**公共部分：**

所有接口中必须含以下五个方法

1、删除方法：

int deleteByPrimaryKey(Integer id);

2、添加方法

int insert(Class ?);

3根据ID查询方法

Category selectByPrimaryKey(Integer id)

4、查询全部方法

List<Class> selectAll();

5、修改方法

int updateByPrimaryKey(Class ?);

**私有部分：**

**渲染入下拉框使用i开头 方法必须放在最后**

1. 管理员Admin

根据a\_username修改密码int updatePassword(a\_username,a\_password)

update Admin set a\_password=#{a\_password} where a\_username=#{a\_username}

根据a\_username判断是否出现过该用户 int selectUserName(a\_username)

select count(\*) from Admin where a\_username=#{a\_username}

根据a\_username修改状态int updateStatus(a\_username,status)

update Admin set status=#{status} where a\_username=#{a\_username}

模糊账号查询编号、账号、状态、备注 List<Admin> selectAllVague(username);

select \* from Admin where a\_username like concat('%',#{a\_username},'%')

根据状态查询编号、账号、状态、备注 List<Admin> selectAllStatus(status);

select \* from Admin where status=#{status}

1. 部门类型DepartmentType

根据id修改顺序 int updateSort(d\_id,d\_sort);

update DepartmentType set d\_sort=#{d\_sort} where d\_id=#{d\_id}

根据d\_id修改状态int updateStatus(d\_id,status);

update DepartmentType set status=#{status} where d\_id=#{d\_id}

根据d\_name判断是否出现过该类型名称 int selectName(d\_name)

select count(\*) from DepartmentType where d\_name=#{d\_name}

根据状态全部查询 List<DepartmentType> selectAllStatus(status);

select \* from DepartmentType where status=#{status}

模糊部门名称全部查询 List<DepartmentType> selectAllVague(d\_name);

select \* from DepartmentType where d\_name like concat('%',#{ d\_name},'%')

根据状态全部查询编号、名称 List<DepartmentType> iselectAllStatus(status);

select d\_id,d\_name from DepartmentType where status=#{status}

1. 权限Jurisdiction

根据j\_id修改权限名称int updateName(j\_id,j\_name);

update Jurisdiction set j\_name=#{j\_name} where j\_id=#{j\_id}

根据j\_id修改状态int updateStatus(j\_id,status);

update Jurisdiction set status=#{status} where j\_id=#{j\_id}

根据状态查询全部 List<Jurisdiction> selectAllStatus(status);

select \* from Jurisdiction where status=#{status}

根据状态查询编号、名称 List<Jurisdiction> iselectAllStatus(status);

select \* from Jurisdiction where status=#{status}

模糊权限名称全部查询 List<Jurisdiction> selectAllVague(j\_name);

select \* from Jurisdictionwhere j\_name like concat('%',#{ j\_name},'%')

1. 班级信息ClassTable

根据c\_id修改权限名称int updateName(j\_id,j\_name)

update ClassTable set c\_name=#{c\_name} where j\_id=#{j\_id}

根据c\_id修改状态int updateStatus(c\_id,status);

update ClassTable set status=#{status} where j\_id=#{j\_id}

模糊查询班主任名字 List<Jurisdiction> selectByName(c\_headmaster\_name);

select \* from ClassTable where and c\_headmaster\_name like concat('%',#{c\_headmaster\_name},'%')

根据状态查询全部 List<Jurisdiction> selectAllStatus(status);

select \* from ClassTable where status=#{status}

根据状态查询编号、班级名称 List<Jurisdiction> iselectAllStatus(status);

select c\_id,c\_name from ClassTable where status=#{status}

1. 会议类型TypeofMeeting

根据t\_id修改状态int updateStatus(t\_id,status);

update TypeofMeeting set status=#{status} where t\_id=#{t\_id}

根据状态查询全部 List<TypeofMeeting> selectAllStatus(status);

select \* from TypeofMeeting where status={#status}

根据状态查询编号、名称 List<Jurisdiction> iselectAllStatus(status);

select t\_id,t\_name from TypeofMeeting where status={#status}

根据t\_id修改顺序 int updateSort(t\_id,t\_sort);

update TypeofMeeting set t\_sort=#{t\_sort} where t\_id=#{t\_id}

模糊名称查询 List<TypeofMeeting> selectAllVague(t\_name);

select \* from TypeofMeeting t\_name like concat('%',#{ t\_name},'%')

1. 职位类型PositionType

根据p\_id修改状态int updateStatus(p\_id,status);

update PositionType set status=#{status} where p\_id=#{p\_id}

根据p\_id修改顺序 int updateSort(p\_id,p\_sort);

update PositionType set p\_sort=#{p\_sort} where p\_id=#{p\_id}

根据状态查询全部 List<PositionType> selectAllStatus(status);

select \* from PositionType where status={#status}

根据状态查询编号、名称 List<PositionType> iselectAllStatus(status);

select p\_id,p\_name from PositionType where status={#status}

模糊名称查询 List<PositionType> selectAllVague(p\_name);

select \* from PositionType p\_name like concat('%',#{ p\_name},'%')

1. 会名（十六届学生会）ExchangeTable

根据e\_id修改状态int updateStatus(e\_id,status);

update ExchangeTable set status=#{status} where e\_id=#{e\_id}

根据e\_id修改顺序 int updateSort(e\_id,e\_sort);

update ExchangeTable set e\_sort=#{e\_sort} where e\_id=#{e\_id}

根据状态查询全部 List<ExchangeTable> selectAllStatus(status);

select \* from ExchangeTable where status={#status}

根据状态查询编号、名称 List<ExchangeTable> iselectAllStatus();

select e\_id,e\_name from ExchangeTable where status={#status}

模糊名称查询 List<ExchangeTable> selectAllVague(e\_name);

select \* from ExchangeTable e\_name like concat('%',#{e\_name},'%')

1. 成员信息MemberInformation

根据工作编号修改密码int updatePassword(String m\_user,String m\_password)

update MemberInformation set m\_password=#{m\_password} where m\_user={m\_user}

根据性别、班级、部门、职位、所属届、状态查询所有成员信息 List<MemberInformation> selectListl(m\_sex,c\_id,d\_id,p\_id,e\_id,status);

(要求多表查询，使用mybatis判断字段传入为空时条件不成立)

select m.m\_id,m\_user,m\_name,m\_sex,p\_photo,m\_qq,m.status,r\_admission\_date,m.p\_remarks,c.c\_name,c\_headmaster\_name,c.c\_phone,d.d\_name,p.p\_name,e.e\_name,j.j\_name from MemberInformation m

join ClassTable c on c.c\_id=m.c\_id

join DepartmentType d on d.d\_id=m.d\_id

join PositionType p on p.p\_id=m.p\_id

join ExchangeTable e on e.e\_id=m.e\_id

join Jurisdiction j on j.j\_id=m.j\_id

根据姓名模糊查询、状态 List<MemberInformation> selectListNameStatus(m\_name,status);

select m.m\_id,m\_user,m\_name,m\_sex,p\_photo,m\_qq,m.status,r\_admission\_date,m.p\_remarks,c.c\_name,c\_headmaster\_name,c.c\_phone,d.d\_name,p.p\_name,e.e\_name,j.j\_name from MemberInformation m

join ClassTable c on c.c\_id=m.c\_id

join DepartmentType d on d.d\_id=m.d\_id

join PositionType p on p.p\_id=m.p\_id

join ExchangeTable e on e.e\_id=m.e\_id

join Jurisdiction j on j.j\_id=m.j\_id

where m.status=#{status} and m.m\_name like concat('%',#{m\_name},'%')

根据成员编号修改成员照片 int updatePhoto(m\_id, photo);

update MemberInformation set photo=#{photo} where m\_id=#{m\_id}

根据编号修改状态int updateStatus(m\_id,status);

update MemberInformation set status=#{status} where m\_id=#{m\_id}

1. 工作安排WorkArrangement

根据时间查询

List<WorkArrangement> selectDate(String beforeDate, String afterDate);

select \* from workarrangement where w\_date >= '2018-10-10' and w\_date <= '2019-01-18'

模糊查询工作标题 List<WorkArrangement> selectAllVagueTitle(String w\_title);

select w\_id,w\_title,w\_contents,w\_date,w\_remarks,d.d\_name,m.m\_name,c\_phone from WorkArrangement w

join MemberInformation m on m.m\_id=w.m\_id

join DepartmentType d on d.d\_id=w.d\_id where w\_title like concat('%',#{w\_title},'%')

根据部门查询所有工作 List<WorkArrangement> selectAllVagueName(int d\_id);

select w\_id,w\_title,w\_contents,w\_date,w\_remarks,d.d\_name,m.m\_name,c\_phone from WorkArrangement w

join MemberInformation m on m.m\_id=w.m\_id

join DepartmentType d on d.d\_id=w.d\_id where d.d\_id=#{d\_id}

1. 公告Notic

根据标题模糊查询、状态查询list<Notic> selectTitlestatus(String n\_title,String status);

select n.n\_id,n\_title,n\_context,n\_date,n\_file,n.status,m.m\_name from Notic n

join MemberInformation m on n.m\_id=m.m\_id

where status=#{status} and like n\_title concat('%',#{n\_title},'%'}

根据发布人查询、状态查询 list<Notic> selectMidStatus(int m\_id,status);

select n.n\_id,n\_title,n\_context,n\_date,n\_file,n.status,m.m\_name from Notic n

join MemberInformation m on n.m\_id=m.m\_id

where m.m\_id=#{m\_id} and status=#{status}

查询编号、标题、内容、发布人、发布时间list<Notic> iselectList(String status);

select n.n\_id,n\_title,n\_context,n\_date,m.m\_name from Notic n

join MemberInformation m on n.m\_id=m.m\_id

where status=#{status}

//多个查询

list<Notic> selects(String nTitle, String mName, String status, String beforDate, String afterDate);

select \* from notic where n\_title like concat('%', '关于', '%') and n\_context like concat('%', '放假', '%') and n\_date >= '2018-08-27' and n\_date <= '2019-1-3' and status like concat('%', '', '%')

根据n\_id修改状态 int updateStatus(int n\_id,String status);

update set status=#{status} where n\_id=#{n\_id}

1. 请假记录LeaveRecord

根据部门、请假时间查询请假记录int selectname(String d\_name，String p\_date);

select l\_id,d\_name,ms.m\_user,ms.m\_name,p\_date,l\_title,l\_type,l\_contexts,l\_file,m.m\_name from LeaveRecord l

join MemberInformation m on l.l\_m\_id=m.m\_id

join MemberInformation ms on l.m\_user=ms.m\_user where d\_name=#{d\_name} or p\_date=#{p\_date}

根据成员工作编号查询请假记录 List<LeaveRecord> selectByMid(String m\_user);

select l\_id,d\_name,ms.m\_user,ms.m\_name,p\_date,l\_title,l\_type,l\_contexts,l\_file,m.m\_name from LeaveRecord l

join MemberInformation m on l.l\_m\_id=m.m\_id

join MemberInformation ms on l.m\_user=ms.m\_user where m\_user=#{m\_user}

1. 个人评优记录PersonalMeritRecord

根据个人、时间查询 List<PersonalMeritRecord> selectDidADate(int m\_id,String d\_date);

List<PersonalMeritRecord> selectDidADate(int m\_id,String d\_date);

select p.p\_id,d\_name,mi.m\_name,p\_title,p\_date,p\_contexts,p\_file,m.m\_name from PersonalMeritRecord p

join MemberInformation mi on p.m\_id=mi.m\_id

join MemberInformation m on p.m\_id=m.m\_id where d.m\_id=#{m\_id} or d\_date=#{d\_date}

1. 部门评优记录DepartmentalExcellenceRecord

根据部门、时间查询 List<DepartmentalExcellenceRecord> selectDidADate(int d\_d\_id,String d\_date);

select d.d\_id,d\_name,d\_title,d\_date,d\_contexts,d\_file,m.m\_name from DepartmentalExcellenceRecord d

join DepartmentType dt on d.d\_d\_id=dt.d\_id

join MemberInformation m on d.m\_id=m.m\_id where d.d\_d\_id=#{d\_d\_id} or d\_date=#{d\_date}

1. 财务管理FinancialManagement

添加交易类型、交易金额、时间、操作人、备注、文件路径(FinancialManagement);

//获取上次的时间价钱

select f\_money from financialmanagement order by f\_id desc limit 1

insert into FinancialManagement(f\_type,f\_money,f\_date,m\_id,f\_remarks,f\_file) values

(#{f\_type},#{f\_money},#{f\_date},#{m\_id},#{f\_remarks},#{f\_file})

1. 出勤情况Attendance

根据类型、时间、部门查询 List<Attendance> selectTypedate(String a\_type,String a\_date,String d\_name);

select a\_id,a\_type,u.m\_user,u.m\_name,a\_titile,a\_date,a\_remaks,m.m\_name from Attendance a

join MemberInformation m on a.a\_m\_id=m.m\_id

join MemberInformation u on a.m\_user=u.m\_user

where a\_type=#{a\_type} or a\_date=#{a\_date} or d\_name=#{d\_name}

根据标题模糊查询 List<Attendance> selectTitle(String title);

select a\_id,a\_type,u.m\_user,u.m\_name,a\_titile,a\_date,a\_remaks,m.m\_name from Attendance a

join MemberInformation m on a.a\_m\_id=m.m\_id

join MemberInformation u on a.m\_user=u.m\_user

where title like concat('%',#{title}'%')

根据m\_user查询 List<Attendance> selectTitle(String m\_user);

select a\_id,a\_type,u.m\_user,u.m\_name,a\_titile,a\_date,a\_remaks,m.m\_name from Attendance a

join MemberInformation m on a.a\_m\_id=m.m\_id

join MemberInformation u on a.m\_user=u.m\_user

where a.m\_user=#{m\_user}

1. 会议记录MinutesOfTheMeeting

根据标题模糊查询 List<MinutesOfTheMeeting> selectTitle(String title);

select m.m\_id,m.m\_title,m\_date,m\_file,mi.m\_name,m\_contexts,t.t\_name from MinutesOfTheMeeting m

join MemberInformation mi on m.m\_m\_id=mi.m\_id

join TypeOfMeeting t on m.t\_id=t.t\_id

where m.title=#{title}

根据上传时间查询 List<MinutesOfTheMeeting> selectTitle(String m\_date);

select m.m\_id,m.m\_title,m\_date,m\_file,mi.m\_name,m\_contexts,t.t\_name from MinutesOfTheMeeting m

join MemberInformation mi on m.m\_m\_id=mi.m\_id

join TypeOfMeeting t on m.t\_id=t.t\_id

where m.m\_date=#{m\_date}

1. 月计划总结表MonthlyPlanSummary

根据标题模糊查询 List<MonthlyPlanSummary> selectTitle(String title);

select m.m\_id,m.m\_title,m\_date,m\_file,mi.m\_name,m\_contexts,d.d\_name from MonthlyPlanSummary m

join MemberInformation mi on m.m\_m\_id=mi.m\_id

join DepartmentType d on m.d\_id=d.d\_id

where title=#{title}

根据时间、部门查询 List<MonthlyPlanSummary> selectTypedate(String m\_date,int d\_id);

select m.m\_id,m.m\_title,m\_date,m\_file,mi.m\_name,m\_contexts,d.d\_name from MonthlyPlanSummary m

join MemberInformation mi on m.m\_m\_id=mi.m\_id

join DepartmentType d on m.d\_id=d.d\_id

where m\_date=#{m\_date} or d\_id=#{d\_id}

1. 备份表backups

**报表需求**