JDBC 概念

- JDBC 他是一个协议 也就是接口.
- JDBC 替我们处理了 数据库链接底层的逻辑
- 当我们对接不同的数据库厂商的时候, 不需要单独写一套代码

JDBC 预编译 API

```
String url = "jdbc:mysql:///db1?useSSL=false"; //定义数据库地址
String username = "root";     //数据库用户名
String password = "1234";     //数据库密码
Connection conn = DriverManager.getConnection(url, username, password);  //发起链接
```

```
//插入数据
String brandName = "香飘飘奶茶";
String companyName = "奶茶";
int ordered = 30;
String description = "一年绕地球一周";
int status = 2;
Connection connection = DriverManager.getConnection(url, user, passwd);

String insertSql = "INSERT INTO tb_brand (brand_name, company_name, ordered, description, status) values (?,?,?,?,?)";
PreparedStatement preparedStatement = connection.prepareStatement(insertSql);
preparedStatement.setString(1, brandName);
preparedStatement.setString(2, companyName);
preparedStatement.setInt(3, ordered);
preparedStatement.setString(4, description);
preparedStatement.setInt(5, status);
int i = preparedStatement.executeUpdate();
if ( i > 0 ) {
    System.out.println("插入成功");
} else {
    System.out.println("插入失败");
}
```

```
//查询数据
```

```java
String brandName = "三只";
Connection connection = DriverManager.getConnection(url, user, passwd);
String selectSql = "SELECT * FROM tb_brand WHERE brand_name LIKE ?";
PreparedStatement preparedStatement = connection.prepareStatement(selectSql);
preparedStatement.setString(1, "%" + brandName + "%");
ResultSet resultSet = preparedStatement.executeQuery();
while ( resultSet.next() ) {
    Brand brand = new Brand();
    brand.setId(resultSet.getInt("id"));
    brand.setBrandName(resultSet.getString("brand_name"));
    brand.setCompanyName(resultSet.getString("company_name"));
    brand.setOrdered(resultSet.getInt("ordered"));
    brand.setDescription(resultSet.getString("description"));
    brand.setStatus(resultSet.getInt("status"));
    System.out.println( brand );
}
```

```java
//更新数据
String brandName = "香飘飘奶茶";
String companyName = "奶茶";
int ordered = 30;
String description = "一年绕地球一周";
int status = 2;
int Id = 2;
Connection connection = DriverManager.getConnection(url, user, passwd);
String insertSql = "UPDATE tb_brand SET brand_name=?,company_name=?,ordered=?,description=?,status=? WHERE id = ?";
PreparedStatement preparedStatement = connection.prepareStatement(insertSql);
preparedStatement.setString(1, brandName);
preparedStatement.setString(2, companyName);
preparedStatement.setInt(3, ordered);
preparedStatement.setString(4, description);
preparedStatement.setInt(5, status);
preparedStatement.setInt(6, Id);
int i = preparedStatement.executeUpdate();
if ( i > 0 ) {
    System.out.println("修改成功");
} else {
    System.out.println("修改失败");
}
```

```java
//删除数据
int Id = 2;
Connection connection = DriverManager.getConnection(url, user, passwd);
String insertSql = "DELETE from tb_brand WHERE id = ?";
PreparedStatement preparedStatement = connection.prepareStatement(insertSql);
preparedStatement.setInt(1, Id);
int i = preparedStatement.executeUpdate();
if ( i > 0 ) {
    System.out.println("删除成功");
} else {
    System.out.println("删除失败");
}
```

数据库链接池

- 使用链接池 可以重复使用链接
- 提升相应速递 减少没必要的 TCP 握手
- 统一管理链接 避免数据库链接 遗漏 (忘记释放资源)

```java
Properties properties = new Properties();
properties.load( new FileInputStream("day03/src/druid.properties"));
DataSource dataSource = DruidDataSourceFactory.createDataSource(properties);
String brandName = "香飘飘奶茶";
String companyName = "奶茶";
int ordered = 30;
String description = "一年绕地球一周";
int status = 2;
String insertSql = "INSERT INTO tb_brand (brand_name, company_name, ordered, description, status) values (?,?,?,?,?)";
Connection connection = dataSource.getConnection();
PreparedStatement preparedStatement = connection.prepareStatement(insertSql);
preparedStatement.setString(1, brandName);
preparedStatement.setString(2, companyName);
preparedStatement.setInt(3, ordered);
preparedStatement.setString(4, description);
preparedStatement.setInt(5, status);
int i = preparedStatement.executeUpdate();
```

```java
if ( i > 0 ) {
    System.out.println("插入成功");
} else {
    System.out.println("插入失败");
}
preparedStatement.close();
```

```properties
//数据库连接池 配置档
driverClassName=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/vip21?useSSL=false&useServerPrepStmts=true
username=root
password=123456
#初始化链接库池大小
initialSize=5
#最大连接数
maxActive=10
#最大等待时间
maxWait=3000
```