

- 1 常见面试问题
 - 1.1 问题1
 - 1.2 问题2
 - 1.3 问题3
- 2 关于ResultMap
 - 2.1 关于resultMap标签
 - 2.2 一对多查询示例
 - 2.3 关于ResultMap练习
 - 2.3.1 练习1
 - 2.3.2 练习2

1 常见面试问题

1.1 问题1

MyBatis中，#{} 和 \${} 的区别是什么？

#{ } 和 \${ } 都是 MyBatis 框架中的占位符，区别在于执行 SQL 语句的方式不同；

- \${ } 是以字符串拼接的方式执行 SQL 语句，存在 SQL 注入攻击的风险；

```
# 需要执行的SQL语句：
DELETE FROM user WHERE id=值
# 用户输入：
1 OR 1=1
# 导致表中所有的数据全都被删除。
```

- #{ } 是预编译 SQL 的方式执行，不存在 SQL 注入攻击的风险，工作中常用。

1.2 问题2

介绍一下数据的三范式

减少数据的冗余.

- 1NF：拆字段

字段是最小的单元不能再拆分；

用户编号	用户名	密码	省	市	区县	街道
1	赵丽颖	123456	河北省	廊坊市	霸州市	幸福大街1号

- 2NF：拆表

满足1NF，表中的字段必须依赖于全部主键而非部分主键.

丽丽系列双肩包，价格是99元

商品ID	商品名称	分类ID	分类名称	价格	仓库	仓库负责人
1	丽丽	1	双肩包	99	A	传奇大爷

商品ID	商品名称	分类ID	分类名称	价格	仓库	仓库负责人
1	丽丽	2	单肩包	88	A	传奇大爷
2	颖颖	1	双肩包	66	B	克晶姐姐

上表不满足第2NF，需要拆表

◦ 商品表

商品ID	商品名称	仓库	仓库负责人	销量
1	丽丽	A	传奇大爷	300

◦ 分类表

分类ID	分类名	销量
1	双肩包	100

◦ 价格表

商品ID	分类ID	价格	销量
1	1	99	100
1	2	88	200

• 3NF-拆表

满足2NF，表中的非主键字段之间不能有传递性依赖。

商品ID	商品名称	仓库	仓库负责人
1	丽丽	B	传奇大爷

上表中，仓库和仓库负责人存在传递性依赖，所以拆表

商品表

商品ID	商品名称	仓库
1	丽丽	B
2	颖颖	B

仓库表

仓库	仓库负责人
A	传奇大爷
B	克晶姐姐

1.3 问题3

介绍一下数据库的反范式

- 通过冗余数据，来提高数据的查询效率；
比如在商品表、分类表、价格表中都冗余 **销量** 字段，增加不同维度的数据查询效率。
- 通过冗余数据，避免数据失去真实性。

- 收货地址表

地址编号	地址名称
------	------

1	大理洱海中央大船
---	----------

- 订单表

订单编号	订单时间	收货地址
------	------	------

001	2020	北京
-----	------	----

002	2021	北京
-----	------	----

003	2023	大理洱海中央大船
-----	------	----------

004	2024	大理洱海中央大船
-----	------	----------

2 关于ResultMap

2.1 关于resultMap标签

查询结果与Java对象属性之间映射关系的一种配置方式，一般用于一对多和多对多的复杂查询。

- `<resultMap>` 标签
 - `id` 属性：唯一标识
 - `type` 属性：指定映射的JAVA类型

- `<id>` 标签

指定映射的 **主键** 字段，包含：

- `column` 属性：查询语句中的列名（或别名）；
- `property` 属性：Java类中的属性名

- `<result>` 标签

指定映射的 **非主键** 字段，包含

- `column属性`：查询语句中的列名（或别名）
- `property属性`：JAVA类中的属性名

- `<collection>` 标签

映射一对多或多对多关系

- `property属性`：JAVA类中的属性名
- `ofType属性`：集合中元素的类型

2.2 一对多查询示例

查询指定用户发布的所有的微博信息, 用户id,用户名, 微博所有字段信息

- 第1步: 封装VO类

```

public class ResultMapVO2 {
    //用户id、用户昵称、该用户发布的所有微博信息 List<Weibo>
    private Long id;
    private String nickname;
    private List<Weibo> weiboList;

    //set get toString
}

```

- 第2步: XML配置SQL

注意: 因为用户表和微博表都有 id 字段, 所以在SQL语句中查询时需要使用别名, 否则会导致无法在 resultMap 标签中映射.

```

<!--ResultMap多表查询: 查询指定用户发布的所有微博信息
    1.<id>:主键
    2.<result>:非主键
    3.column属性:SQL语句查询字段名[可以为别名]
    4.property属性:JAVA类属性名
    5.<collection>标签:映射集合类型属性,主要应用于一对多和多对多
    6.ofType属性:集合中的JAVA类型
-->
<resultMap id="bbb" type="cn.tedu._04mybatis.pojo.vo.ResultMapVO2">
    <id column="uid" property="id"/>
    <result column="nickname" property="nickname"/>
    <collection property="weiboList"
ofType="cn.tedu._04mybatis.pojo.entity.Weibo">
        <id column="wid" property="id"/>
        <result column="content" property="content"/>
        <result column="created" property="created"/>
        <result column="user_id" property="userId"/>
    </collection>
</resultMap>
<!-- 查询多张表中相同字段名时, 注意使用别名,比如: u.id uid, w.id wid-->
<select id="selectMap2" resultMap="bbb">
    SELECT u.id uid,
    u.nickname,
    w.id wid,
    w.content,
    w.created,
    w.user_id
    FROM user u
    JOIN weibo w ON u.id = w.user_id
    WHERE u.id = #{uid}
</select>

```

- 第3步: 结果展示

```

ResultMapVO2{id=100, username='Lucy', weiboList=[weibo{id=200, content='lucy
的第1条微博', created=Sat Jan 01 00:00:00 CST 2000, userId=100}, weibo{id=201,
content='lucy的第2条微博', created=Sat Jan 01 00:00:00 CST 2000, userId=100}]}

```

2.3 关于ResultMap练习

2.3.1 练习1

- 获取指定微博的所有评论信息：微博ID、微博内容、评论集合List
- 步骤
 - 第1步：创建VO类[ResultMapVO3]
 - 第2步：接口方法[selectMap3(int wid)]
 - 第3步：xml配置SQL
 - 第4步：测试[selectMap3Test(){}]

2.3.2 练习2

- 获取指定微博的所有评论信息：微博ID、微博内容、评论ID、评论内容、评论时间；
将评论ID内容和时间放到List集合中。