```
package model.dao.impl;

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

import db.DB;
import db.DbException;
import db.DbIntegrityException;
import model.dao.DepartmentDao;
import model.entities.Department;

public class DepartmentDaoJDBC implements DepartmentDao {

        private Connection conn;

        public DepartmentDaoJDBC(Connection conn) {
                this.conn = conn;
        }

        @Override
        public Department findById(Integer id) {
                PreparedStatement st = null;
                ResultSet rs = null;
                try {
                        st = conn.prepareStatement(
                                "SELECT * FROM department WHERE Id = ?");
                        st.setInt(1, id);
                        rs = st.executeQuery();
                        if (rs.next()) {
                                Department obj = new Department();
                                obj.setId(rs.getInt("Id"));
                                obj.setName(rs.getString("Name"));
                                return obj;
                        }
                        return null;
                }
                catch (SQLException e) {
                        throw new DbException(e.getMessage());
                }
                finally {
                        DB.closeStatement(st);
                        DB.closeResultSet(rs);
                }
        }

        @Override
        public List<Department> findAll() {
                PreparedStatement st = null;
                ResultSet rs = null;
                try {
                        st = conn.prepareStatement(
                                "SELECT * FROM department ORDER BY Name");
                        rs = st.executeQuery();

                        List<Department> list = new ArrayList<>();

                        while (rs.next()) {
                                Department obj = new Department();
                                obj.setId(rs.getInt("Id"));
                                obj.setName(rs.getString("Name"));
                                list.add(obj);
                        }
                        return list;
                }
                catch (SQLException e) {
                        throw new DbException(e.getMessage());
                }
                finally {
                        DB.closeStatement(st);
                        DB.closeResultSet(rs);
                }
        }

        @Override
        public void insert(Department obj) {
                PreparedStatement st = null;
                try {
                        st = conn.prepareStatement(
                                "INSERT INTO department " +
                                "(Name) " +
                                "VALUES " +
                                "(?)",
                                Statement.RETURN_GENERATED_KEYS);

                        st.setString(1, obj.getName());

                        int rowsAffected = st.executeUpdate();

                        if (rowsAffected > 0) {
                                ResultSet rs = st.getGeneratedKeys();
                                if (rs.next()) {
                                        int id = rs.getInt(1);
                                        obj.setId(id);
                                }
                        }
                        else {
                                throw new DbException("Unexpected error! No rows
affected!");
                        }
                }
                catch (SQLException e) {
                        throw new DbException(e.getMessage());
                }
                finally {
                        DB.closeStatement(st);
                }
        }

        @Override
        public void update(Department obj) {
                PreparedStatement st = null;
                try {
                        st = conn.prepareStatement(
                                "UPDATE department " +
                                "SET Name = ? " +
                                "WHERE Id = ?");

                        st.setString(1, obj.getName());
                        st.setInt(2, obj.getId());

                        st.executeUpdate();
                }
                catch (SQLException e) {
                        throw new DbException(e.getMessage());
                }
                finally {
                        DB.closeStatement(st);
                }
        }

        @Override
        public void deleteById(Integer id) {
                PreparedStatement st = null;
                try {
                        st = conn.prepareStatement(
                                "DELETE FROM department WHERE Id = ?");

                        st.setInt(1, id);

                        st.executeUpdate();
                }
                catch (SQLException e) {
                        throw new DbIntegrityException(e.getMessage());
                }
                finally {
                        DB.closeStatement(st);
                }
        }
}
```