// ResultSet resultSet = statement.executeQuery("SELECT \* FROM country");

// while (resultSet.next()) {

// System.out.println(resultSet.getString(2));

// System.out.println(resultSet.getString("name"));

// }

// resultSet.close();

}

catch (SQLException ex) {

System.err.println("Exception: " + ex.getMessage());

}

}

(the City.java file has variables for each DB column, and getters and setters for each variable, and a to string override to return all variables as a string with labels for each)

import java.sql.Connection, java.sql.DriverManager, java.sql.ResultSet, java.sql.SQLException, java.sql.Statement, java.util.List;

import daos.CityDao, entities.City, entities.Language, entities.Database;

public class App {

public static void main(String[] args) throws Exception {

System.out.println("Hello, World!");

List<City> cityList;

// String url = "jdbc:mysql://localhost:3306/world";

// String user = "root";

// String password = "8064";

try (Connection connection = Database.getDatabaseConnection();

Statement statement = connection.createStatement();) {

CityDao cityDao = new CityDao(connection);

cityList = cityDao.findAll();

//Cities

System.out.println("Printing Cities");

for(City city: cityList) {

System.out.println(city);

}

//Insert

// City insertCity = new City();

// insertCity.setCountryCode("CAN");

// insertCity.setDistrict("Kings");

// insertCity.setName("Kingston");

// insertCity.setPopulation(136685);

// cityDao.insert(insertCity);

//findById

City city = new City();

city = cityDao.findById(4080);

System.out.println("City returned from findById (4087): " + city);

//Update

city.setPopulation(10000);

Boolean success = cityDao.update(city);

System.out.println("City after the update: " + cityDao.findById(4087));

//Delete

cityDao.delete(4080);

// ResultSet resultSet = statement.executeQuery("SELECT \* FROM country");

public Boolean update(City city) {

Boolean success = true;

String update = "UPDATE city SET population=? WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(update);) {

ps.setInt(1, city.getPopulation());

ps.setInt(2, city.getID());

ps.executeUpdate();

}

catch(SQLException e) {

System.err.println(e.getMessage());

success = false; } return success; }

public Boolean delete(Integer pk) {

Boolean success = false;

String delete = "DELETE FROM city WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(delete)) {

ps.setInt(1, pk);

if(ps.executeUpdate() != 0) {

success = true; } }

catch(SQLException e) { System.err.println(e.getMessage()); } return success; }

public City findById(Integer pk) {

City city = new City();

String select = "SELECT \* FROM city WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(select);) {

ps.setInt(1, pk);

ResultSet result = ps.executeQuery();

if (result.next()) {

city.setCountryCode(result.getString("CountryCode"));

city.setDistrict(result.getString("District"));

city.setID(result.getInt("ID"));

city.setName(result.getString("Name"));

city.setPopulation(result.getInt("Population"));

}

}

catch(SQLException e) {

System.err.println(e.getMessage()); }

return city;

}

}

(CITYDAO) package daos;

import java.sql.Connection, java.sql.PreparedStatement, java.sql.ResultSet, java.sql.SQLException, java.sql.Statement, java.util.ArrayList, java.util.List, entities.City;

public class CityDao implements Dao<City, Integer> {

Connection connection;

public CityDao(Connection connection) {

this.connection = connection;

}

public List<City> findAll() {

List<City>cities = new ArrayList<>();

try(Statement statement = connection.createStatement()) {

ResultSet result = statement.executeQuery("SELECT \* FROM city");

while (result.next()) {

City city = new City();

city.setCountryCode(result.getString("CountryCode"));

city.setDistrict(result.getString("District"));

city.setID(result.getInt("ID"));

city.setName(result.getString("Name"));

city.setPopulation(result.getInt("Population"));

cities.add(city); } }

catch(SQLException e) { System.err.println(e.getMessage()); } return cities; }

public void insert(City city) {

try(Statement statement = connection.createStatement()) {

String insert = "INSERT INTO city VALUES (?, ?, ?, ?, ?)";

PreparedStatement ps = connection.prepareStatement(insert, Statement.RETURN\_GENERATED\_KEYS);

ps.setString(1, null);

ps.setString(2, city.getName());

ps.setString(3, city.getCountryCode());

ps.setString(4, city.getDistrict());

ps.setInt(5, city.getPopulation());

ps.executeUpdate();

ResultSet keys = ps.getGeneratedKeys();

if(keys.next()) {

city.setID(keys.getInt(1));

}

}

catch(SQLException e) {

System.err.println(e.getMessage());

}

}

/\*\*

\* pass in a city object to update the object this is run on to the values of the passed object

\* returns true or false for whether it succeded or failed

\*/

public Boolean update(City city) {

Boolean success = true;

String update = "UPDATE city SET population=? WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(update);) {

ps.setInt(1, city.getPopulation());

ps.setInt(2, city.getID());

ps.executeUpdate();

}

catch(SQLException e) {

System.err.println(e.getMessage());

success = false;

}

return success;

}

/\*\*

\* pass the primary key of the row you want to delete from the city table

\*/

public Boolean delete(Integer pk) {

Boolean success = false;

String delete = "DELETE FROM city WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(delete)) {

ps.setInt(1, pk);

if(ps.executeUpdate() != 0) {

success = true;

}

}

catch(SQLException e) {

System.err.println(e.getMessage());

}

return success;

}

/\*\*

\* returns a city object filled with values retrieved from the table with the passed primary key

\*/

public City findById(Integer pk) {

City city = new City();

String select = "SELECT \* FROM city WHERE id=?";

try(PreparedStatement ps = connection.prepareStatement(select);) {

ps.setInt(1, pk);

ResultSet result = ps.executeQuery();

if (result.next()) {

city.setCountryCode(result.getString("CountryCode"));

city.setDistrict(result.getString("District"));

city.setID(result.getInt("ID"));

city.setName(result.getString("Name"));

city.setPopulation(result.getInt("Population"));

}

}

catch(SQLException e) {

System.err.println(e.getMessage());

}

return city;

}

}