("nombre") + ", Edad: " + rs.getInt("edad"));
                }
            }

            String query2 = "SELECT e.nombre, d.direccion FROM ejemplo e JOIN direcciones d ON e.id = d.ejemplo_id";
            try (Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(query2)) {
                while (rs.next()) {
                    System.out.println("Nombre: " + rs.getString("nombre") + ", Dirección: " + rs.getString("direccion"));
                }
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San A
> javac -cp "postgresql-42.7.4.jar;." EjemploPostgreSQL.java
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San A
> java -cp "postgresql-42.7.4.jar;." EjemploPostgreSQL
Conexión exitosa a PostgreSQL
ID: 1, Nombre: Maria, Edad: 35
ID: 2, Nombre: Carlos, Edad: 40
Nombre: Maria, Dirección: Calle 123
Nombre: Carlos, Dirección: Avenida 456
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San A
>
```