**1、本章面试题**

  说一说 mybaits动态sql的用法

说说数据库中一对一，一对多，多对一，多对多的关系，在mybatis中如何处理

**2、知识点**

**2.1、课程回顾**

             什么是orm

mybatis 的工作原理

配置文件中一些问题

**2.2、本章重点**

         1，动态sql的使用

2, 高级结果映射（一对一，一对多，多对一，多对多）

**3、具体内容**

**3.1 Mybatis动态SQL**

**什么是动态SQL:**

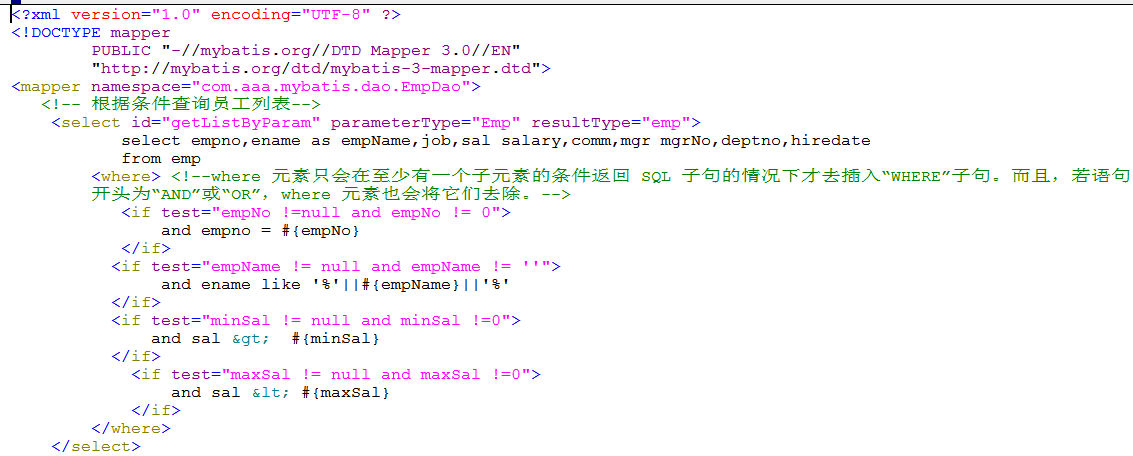
MyBatis 的强大特性之一便是它的动态 SQL。如果你有使用 JDBC 或其它类似框架的经验，你就能体会到根据不同条件拼接 SQL 语句的痛苦。例如拼接时要确保不能忘记添加必要的空格，还要注意去掉列表最后一个列名的逗号。利用动态 SQL 这一特性可以彻底摆脱这种痛苦。

**3.1.1 if**

EmpDao接口



EmpMapper

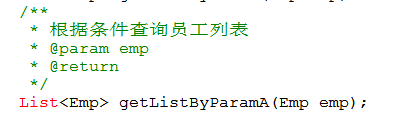


EmpDaoTest

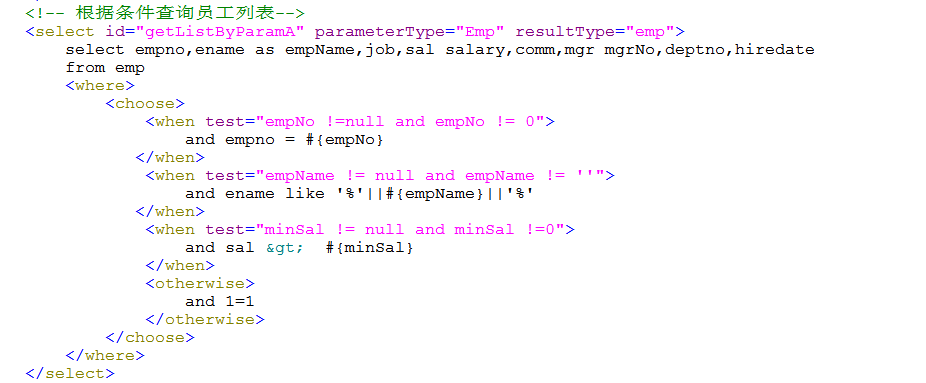


**3.1.2 choose (when, otherwise)**

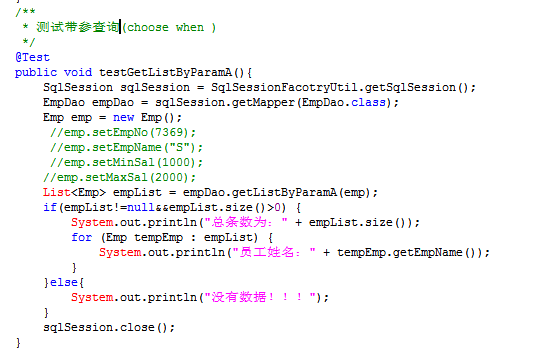
EmpDao接口



EmpMapper

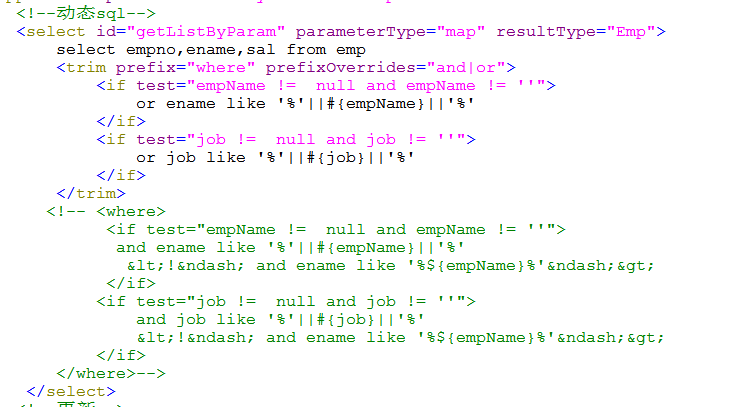


EmpDaoTest

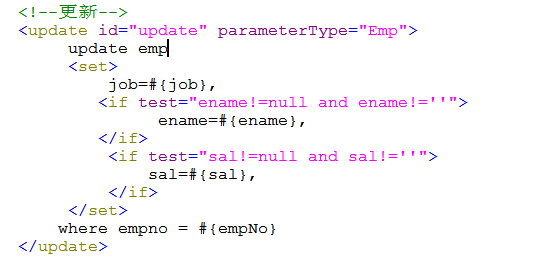


**3.1.3 trim set**

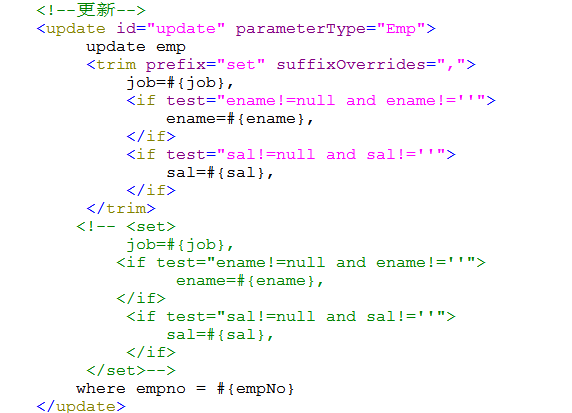
trim prefixOveverides



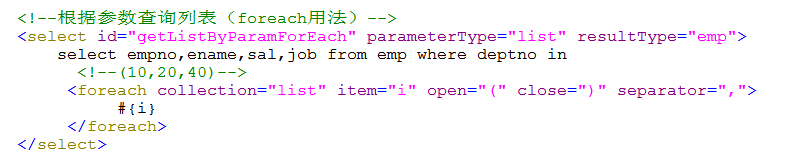
set



trim suffixOveverides



**3.1.4 foreach**





**3.2 高级结果映射（关联查询）**

**3.2.1 主要知识点**

数据库中表和表之间的关系，主要有：一对一 一对多 多对一和多对多

resultMap 属性和标签 用法

id result标签

association 一对一，多对一的时候，使用

collection 一对多，多对多的时候，使用

**3.2.2 一对多，多对多的实例（**collection**）：**

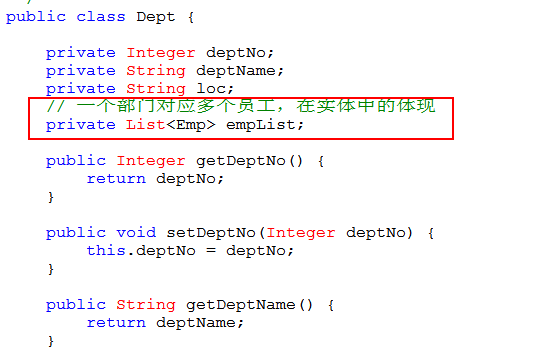
**一对多：**

需求：查询所有部门，在显示部门名称和位置时，显示该部门下的所有人员数量及其人员具体信息（如果说部门下没有人，也显示）

1，数据库语句编写

select d.dname,d.loc,e.\* from dept d left join emp e on d.deptno=e.deptno;

2，编写实体，在java实例中要体现数据库一对多的关系（实例创建和以前一样的）



**3，编写DeptDao接口方法和mapper 文件**



**4，测试类编写**



**多对多：**

需求：查询员工表集合，同时显示该员工对应的角色集合（没有角色的员工也要显示）

**1，创建表，编写查询语句**

-- 创建用户表

create table tb\_user(

user\_id number(10) primary key,

user\_name varchar2(20) not null,

real\_name varchar2(30),

gender number(1) default 0,

age number(3)

);

-- 创建角色表

create table tb\_role(

role\_id number(10) primary key,

role\_name varchar2(20),

describe varchar2(50)

);

-- 多对多，所以一定要有中间表

create table tb\_user\_role(

user\_id number(10) not null,

role\_id number(10) not null,

constraint fk\_userid\_user\_userid foreign key(user\_id) references tb\_user(user\_id),

constraint fk\_roleid\_role\_roleid foreign key(role\_id) references tb\_role(role\_id)

);

-- 插入模拟数据

insert into tb\_user values(1001,'zhangsan1','张三1',0,20);

insert into tb\_user values(1002,'zhangsan2','张三2',1,30);

insert into tb\_user values(1003,'zhangsan3','张三3',0,40);

insert into tb\_user values(1004,'zhangsan4','张三4',1,50);

insert into tb\_role values(101,'副总1','高管1，时间自由');

insert into tb\_role values(102,'副总2','高管2，时间自由');

insert into tb\_role values(103,'副总3','高管3，时间自由');

insert into tb\_user\_role values(1001,101);

insert into tb\_user\_role values(1001,102);

insert into tb\_user\_role values(1002,102);

insert into tb\_user\_role values(1002,103);

insert into tb\_user\_role values(1004,101);

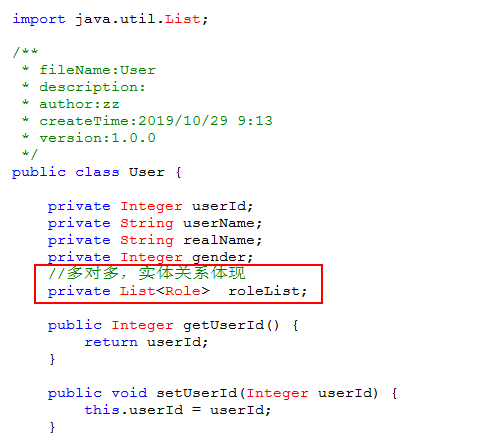
insert into tb\_user\_role values(1004,103);

-- 查询语句

select u.user\_id,u.user\_name,u.real\_name,u.gender,r.role\_id,r.role\_name,r.describe from tb\_user u left join tb\_user\_role ur on u.user\_id=ur.user\_id

left join tb\_role r on ur.role\_id=r.role\_id;

**2，编写实体，在java实例中要体现数据库多对多的关系（其他实体一样）**



**3，编写DeptDao接口方法和mapper 文件**





**4， 测试类编写**



**3.2.3 多对一 一对一**

需求，查询员工列表时，显示该员工的部门信息，即使没有部门也显示

**多对一：**

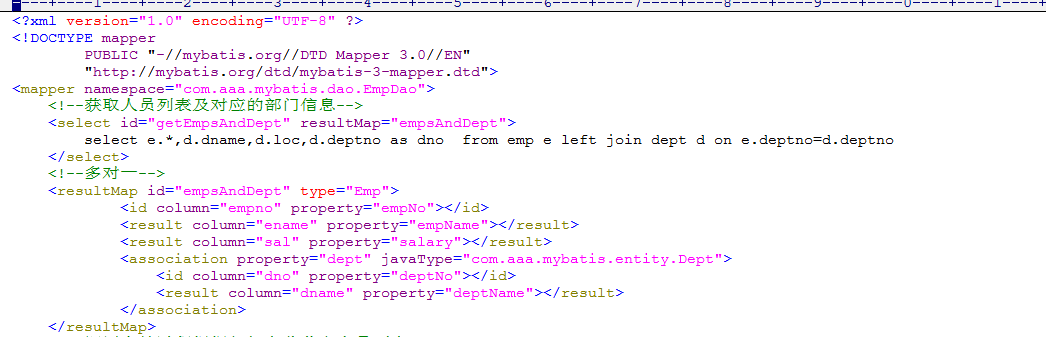
**1，根据需求编写查询语句**

select e.\*,d.dname,d.loc,d.deptno as dno from emp e left join dept d on e.deptno=d.deptno

**2，编写实体，在java实例中要体现数据库多对一的关系**



**3，编写DeptDao接口方法和mapper 文件**

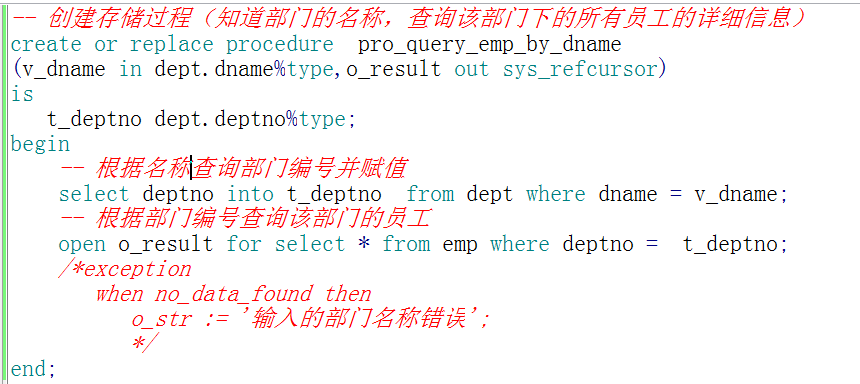


**4， 测试类编写**

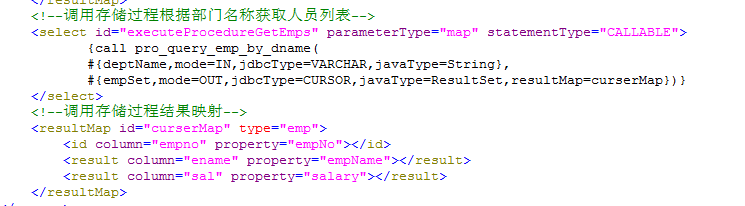


**3.3 调用存储过程**

1，编写存储过程（知道部门的名称，查询该部门下的所有员工的详细信息）



2，编写调用存储过程的mapper



3, 编写测试类



**4、本章总结**

**4.1 总结本章知识点**

**4.2 面试题答案**

**4.3 预习下一章重点**