

MyBatis

ORM

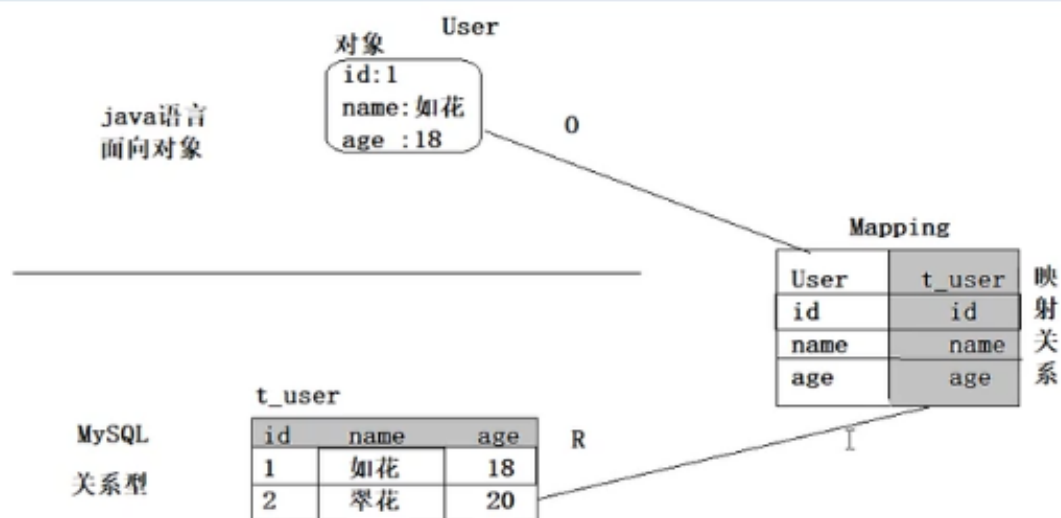
(Object Relational Mapping) 对象关系映射

对象指面向对象

关系指关系型数据库

java到MySQL的映射，开发者可以以面对对象的思想来管理数据库

ORM的图解：



简介

MyBatis 本是apache的一个开源项目iBatis, 2010年这个项目由apache software foundation 迁移到了google code, 并且改名为MyBatis。

2013年11月迁移到Github。MyBatis是一个优秀的持久层框架，它对jdbc的操作数据库的过程进行封装，使开发者只需要关注 SQL 本身，而不需要花费精力去处理例如注册驱动、创建connection、创建statement、手动设置参数、结果集检索等jdbc繁杂的过程代码。Mybatis通过xml或注解的方式将要执行的各种statement (statement、preparedStatemnt、CallableStatement) 配置起来，并通过java对象和statement中的sql进行映射生成最终执行的sql语句，最后由mybatis框架执行sql并将结果映射成java对象并返回。

- [mybatis官方文档地址](#)

为什么使用mybatis

jdbc连接数据库的连接方法

```

public static void main(String[] args) {
    Connection connection = null;
    PreparedStatement preparedStatement = null;
    ResultSet resultSet = null;

    try {
        //1、加载数据库驱动
        Class.forName("com.mysql.jdbc.Driver");
        //2、通过驱动管理类获取数据库链接
        connection =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/mybatis", "root",
"root");

        //3、定义sql语句 ?表示占位符
        String sql = "select * from user where username = ?";
        //4、获取预处理statement
        preparedStatement = connection.prepareStatement(sql);
        //5、设置参数，第一个参数为sql语句中参数的序号（从1开始），第二个参数为设置
的参数值
        preparedStatement.setString(1, "王五");
        //6、向数据库发出sql执行查询，查询出结果集
        resultSet = preparedStatement.executeQuery();
        //7、遍历查询结果集
        while(resultSet.next()){
            User user
            System.out.println(resultSet.getString("id")+"
"+resultSet.getString("username"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally{
        //8、释放资源
        if(resultSet!=null){
            try {
                resultSet.close();//释放结果集
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        if(preparedStatement!=null){
            try {
                preparedStatement.close();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        if(connection!=null){
            try {
                connection.close();//关闭数据库连接
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

```
}
```

使用jdbc的缺陷

- 代码比较多，开发效率低
- 需要关注对象创建和销毁
- 对ResultSet查询的结果，需要自己封装为List
- 重复代码比较多
- 业务代码和数据库操作混在一起

开发人员要做的是：提供SQL语句

mybatis相对jdbc的优势：

- 1.在创建connection的时候，存在硬编码问题（也就是直接把连接信息写死，不方便后期维护）
- 2.preparedStatement对象在执行sql语句的时候存在硬编码问题。
- 3.每次在进行一次数据库连接后都会关闭数据库连接，频繁的开启/关闭数据连接影响性能。

简单的说一下mybatis相对jdbc的优势：

- 1.mybatis是把连接数据库的信息都是写在配置文件中，因此不存在硬编码问题，方便后期维护。
- 2.mybatis执行的sql语句都是通过配置文件进行配置，不需要写在java代码中。
- 3.mybatis的连接池管理、缓存管理等让连接数据库和查询数据效率更高。

总结：

mybatis是一个sql映射框架，提供对数据库的操作能力。可以看做是增强的JDBC，使开发人员只需要关注sql。