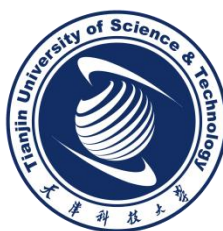


# 《高级数据库应用实验（Oracle）》项目报告

题目：叮咚办公(审批)系统



学院：人工智能学院

专业：计算机科学与技术

分组号		
完成人	学号	姓名
负责人	19101114	董**
成员一		
成员二		

## 相关说明

1. 请项目负责人和成员共同完成本报告撰写；
2. 本报告所涉及资料应由团队共同完成；
3. 本文档打印后签字提交；
4. 请将本资料电子版和系统代码、相关数据等资料打包，以“负责人  
学号\_项目名”形式命名压缩包，课程结束后交给老师。

## 【开发技术】

SQL+Oracle 11g 快捷版+VUE+KOA+JavaScript

## 【项目背景】

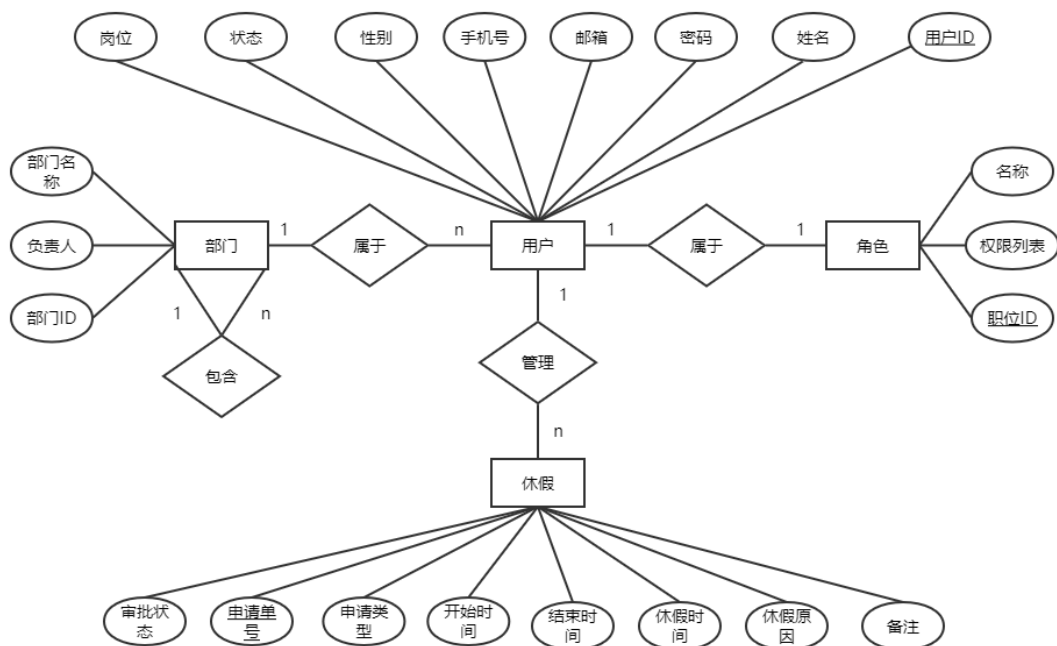
不论是否在疫情期间，线下审批有时候会遇到麻烦，多个盖章和流程，审批人员不在线下，使得请假等各各审批效率低效，因此开通在线办公系统有利于加快审批速度。基于这样的现状，开了一款 oracle+vue+koa 的叮咚办公系统。

## 【实验结果与分析】

### 1. 项目功能分析

- [1] 用户管理：增加和更新员工姓名，邮箱，角色，状态，以及删除员工信息
  - [2] 角色管理：创建和删除角色名称，能够增加和删除权限
  - [3] 部门管理：能够删除和添加部门信息，负责人
  - [4] 休假申请：增加和删除自己假期申请：休假类型，休假时间，休假原因等
  - [5] 审批管理：能够对休假申请进行管理
  - [6] 能够查询请假记录，并且打印单个结果
  - [7] 权限判断：有权限才能管理用户，审批，管理部门和角色，所有人都能够休假申请
- 总而言之，流程就是，拥有权限的用户可以进行用户，部门，角色，休假管理，所有人都有申请休假的权利，当用户申请休假，选择审批人，提交之后，只有选择的审批人能够看到这条审批单。

### 2. 项目 er 图



### 3. 新建表空间和用户

(1) 创建表空间

```
CREATE TABLESPACE officespace
    datafile 'officespace.dbf'
    size 50M
    autoextend on next 5M
    maxsize 100M;
```

(2) 创建临时表空间

```
CREATE TEMPORARY TABLESPACE officetmpspace
    tempfile 'officetmpspace.dbf'
    size 50M
    autoextend on next 5M
    maxsize 100M;
```

(3) 创建用户

```
CREATE USER office IDENTIFIED BY office123
    DEFAULT TABLESPACE officespace
    TEMPORARY TABLESPACE officetmpspace
    quota unlimited on officespace;
```

(4) 创建角色

```
CREATE ROLE master not identified;
```

(5) 赋予角色权限

```
GRANT CONNECT,RESOURCE TO master;
```

(6) 赋予开发者角色

```
GRANT master TO office;
```

4. 创建表并设置完整性约束

(1) 创建数据表

a. 部门表

属性名	类型	说明	约束
bmid	varchar2(10)	部门 id	主键
bmm	varchar2(20)	部门名称	非空
bmr	varchar2(20)	部门负责人	外键
parent	varchar2(10)	父级部门 ID	一级部门默认为 null
created_time	date	创建时间	默认为当前时间
update_time	date	更新时间	

```
CREATE TABLE office_bm (
    bmid varchar2(10) CONSTRAINT pk_bmid PRIMARY KEY,
    bmm varchar2(20) NOT NULL,
```

```

bmr varchar2(20),
parent varchar2(10) DEFAULT NULL,
created_time date DEFAULT SYSDATE,
update_time date DEFAULT SYSDATE
);

```

b. 员工表

属性名	类型	说明	约束
userid	varchar2(10)	用户 id	主键
uname	varchar2(20)	姓名	非空
upwd	varchar2(60)	密码	非空
umail	varchar2(100)	邮箱	
umobile	varchar2(11)	手机号	
usex	varchar2(3)	性别	取值为男或女
ustate	varchar2(6)	状态	取值为在职，离职，实习
deptId	varchar2(10)	部门	外键
job	varchar2(100)	职位	
urole	varchar2(30)	身份	外键
created_time	date	创建时间	默认为当前时间
update_time	date	更新时间	

```

CREATE TABLE office_users (
    userid varchar2(10) CONSTRAINT pk_userid PRIMARY KEY,
    uname varchar2(20) NOT NULL,
    upwd varchar2(60) NOT NULL,
    umail varchar2(100),
    umobile varchar2(11),
    usex varchar2(3) CHECK(usex in ('男','女')),
    ustate varchar2(6) CHECK(ustate in ('在职','离职','实习')),
    deptId varchar2(10),
    job varchar2(100),
    urole varchar2(30),
    created_time date DEFAULT SYSDATE,
    update_time date DEFAULT SYSDATE
);

```

c. 角色表

属性名	类型	说明	约束
roleid	varchar2(10)	身份 id	主键
rolename	varchar2(60)	名称	非空
permission	varchar2(100)	权限	
created_time	date	创建时间	默认为当前时间
update_time	date	更新时间	

```
CREATE TABLE office_role (
    roleid varchar2(10) CONSTRAINT pk_roleid PRIMARY KEY,
    rolename varchar2(60) NOT NULL,
    permission varchar2(100) NOT NULL,
    created_time date DEFAULT SYSDATE,
    update_time date DEFAULT SYSDATE
);
```

d.请假表

属性名	类型	说明	约束
leaveid	varchar2(10)	申请单号	主键
leavetype	varchar2(6)	申请类型	取值为事假，调休，年假
lstart	date	开始时间	非空
lend	date	结束时间	非空
total	number(10)	休假时间	非空
reason	varchar2(1000)	休假原因	
lstate	varchar2(10)	审批状态	取值为待审批，拒绝，同意，作废
remark	varchar2(1000)	备注	
applicant	varchar2(10)	申请人	外键
approver	varchar2(10)	审批人	外键
created_time	date	创建时间	默认为当前时间
update_time	date	更新时间	

```
CREATE TABLE office_leave (
    leaveid varchar2(10) CONSTRAINT pk_leaveid PRIMARY KEY,
    leavetype varchar2(6) CHECK(leavetype in ('事假','调休','年假')),
    lstart date NOT NULL,
    lend date NOT NULL,
    total number(10) NOT NULL,
    reason varchar2(1000),
    lstate varchar2(10) CHECK(lstate in ('待审批','拒绝','同意','作废')),
    remark varchar2(1000),
    applicant varchar2(10),
    approver varchar2(10),
    created_time date DEFAULT SYSDATE,
    update_time date DEFAULT SYSDATE
);
```

(2) 设置完整性约束：

说明：在前面的 create 语句中已经设置了 CHECK 约束、NOT NULL 约束、主键约束，这里只额外设置外键约束

a. 部门表:

```
alter table office_bm add constraint fk_bmr foreign key(bmr) references office_users(userid) on delete set null;
```

b. 员工表:

```
alter table office_users add constraint fk_deptId foreign key(deptId) references office_bm(bmid) on delete set null;
```

```
alter table office_users add constraint fk_roleid foreign key(urole) references office_role(roleid) on delete set null;
```

c. 请假表:

```
alter table office_leave add constraint fk_applicant foreign key(applicant) references office_users(userid) on delete set null;
```

```
alter table office_leave add constraint fk_approver foreign key(approver) references office_users(userid) on delete set null;
```

## 5. 序列

*用于 id 的自增。*

(1) 用户序列

```
CREATE SEQUENCE user_seq  
INCREMENT BY 1  
START WITH 10000  
NOMAXVALUE  
NOCYCLE  
CACHE 10;
```

(2) 部门序列

```
CREATE SEQUENCE bm_seq  
INCREMENT BY 1  
START WITH 10000  
NOMAXVALUE  
NOCYCLE  
CACHE 10;
```

(3) 角色序列

```
CREATE SEQUENCE role_seq  
INCREMENT BY 1  
START WITH 10000  
NOMAXVALUE  
NOCYCLE  
CACHE 10;
```

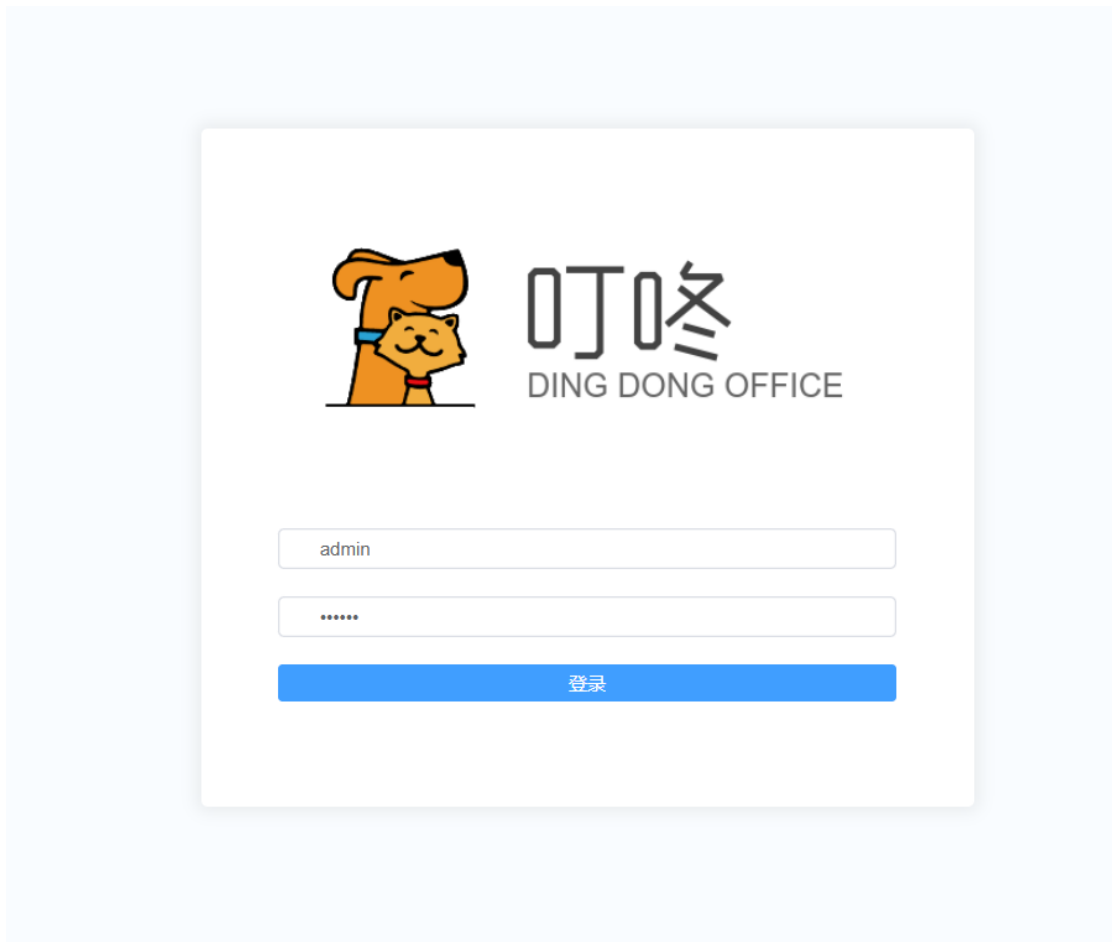
## 6. SQL 语句

(1) 系统登录功能的实现

*这里先展示简单的 select 语句，后面用存储过程升级了一下。*

```
select * from office_users where  
uname = v_username and upwd = v_password;
```

登陆界面如图所示：



登陆后，跳转到首页：



## (2) 数据添加功能的实现

### 1.1 用户添加


实现代码：

```
INSERT INTO office_users(userid,uname,upwd,umail,umobile,usex,ustate,deptid,job,urole)
```



```
VALUES (user_seq.nextval,:v_username, :v_password, :v_mail,:v_mobile, :v_usex,  
       :v_ustate, :v_did, :v_job, :v_role)
```

界面所图所示:




The image shows a '用户新增' (Add User) modal form. It contains the following fields: '用户名' (Username) with a text input, '邮箱' (Email) with a text input and a domain dropdown set to '@edu.tust.cn', '手机号' (Mobile Number) with a text input, '岗位' (Position) with a text input, '状态' (Status) with a dropdown menu, '角色' (Role) with a dropdown menu, and '部门' (Department) with a dropdown menu. At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

## 1.2 部门添加

实现代码:

```
INSERT INTO office_bm(bmid,bmm,bmr,parent) VALUES  
(bm_seq.nextval, :bmm, :bmr, :parent)
```

界面:




The image shows a '创建部门' (Create Department) modal form. It contains the following fields: '上级部门' (Superior Department) with a dropdown menu, '部门名称' (Department Name) with a text input, and '负责人' (Responsible Person) with a dropdown menu. At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

## 1.3 角色添加

实现代码:

```
INSERT INTO office_role(roleid,rolename) VALUES (role_seq.nextval, :name)
```

界面:

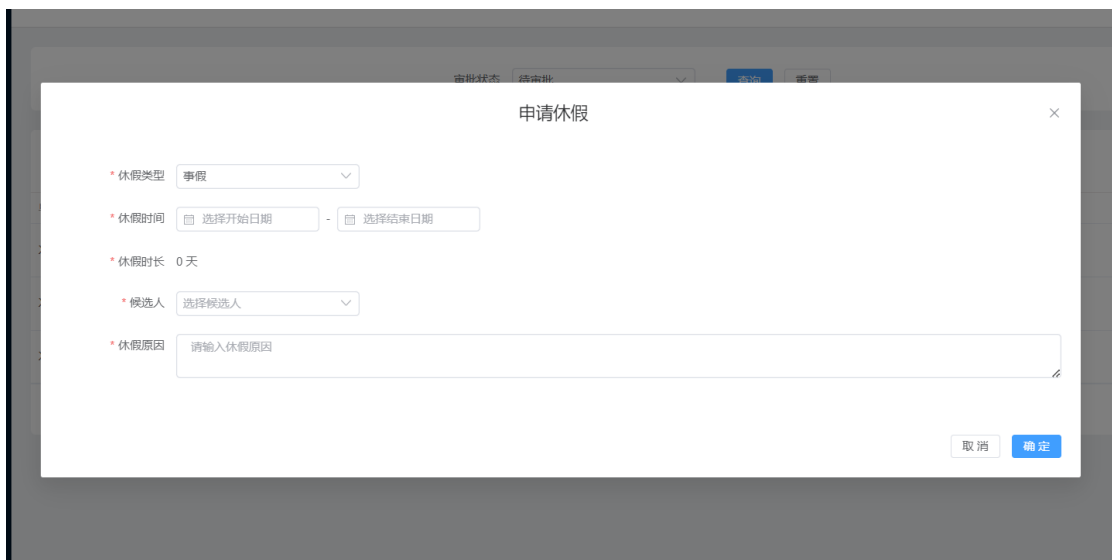


#### 1.4 休假申请

实现代码:

```
INSERT INTO office_leave(leaveid,leavetype,lstart,lend,reason,lstate,applicant,approver)
VALUES (:0,:1,to_date(:2, 'yyyy/MM/dd'),to_date(:3, 'yyyy/MM/dd'),:4,:5,:6,:7)
```

界面:



### (3) 数据修改功能的实现

#### 1.1 用户编辑

实现代码:

```
update office_users set umobile=:umobile,ustate=:ustate,deptid=:deptid,job=:job,urole=:urole
WHERE userid=:id
```

界面:

用户ID

请输入用户ID

用户名称

请输入用户名称

状态

在职

用户编辑

\* 用户名

dyedd

\* 邮箱

dyedd

@edu.tust.cn

手机号

110

岗位

前端工程师

状态

在职

角色

管理员

部门

请选择所属部门

取消

确定

## 1.2 用户离职

实现代码:

```
update office_users set ustate=:ustate WHERE userid in (${condition.join(',')})
```

说明: 用户模块的删除就是实现用户的离职, 本质还是个 *update* 语句

界面:

离职成功的

<input type="checkbox"/>	ID	用户名	邮箱	角色	岗位	状态	注册时间	更新时间	操作
<input type="checkbox"/>	10015	test002	test002@edu.tust.cn			离职	2022-05-30 09:45:04	2022-06-02 22:06:47	编辑 删除
<input type="checkbox"/>	10014	test001	test001@edu.tust.cn			离职	2022-05-30 09:42:16	2022-05-30 09:42:16	编辑 删除
<input type="checkbox"/>	10004	simoin	null@tust.edu.cn		10001	离职	2022-05-17 11:00:09	2022-05-17 11:00:09	编辑 删除
<input type="checkbox"/>	10016	test003	test003@tust.edu.cn			离职	2022-05-30 14:08:52	2022-05-30 14:08:52	编辑 删除

<

1

>

## 1.3 部门编辑

实现代码:

```
update office_bm set bmm=:bmm,bmr=:bmr,parent=:parent WHERE bmid=:id
```

界面:

部门名称 请输入部门名称 查询 重置

### 编辑部门

上级部门 请选择上级部门

\* 部门名称 安软慧视

\* 负责人 请选择部门负责人

取消 确定

#### 1.4 角色编辑

实现代码:

```
update office_role set rolename=:0 WHERE roleid =:1
```

界面:

角色名称 请输入角色名称 查询 重置

### 角色编辑

\* 角色名称 管理员

取消 确定

#### 1.5 权限编辑

实现代码:

```
update office_role set permission=:0 WHERE roleid =:1
```

界面:

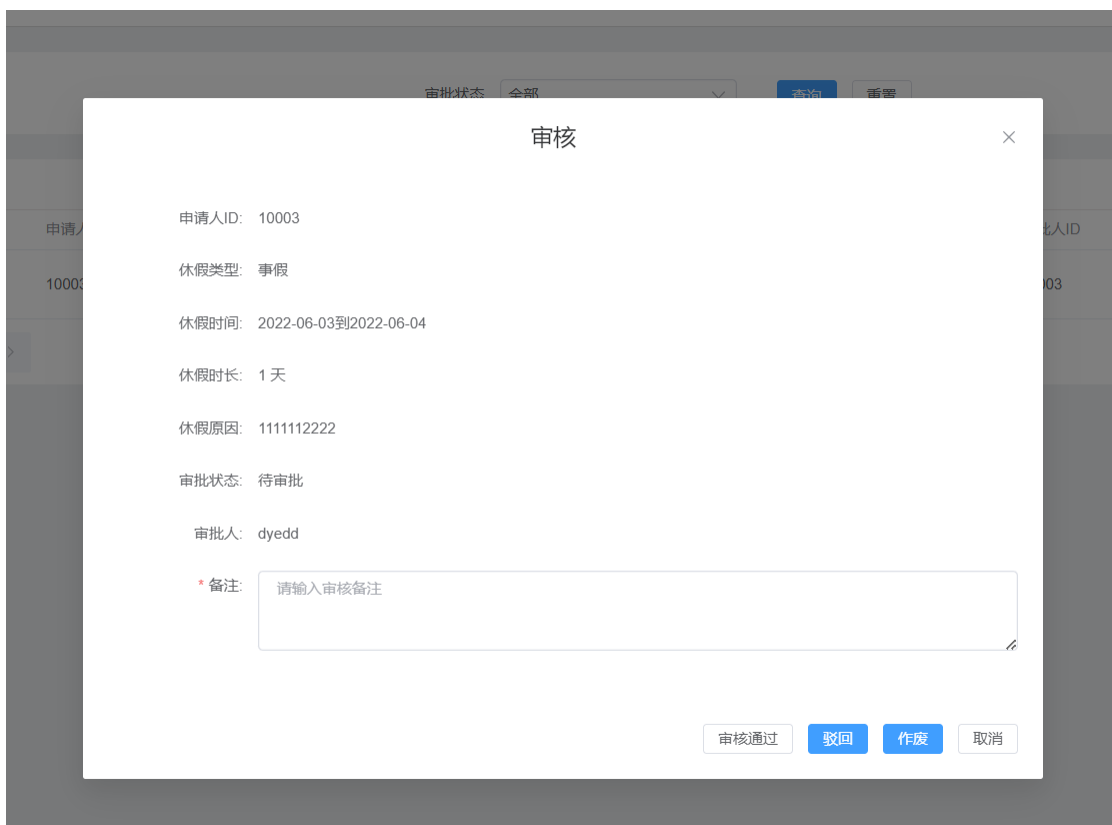


## 1.6 审批管理

实现代码:

```
update office_leave set lstate=:0,remark=:1,approver=:2 WHERE leaveid=:3
```

界面:



## (4) 数据删除功能的实现

说明: 删除都是一键不存在结果的, 因此接下来的删除功能没有配图

### 1.1 部门删除

实现代码:

```
DELETE FROM office_bm WHERE BMID=:id
```

### 1.2 角色删除

实现代码:

```
DELETE FROM office_role WHERE roleid = :id
```

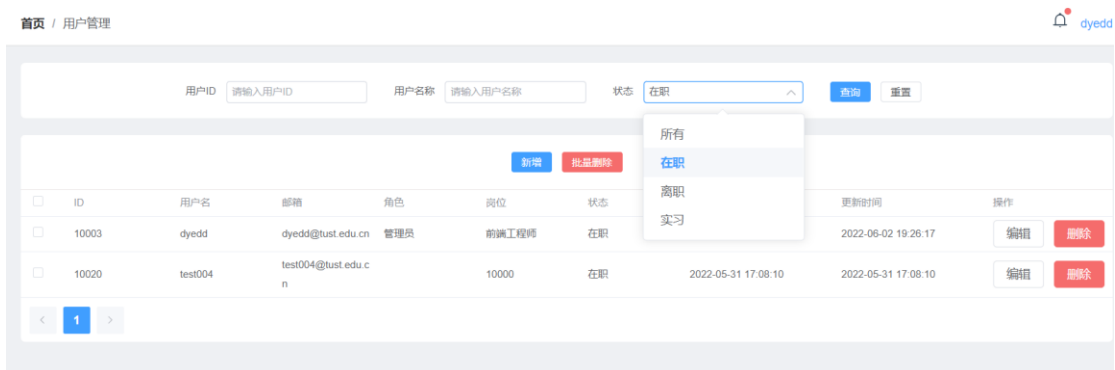
(5) 数据查询功能的实现

### 1.1 全部用户查询

实现代码:

```
SELECT userid,uname,umail,nvl(urole,'员工 '),nvl(job,'新入职'),ustate,created_time,update_time FROM USER_VIEW
```

界面:



### 1.2 用户分页查询

实现代码:

```
SELECT * FROM USER_VIEW WHERE ${params} rownum >= ${skipIndex} AND rownum <= ${skipIndex + page.pageSize - 1}
```

说明: 通过 *rownum* 进行了分页查询, 每一次通过给的页面大小和页数查询出用户

### 1.3 部门查询

实现代码:

```
if (deptname) {
    result = await ctx.db.execute(
        `SELECT * FROM office_bm o left join BM_NUM_VIEW b on o.bmid =
b.deptid where o.bmm = :name`,
        [deptname]);
} else {
    result = await ctx.db.execute(
        `SELECT * FROM office_bm o left join BM_NUM_VIEW b on o.bmid =
b.deptid`
    );
}
```

说明: 通过 *deptname* 来查询是否查询单个部门还是所有部门, 部门列表实现了树状结构

所有部门界面:

部门名称 <input type="text" value="请输入部门名称"/>						查询	重置
						创建	
部门名称	负责人	部门人数	更新时间	创建时间	操作		
√ 安软基础		3	2022-06-02T14:06:47.000Z	2022-05-17T02:42:57.000Z	编辑	删除	
前调部门		1	2022-05-17T03:01:57.000Z	2022-05-17T03:01:57.000Z	编辑	删除	
淘宝		1	2022-05-17T02:43:41.000Z	2022-05-17T02:43:41.000Z	编辑	删除	
编程教育		0	2022-05-17T02:43:59.000Z	2022-05-17T02:43:59.000Z	编辑	删除	
QQ		0	2022-05-30T09:32:35.000Z	2022-05-30T09:32:35.000Z	编辑	删除	

查询某个部门界面:

部门名称

查询

重置

#### 1.4 角色查询

实现代码:

```
SELECT * FROM office_role WHERE ${params} rownum >= ${skipIndex} AND rownum
<= ${skipIndex + page.pageSize - 1}
```

界面:

角色名称 <input type="text" value="请输入角色名称"/>				查询	重置
				创建	
角色名称	更新时间	创建时间	操作		
管理员	2022-06-01 14:44:04	2022-06-01 14:44:04	编辑	设置权限	删除
测试2	2022-06-01 14:46:27	2022-06-01 14:46:27	编辑	设置权限	删除

<

1

>

#### 1.5 我的休假列表

实现代码:

```
if(action == 'create'){
    params.length > 0 ? params += 'AND applicant =:applicant ' : params = 'applicant
=:applicant ';
} else if(action == 'accept'){
    params.length > 0 ? params += 'AND approver =:approver ' : params = 'approver
=:approver ';
} else {
    ctx.body = util.fail('请传入 action 参数');
    return;
}
values.push(userid);
SELECT * FROM office_leave WHERE  ${params}
    rownum >= ${skipIndex} AND rownum <= ${skipIndex + page.pageSize - 1}
```

界面:

首页 / 休假申请

审批状态: 待审批

查询 重置

申请休假

单号	休假时间	休假时长	休假类型	休假原因	申请时间	审批人	审批状态	操作
X202206033	2022-06-03到2022-06-04	1	事假	1111112222	2022-06-03 20:12:14	10003	待审批	查看
X202206020	2022-06-02到2022-06-03	1	事假	1111	2022-06-02 19:47:57		待审批	查看
X202206022	2022-06-04到2022-06-16	12	事假	11111	2022-06-02 20:43:35		待审批	查看
X202206021	2022-06-03到2022-06-10	7	事假	2222222	2022-06-02 20:29:45		待审批	查看

< 1 >



## 1.6 审批列表

实现代码:

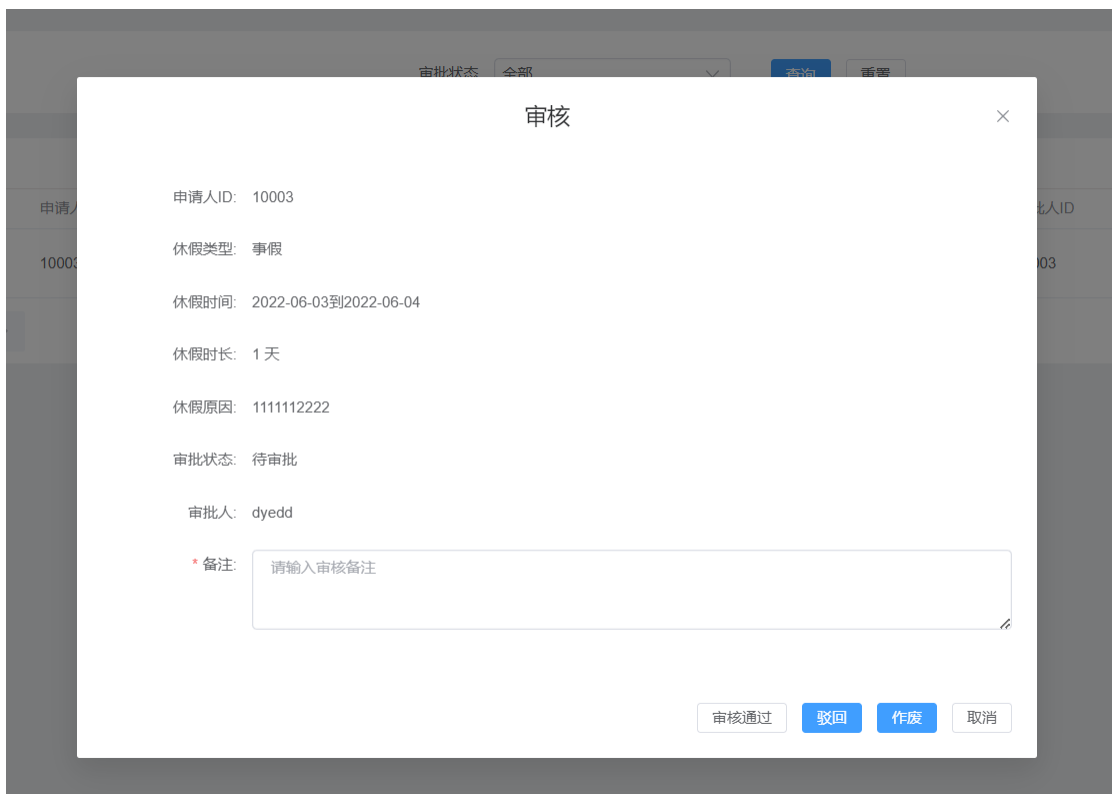
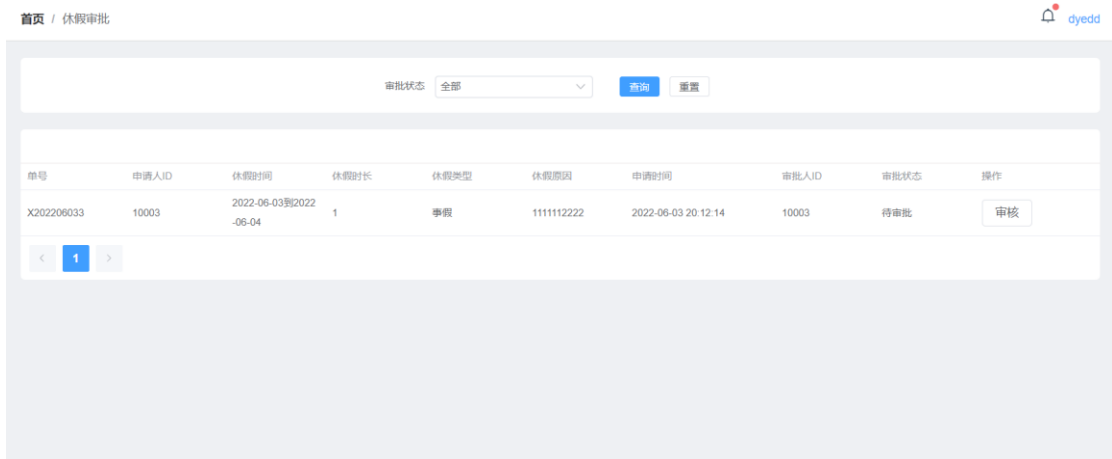
```
if(action == 'create'){
    params.length > 0 ? params += 'AND applicant =:applicant ' : params = 'applicant
=:applicant ';
} else if(action == 'accept'){
    params.length > 0 ? params += 'AND approver =:approver ' : params = 'approver
=:approver ';
} else{
    ctx.body = util.fail('请传入 action 参数');
    return;
}
```



```
values.push(userid);
SELECT * FROM office_leave WHERE  ${params}
rownum >= ${skipIndex} AND rownum <= ${skipIndex + page.pageSize - 1} o
```

说明：按照天数排列

界面：



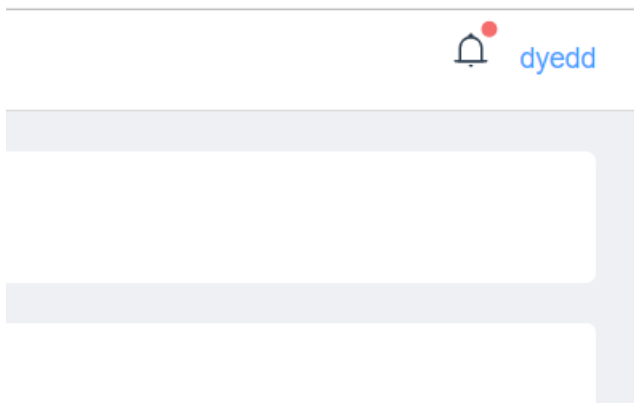
说明：只能显示审批人是该用户的

1.7 查询通知的个数，即需要审批多少个

实现代码：

```
SELECT count(*) FROM office_leave WHERE approver =:approver and lstate='待审批'
```

界面：



## 1.8 显示部门人数

实现代码:

```
SELECT deptid, count(*) as bmnum FROM office_users group by deptid
```

界面:

部门名称	负责人	部门人数	更新时间	创建时间	操作
> 安欣慧视		3	2022-06-02T14:06:47.000Z	2022-05-17T02:42:57.000Z	<a href="#">编辑</a> <a href="#">删除</a>
淘宝		1	2022-05-17T02:43:41.000Z	2022-05-17T02:43:41.000Z	<a href="#">编辑</a> <a href="#">删除</a>
编程教育		0	2022-05-17T02:43:59.000Z	2022-05-17T02:43:59.000Z	<a href="#">编辑</a> <a href="#">删除</a>
QQ		0	2022-05-30T09:32:35.000Z	2022-05-30T09:32:35.000Z	<a href="#">编辑</a> <a href="#">删除</a>

## 1.9 显示审批人列表

实现代码:

```
select * from office_users u left join office_role r on u.urole = r.roleid
where r.permission like '%leave%'
OR u.userid = (select bmr from office_bm where bmrid = :0)
```

界面:

\* 休假时长 0天

\* 候选人

\* 休假原因

说明: 拥有 leave 权限以及部门主管才有资格成为候选人

## 7. 视图和存储过程

### 1.1 用户视图

说明: 出于安全性考虑, 一般返回给用户的信息不包含密码

实现代码:

```
CREATE OR REPLACE VIEW USER_VIEW AS
SELECT userid,uname,umail,umobile,usex,ustate, deptid, job,urole,
created_time,update_time FROM OFFICE_USERS
WITH CHECK OPTION;
```

## 1.2 部门视图

*说明：将部门人数和部门id 组成一个视图，便于连接*

```
CREATE OR REPLACE VIEW BM_NUM_VIEW AS
SELECT deptid, count(*) as bmnum FROM office_users group by deptid
WITH CHECK OPTION;
```

## 1.3 存储过程，登陆

*实现代码：*

```
create or replace procedure login(
v_username in office_users.uname%type,
v_password in office_users.upwd%type,
v_info out SYS_REFCURSOR
)
as
begin
    OPEN v_info FOR
    select * from office_users where
    uname = v_username and upwd = v_password;
end;
```

*说明：最后返回一个游标，供后端获取信息*

## 1.4 存储过程，插入用户

*实现代码：*

```
create or replace procedure register(
v_username in office_users.uname%type,
v_password in office_users.upwd%type,
v_mail in office_users.umail%type,
v_mobile in office_users.umobile%type,
v_usex in office_users.usex%type,
v_ustate in office_users.ustate%type,
v_did in office_users.deptid%type,
v_job in office_users.job%type,
v_role in office_users.urole%type
)
as
begin
    INSERT INTO office_users(userid,uname,upwd,umail,umobile,usex,ustate,deptid,job,urole)
VALUES

(user_seq.nextval,v_username,v_password,v_mail,v_mobile,v_usex,v_ustate,v_did,v_job,v_r
ole);
end;
```

## 1.5 存储过程，部门主管和拥有权限的人才能审批

*实现代码：*

```
create or replace procedure approve_leave(
```

```

lstate in office_leave.lstate%type,
remark in office_leave.remark%type,
approver in office_leave.approver%type,
v_leaveid in office_leave.leaveid%type
)
as
BEGIN
declare
userid office_users.userid%type;
bmr office_bm.bmr%type;
BEGIN
    select count(office_users.userid) into userid from office_users left join office_role on
office_users.urole = office_role.roleid
        where office_role.permission like '%leave%' and office_users.userid = approver;
    if not userid = 0
        then
            goto do_execute;
        end if;
    select office_bm.bmr into bmr from office_bm where office_bm.bmid = (
        select deptid from office_users where office_users.userid =
            (select applicant from office_leave where office_leave.leaveid = v_leaveid)
    );
    if bmr = approver
        then
            goto do_execute;
        end if;
    raise_application_error(-20998, 'no sufficient permissions');
    <<do_execute>>
    update office_leave set office_leave.lstate = lstate, office_leave.remark = remark,
        office_leave.approver = approver where office_leave.leaveid = v_leaveid;
end;
END;

```

## 8. 触发器

### 1.1 实现部门更新资料，自动更新时间

实现代码：

```

CREATE OR REPLACE Trigger bm_updated
Before Update On office_bm
For Each Row
Declare
Begin
    :new.update_time := Sysdate;
End;

```

### 1.2 晚上 11 点到早上 7 点不允许对部门进行操作

实现代码：

```

create or replace trigger prohibit_bm
before insert or delete or update on office_bm
for each row
begin
    if(to_number(to_char(sysdate,'HH24')) >= 23 or to_number(to_char(sysdate,'HH24')) <= 7)
then
    raise_application_error(-20000,'每天晚上 11 点到早上 7 点不允许对部门进行操作');
    end if;
end;

```

1.3 实现审批修改信息，自动更新时间

*实现代码：*

```

CREATE OR REPLACE Trigger leave_updated
Before Update On office_leave
For Each Row
Declare
Begin
    :new.update_time := Sysdate;

```

End;

1.4 实现新建休假事情时候，总计天数自动计算

*实现代码：*

```

CREATE OR REPLACE TRIGGER leave_time_total
BEFORE INSERT ON office_leave
FOR EACH ROW
BEGIN
    if inserting then
        :new.total := to_number(:new.lend - :new.lstart);
    end if;
END;

```

1.5 实现角色修改信息，自动更新时间

*实现代码：*

```

CREATE OR REPLACE Trigger role_updated
Before Update On office_role
For Each Row
Declare
Begin
    :new.update_time := Sysdate;

```

End;

1.6 实现用户修改信息，自动更新时间

*实现代码：*

```

CREATE OR REPLACE Trigger user_updated
Before Update On office_users
For Each Row
Declare
Begin

```

```

        :new.update_time := Sysdate;
End;
1.7 实现用户离职，部门主管自动为空
实现代码：
create or replace trigger user2bm
after update on office_users
for each row
begin
    if updating then
        if :new.ustate = '离职' then
            update office_bm set bmr = " where bmr = :old.userid;
        end if;
    end if;
end;

```

1.8 增加用户时，若用户输入手机号小于 11 位情况下不能注册

实现代码：

```

create or replace trigger check_umobile
before insert or update on office_users
for each row
begin
    if :new.umobile is not null
    then
        if not regexp_like(:new.umobile, '^d{11}$')
        then
            raise_application_error(-2000, 'Invalid mobile number');
        end if;
    end if;
end;

```

1.9 部门人数不能超过 30 人

实现代码：

```

create or replace trigger check_bm_num
before insert on office_bm
for each row
declare
    people number;
begin
    select count(*) into people where bmid = :new.bmid;
    if people >= 30 then
        raise_application_error(-20002, '部门人数最多 30 人');
    end if;
end;

```

其它整体界面：

管理面板

系统管理

用户管理

部门管理

角色管理

休假申请

审批管理

首页 / 用户管理

用户ID: 请输入用户ID

用户名称: 请输入用户名称

状态: 在职

查询

重置

新增

批量删除

<input type="checkbox"/>	ID	用户名	邮箱	角色	岗位	状态	注册时间	更新时间	操作
<input type="checkbox"/>	10003	dyedd	dyedd@tust.edu.cn	管理员	前端工程师	在职	2022-05-17 10:59:45	2022-06-02 19:26:17	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	10020	test004	test004@tust.edu.cn		10000	在职	2022-05-31 17:08:10	2022-05-31 17:08:10	<div>编辑</div> <div>删除</div>

<

1

>

管理面板

系统管理

用户管理

部门管理

角色管理

休假申请

审批管理

首页 / 部门管理

部门名称: 请输入部门名称

查询

重置

创建

部门名称	负责人	部门人数	更新时间	创建时间	操作
> 安软慧捷		3	2022-06-02T14:06:47.000Z	2022-05-17T02:42:57.000Z	<div>编辑</div> <div>删除</div>
淘宝		1	2022-05-17T02:43:41.000Z	2022-05-17T02:43:41.000Z	<div>编辑</div> <div>删除</div>
编程教育		0	2022-05-17T02:43:59.000Z	2022-05-17T02:43:59.000Z	<div>编辑</div> <div>删除</div>
QQ		0	2022-05-30T09:32:35.000Z	2022-05-30T09:32:35.000Z	<div>编辑</div> <div>删除</div>

管理面板

系统管理

用户管理

部门管理

角色管理

休假申请

审批管理

首页 / 角色管理

角色名称: 请输入角色名称

查询

重置

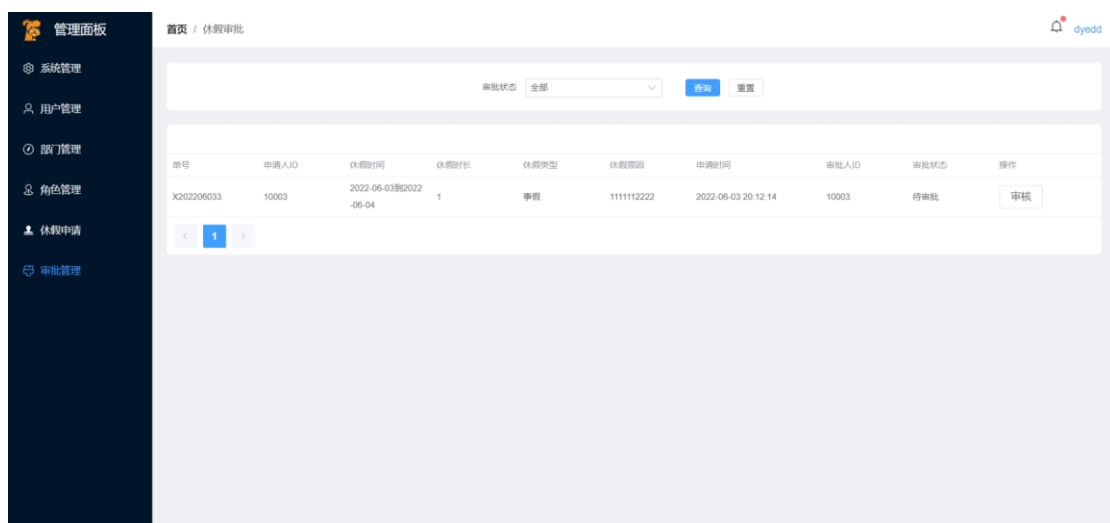
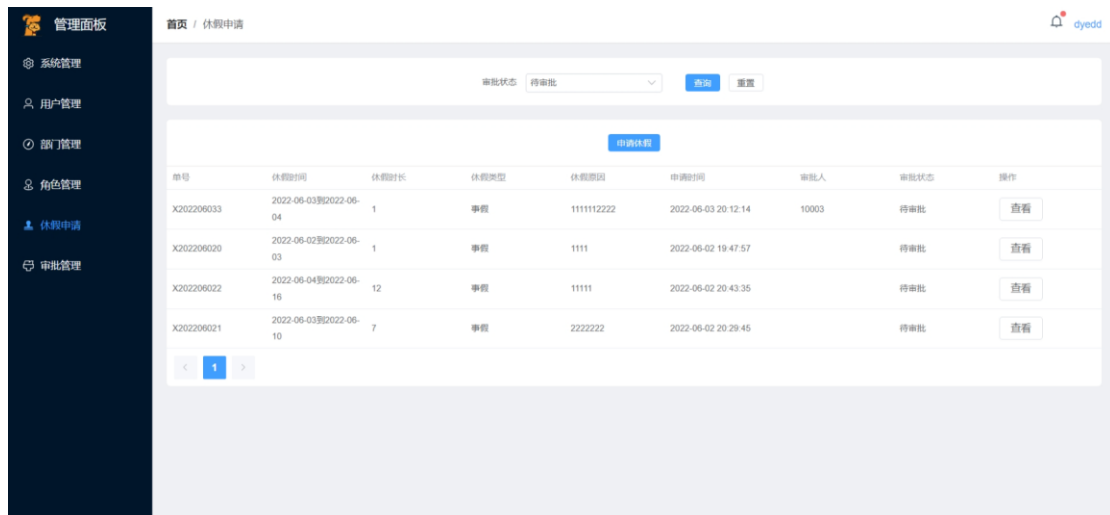
创建

角色名称	更新时间	创建时间	操作
管理员	2022-06-01 14:44:04	2022-06-01 14:44:04	<div>编辑</div> <div>设置权限</div> <div>删除</div>
测试2	2022-06-01 14:46:27	2022-06-01 14:46:27	<div>编辑</div> <div>设置权限</div> <div>删除</div>

<

1

>



## 【实验思考及总结】

本次课程设计我选择审批系统，因为我觉得这个十分符号当下疫情下的主题，这也将是我对 oracle 融合项目的第一次使用。我也喜欢挑战，后期使用的后端语言也是 JavaScript，试试 koa 对 oracle 的融合。

我也做过了很多数据库的建立，在 oracle 建表上，在迁移上有许多相似的影子，也因此对 oracle 专有的写法和类型有了进一步的记忆。

当然，对于本次实验，一开始头疼在于 ER 图，在头疼怎么表示联系好，经过查阅书籍和网上例子，得出一些结论。因此，我对他们都有所深入了解。

在开发过程中，我发现我以前往往在设计时候没有把很多功能交给数据库，因此对于触发器，存储过程是没有用过的，把所有功能交给了前端或者后端开发。本次课程设计让我对触发器，存储过程更加熟悉了。