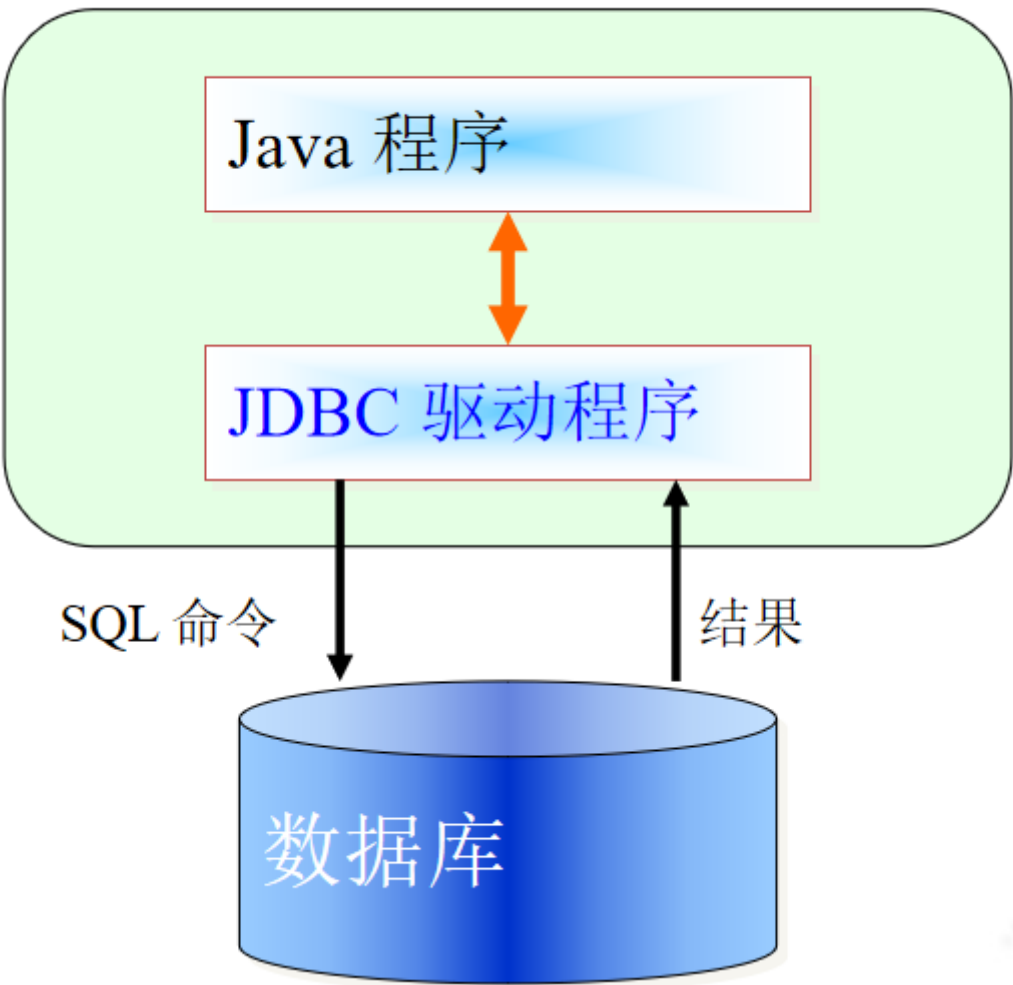


1、什么是JDBC

JDBC (Java DataBase Connectivity) 是由Sun Microsystem公司提供的API (Application Programming Interface应用程序编程接口) ；

- 它为Java应用程序提供了一系列的类，使其能够快速高效地访问数据库；
- 这些功能是由一系列的类和对象来完成的，我们只需使用相关的对象，即可完成对数据库的操作；
- 这些类和接口都是在 `import java.sql.*;` 包下的。



2、JDBC 提供的 类与接口

- `java.sql`包中的一些接口

接口名称	说 明
Connection	连接对象，用于与数据库取得连接
Driver	用于创建连接（Connection）对象
Statement	语句对象，用于执行SQL语句，并将数据检索到结果集（ResultSet）对象中
PreparedStatement	预编译语句对象，用于执行预编译的SQL语句，执行效率比Statement高
CallableStatement	存储过程语句对象，用于调用执行存储过程
ResultSet	结果集对象，包含执行SQL语句后返回的数据的集合

PreparedStatement 继承自 Statement

PreparedStatement 执行效率比 Statement 要

PreparedStatement 有预编译的机制，可以避免sql注入的问题

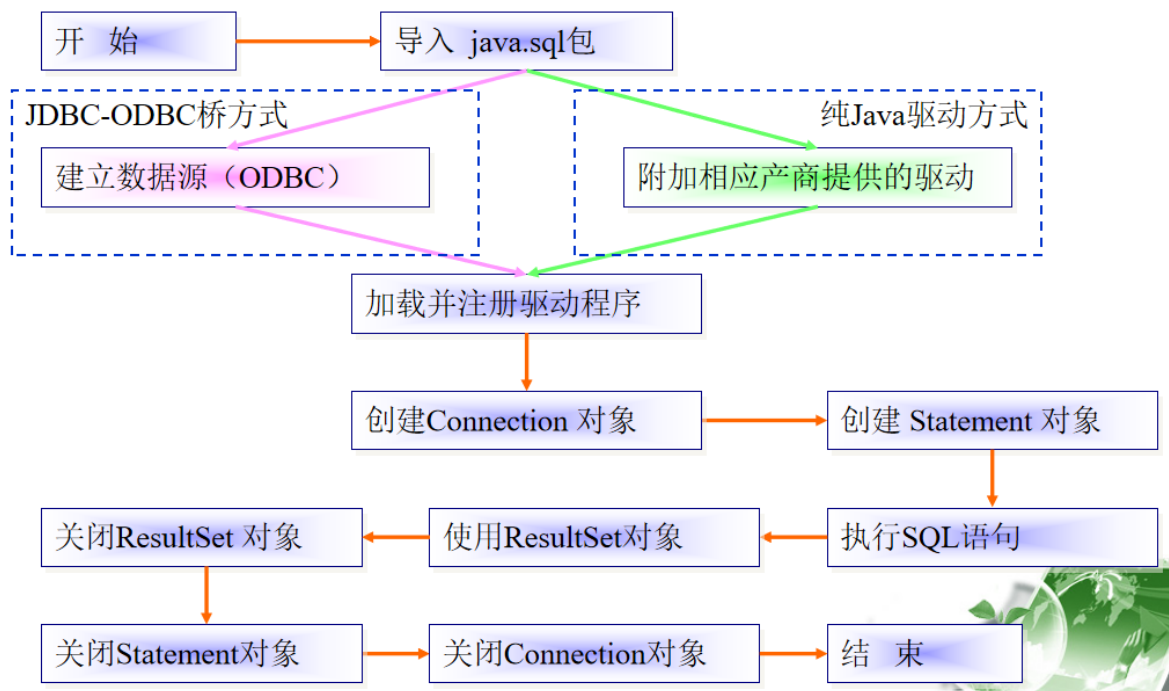
DQL 语句：select

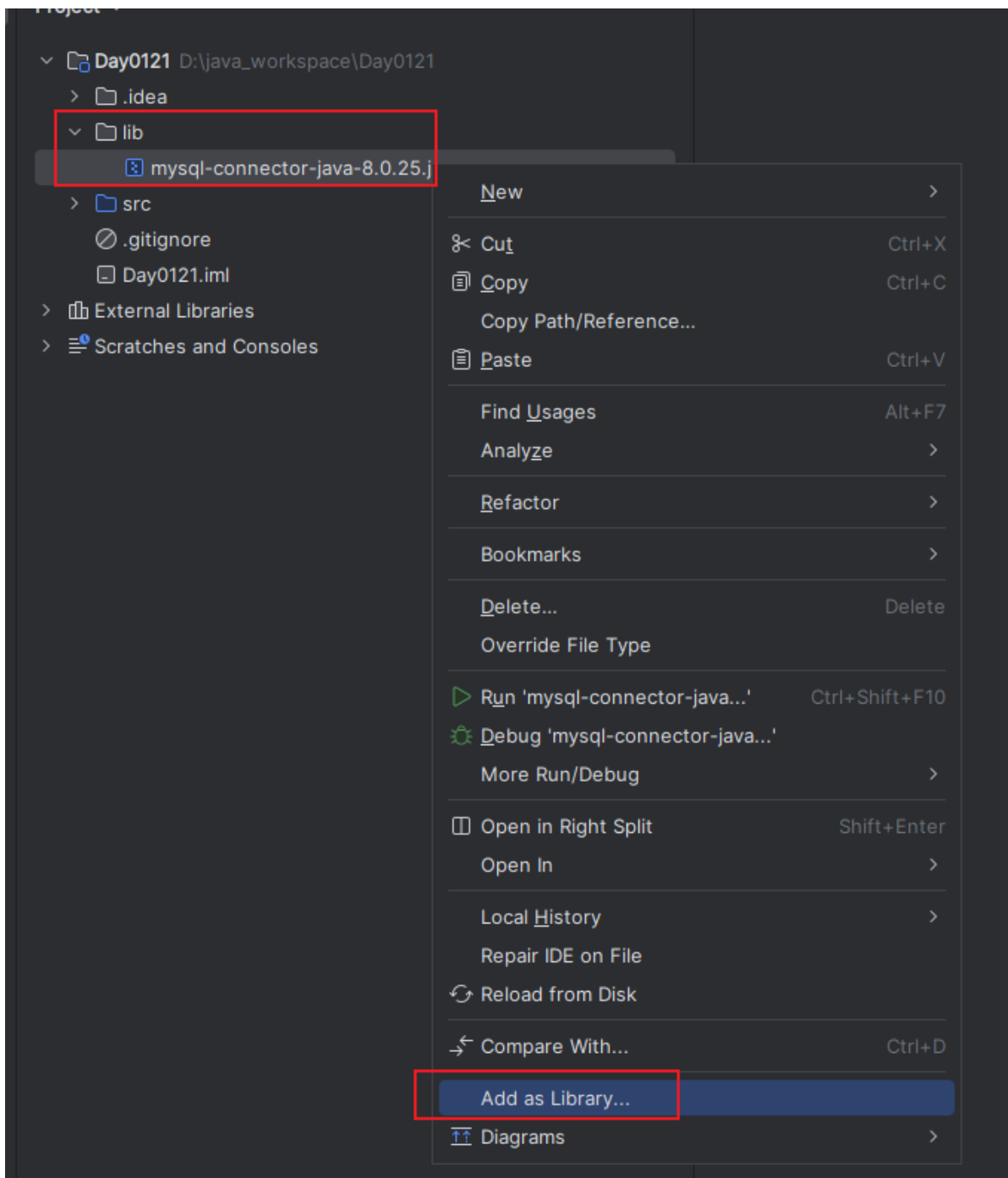
DML 语句：insert delete update

- java.sql包中的一些类

类 名 称	说 明
SQLException	数据库异常类，是其它JDBC异常类的根类，继承于java.lang.Exception，绝大部分对数据库进行操作的方法都有可能抛出该异常
DriverManager	驱动程序管理类，用于加载和卸载各种驱动程序，并建立与数据库的连接
Date	该类中包含有将SQL日期格式转换成Java日期格式的方法
TimeStamp	表示一个时间戳，能精确到纳秒

3、JDBC执行步骤





- 1、加载驱动程序

- `Class.forName("com.mysql.cj.jdbc.Driver");`

- 2、创建Connection对象

- ```
// localhost 本地 (127.0.0.1、IP地址)
// 3306 mysql数据库的端口号
String url = "jdbc:mysql://localhost:3306/数据库名";
// 用户名
String user = "root";
// 密码
String password = "root";
Connection conn = DriverManager.getConnection(String url, String
user,String password);
```

- 3、创建PreparedStatement对象

- ```
// Statement sta = conn.createStatement();
// DML 语句
// int rows = sta.executeUpdate(String sql)
// DQL 语句
// ResultSet rs = sta.executeQuery(String sql)

// ? 可以理解为是预编译
PreparedStatement ps = conn.prepareStatement("update 表名 set
first_name=?,salary=? where empid=?");
```

- 4、执行sql 语句

- ```
// DML 语句
// 替换 ? ps.setXxx(x,y) 第一个参数? 的位置，第二个参数是替换后的值
ps.setString(1,"张三");
ps.setInt(2,8000);
ps.setInt(3,100);
int rows = ps.executeUpdate();

// DQL 语句
ResultSet rs = ps.executeQuery();
// next() 判断结果集中是否有下一行记录，有，返回true
while(rs.next()){
 // 获取每一列（字段）的值
 int empid = rs.getInt("employee_id"); // 根据字段名获取值
 String first_name = rs.getString(2); // 根据字段的位置获取（位置是从1开始的）
}
```

- 5、关闭资源

- 先开的后关,后开的先关
- ```
// 关闭Result
rs.close();
// 关闭PreparedStatement、
ps.close();
// 关闭Connection
conn.close();
```

4、连接数据库的工具类

```
package com.zretc3.util;

import java.sql.*;

public class DBUtil {
    private static Connection conn;
    private static PreparedStatement ps;
    private static ResultSet rs;

    // 获取数据库连接对象
```

```

    public static Connection getConn() {
        try {
            // 加载驱动
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root", "root");
            return conn;
        } catch (ClassNotFoundException | SQLException e) {
            throw new RuntimeException(e);
        }
    }

    // 关闭资源
    public static void closeAll(ResultSet rs, PreparedStatement ps, Connection
conn) {
        try {
            if (rs != null) {
                rs.close();
            }
            if (ps != null) {
                ps.close();
            }
            if (conn != null) {
                conn.close();
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}

```

5、MVC设计模式

M 层： Model 模型

pojo (实体类) -- 一张表对应一个实体类

dao (增删改查)

service (业务处理)

V 层： View 视图

html、jsp

C 层： Controller 控制层

在Controller中访问service,在service中访问dao

