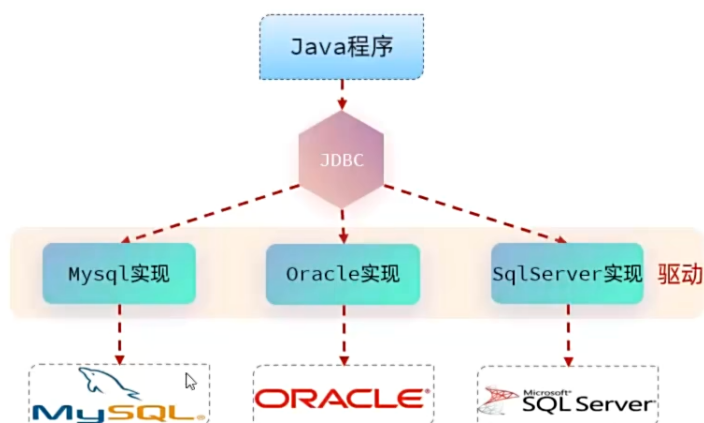


06 Java程序操作数据库JDBC

• 1 JDBC介绍

- JDBC: (Java DataBase Connectivity), 就是使用Java语言操作关系型数据库的一套API。



```
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <version>8.0.33</version>
</dependency>
```

● 本质:

- sun公司官方定义的一套操作所有关系型数据库的规范, 即接口。
- 各个数据库厂商去实现这套接口, 提供数据库驱动jar包。
- 我们可以使用这套接口 (JDBC) 编程, 真正执行的代码是驱动jar包中的实现类。

• 2 JDBC - 入门程序

- 需求: 基于JDBC程序, 执行update语句 (`update user set age = 25 where id = 1`)

● 步骤:

- 准备工作: 创建一个maven项目, 引入依赖; 并准备数据库表user。
- 代码实现: 编写JDBC程序, 操作数据库

```
pom.xml
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <version>8.0.33</version>
</dependency>
```

```
JDBC程序
//1. 注册驱动
Class.forName("com.mysql.cj.jdbc.Driver");
//2. 获取连接
String url = "jdbc:mysql://localhost:3306/web01";
String username = "root";
String password = "1234";
Connection connection = DriverManager.getConnection(url, username, password);
//3. 获取SQL语句执行对象
Statement statement = connection.createStatement();
//4. 执行SQL
int i = statement.executeUpdate("update user set age = 25 where id = 1");
//5. 释放资源
statement.close();
connection.close();
```

• 3 JDBC - 查询数据

- 需求: 基于JDBC执行如下select语句, 将查询结果封装到User对象中。
- SQL: `select * from user where username = 'daqiao' and password = '123456'`

- **ResultSet (结果集对象)** : `ResultSet rs = statement.executeQuery()`
 - ◆ `next()`: 将光标从当前位置向前移动一行, 并判断当前行是否为有效行, 返回值为boolean。
 - `true`: 有效行, 当前行有数据
 - `false`: 无效行, 当前行没有数据
 - ◆ `getXxx(...)`: 获取数据, 可以根据列的编号获取, 也可以根据列名获取 (推荐)。

● 结果解析步骤:

```
while (resultSet.next()) {
    int id = resultSet.getInt("id");
    //...省略
}
```

	id	username	password	name
1	1	daqiao	1234567890	大乔
2	2	xiaoqiao	123456	小乔
3	3	diaochan	123456	貂蝉
4	4	lvbu	123456	吕布
5	5	zhaoyun	12345678	赵云

1. JDBC程序执行DML语句? DQL语句?

- DML语句: `int rowsAffected = statement.executeUpdate();`
- DQL语句: `ResultSet rs = statement.executeQuery();`

2. DQL语句执行完毕结果集ResultSet解析?

- `resultSet.next()`: 光标往下移动一行
- `resultSet.getXxx()`: 获取字段数据

● 4 预编译

1. 静态SQL (参数硬编码)

```
Statement statement = connection.createStatement();
int i = statement.executeUpdate("update user set age = 25 where id = 1");
System.out.println("SQL执行完毕, 影响的记录数为: " + i);
```

2. 预编译SQL (参数动态传递)

```
PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM user WHERE username = ? AND password = ?");
pstmt.setString(1, "daqiao");
pstmt.setString(2, "123456");
ResultSet resultSet = pstmt.executeQuery();
```

安全
性能更高

- 优势一：可以防止SQL注入，更安全

SQL注入：通过控制输入来修改事先定义好的SQL语句，以达到执行代码对服务器进行攻击的方法。



- 优势二：性能更高

