**数据库大作业文档**

1. **系统定义**

1、实现意图

为了方便学生和老师管理和查询成绩，分析成绩数据，建立了一个成绩管理系统。

2、实现目标

该学生成绩管理系统，实现简单的密码修改、成绩查询、课程查询、学生查询等操作，并能实现简单的数据统计。

1. **需求分析**

1、需求分析

（一）用户登录

在登录程序之前会先弹出一个登录对话框，在正确的输入了数据库中存储

的用户和密码后才能登录程序。

（二）用户密码的修改

在成功登录了程序之后用户可以根据自己的需要修改当前密码。

（三）成绩查询

1.成绩修改前选中要修改的成绩信息，然后进行修改。

2.成绩添加直接弹出添加对话框，然后进行添加。

3.成绩删除前选中要删除的成绩信息，然后直接删除。

（四) 课程查询

1.课程修改实现对课程名、学时、学分的修改。

2.课程添加对一门新开设的课程进行录入，并存入数据库。

3.课程删除对一门不再开设的课程进行删除，并从数据库中删除。

（五）学生查询

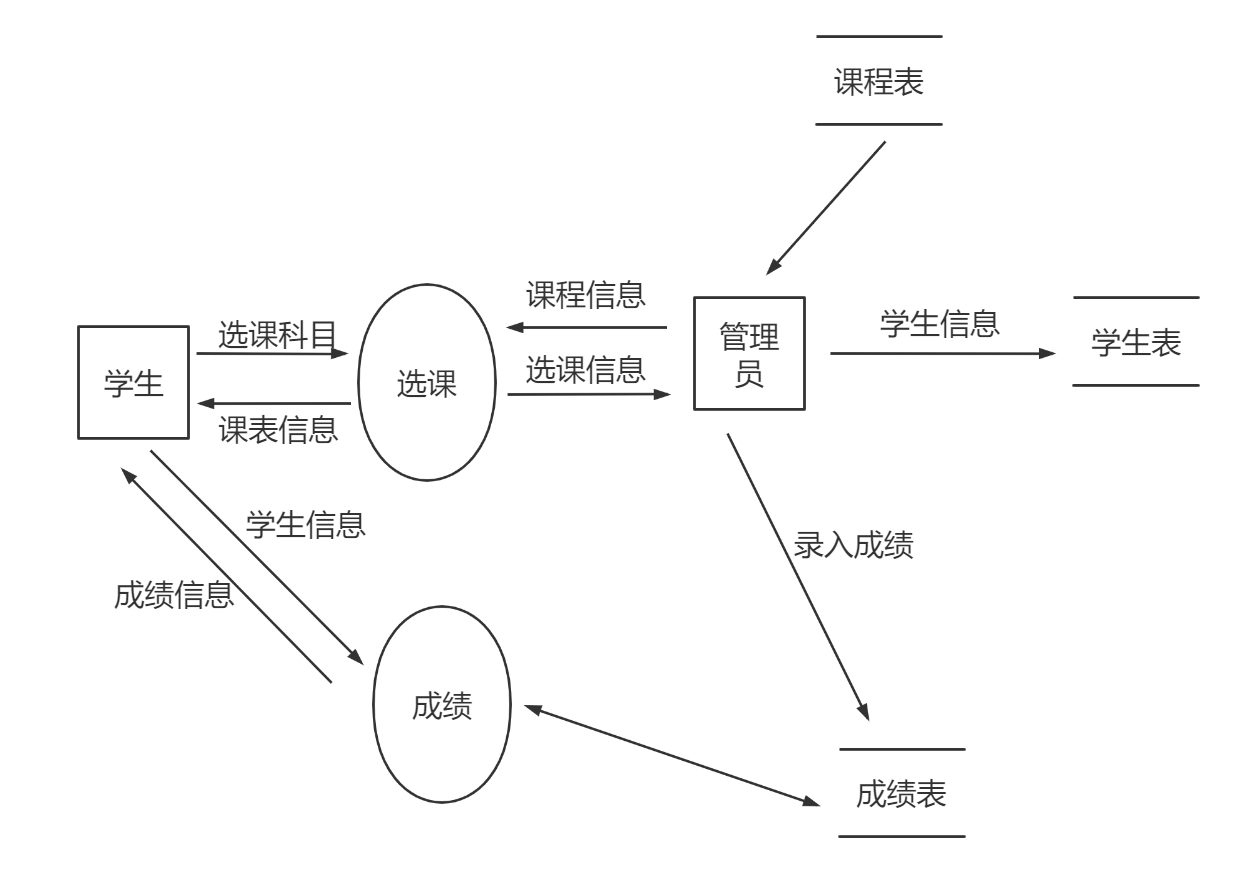
1 .学生信息修改修改选中当前学生的信息。

2 .学生信息添加增加新学生的信息。

3.学生信息删除删除选中当前学生信息。

2、

（1）数据流图



（2）数据字典

数据库中用到的表：

|  |  |  |
| --- | --- | --- |
| 数据库表名 | 关系模式名称 | 备注 |
| Student | 学生 | 学生信息表 |
| Course | 课程 | 课程信息表 |
| Score | 成绩 | 成绩表 |
| Teacher | 老师 | 老师信息表 |

Student基本情况数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| Student \_id | Varchar(10) | 主键 | 学号 |
| Student \_name | Varchar(20) | 非空 | 学生姓名 |
| sex | Char(1) | ‘男’或‘女’ | 性别 |
| College\_name | Varchar(20) | 非空 | 学院名称 |
| major | Varchar(20) | 非空 | 专业 |
| Password | Varchar(20) | 非空 | 密码 |

course数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| course\_id | Varchar(10) | 主键 | 课程号 |
| course\_cname | Varchar(10) | 非空 | 课程名称 |
| course\_hour | int | 非空 | 课时 |
| course\_score | Int | 非空 | 学分 |
| Teacher\_id | Varchar(10) | 非空 | 老师编号 |

score情况数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| course\_id | Varchar(10) | 外键 | 课程号 |
| student\_id | Varchar(10) | 外键 | 学号 |
| score | int |  | 成绩 |

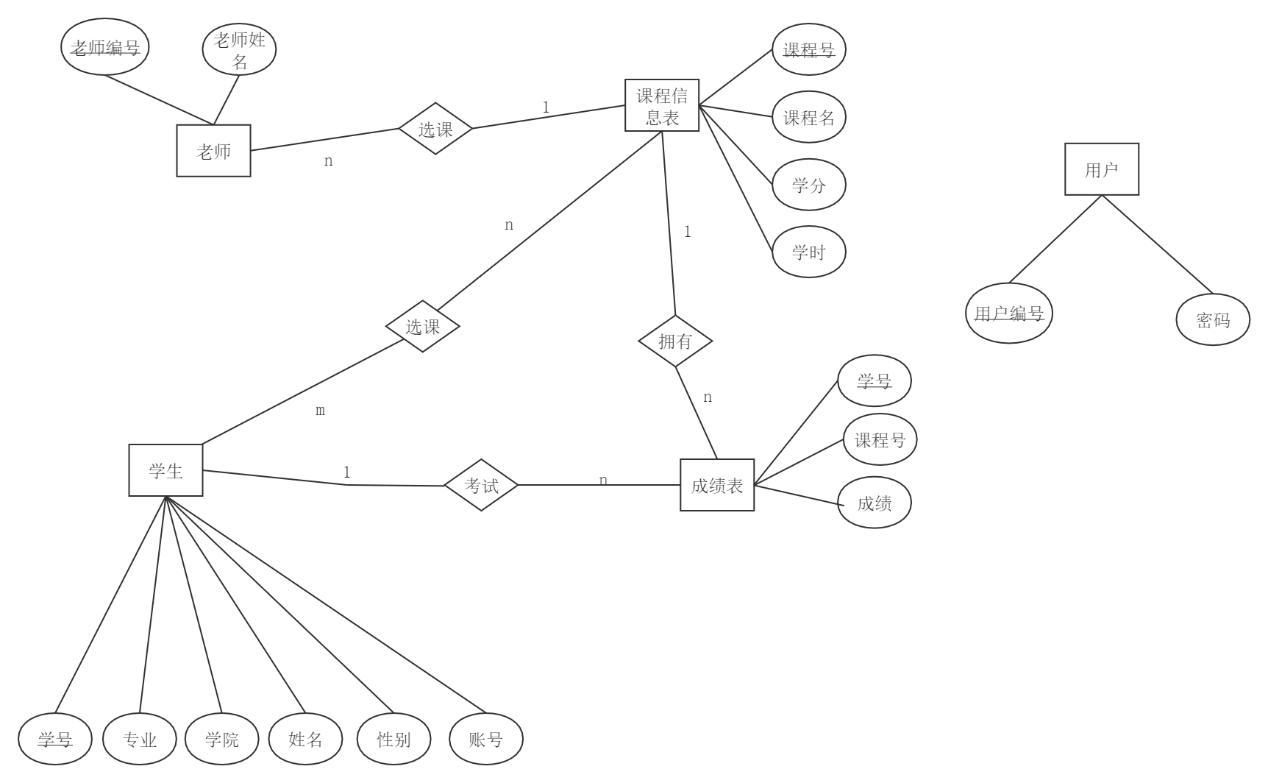
teacher情况数据表，结构如下：

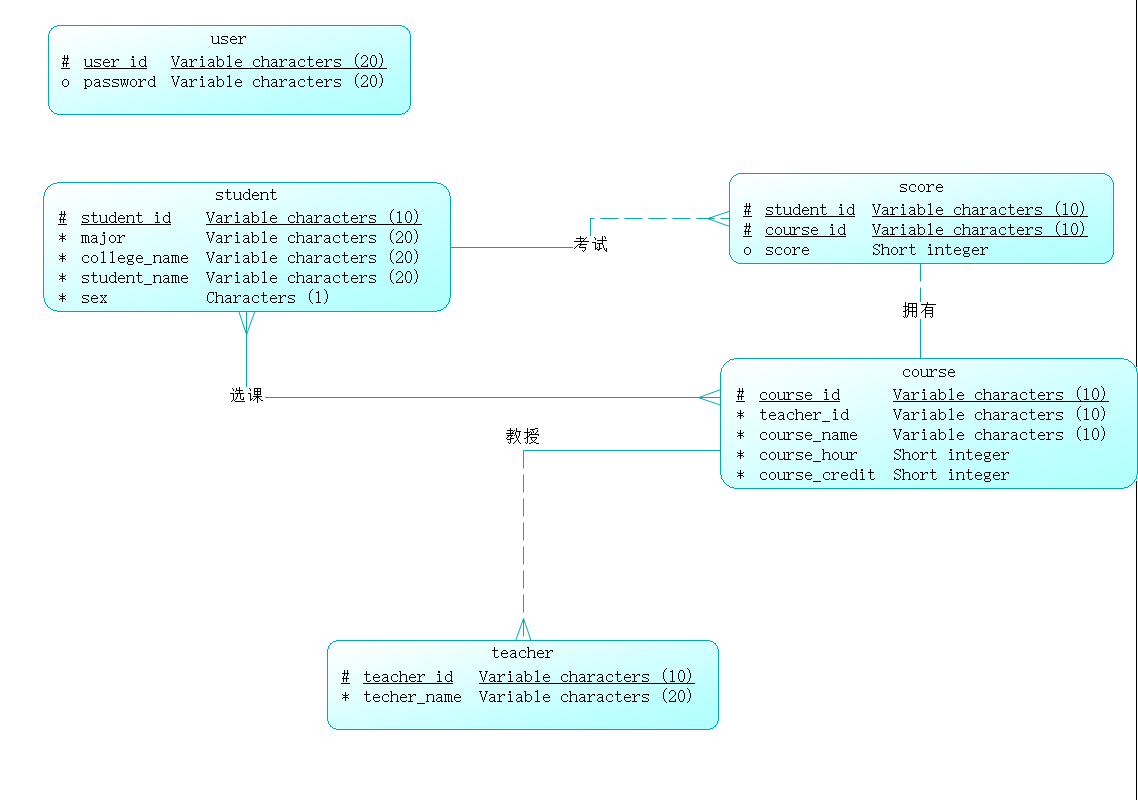
|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| teacher\_id | Varchar(10) | 主键 | 老师编号 |
| teacher\_name | Varchar(20) | 非空 | 老师姓名 |
| Password | Varchar(20) | 非空 | 密码 |

**三、系统设计**

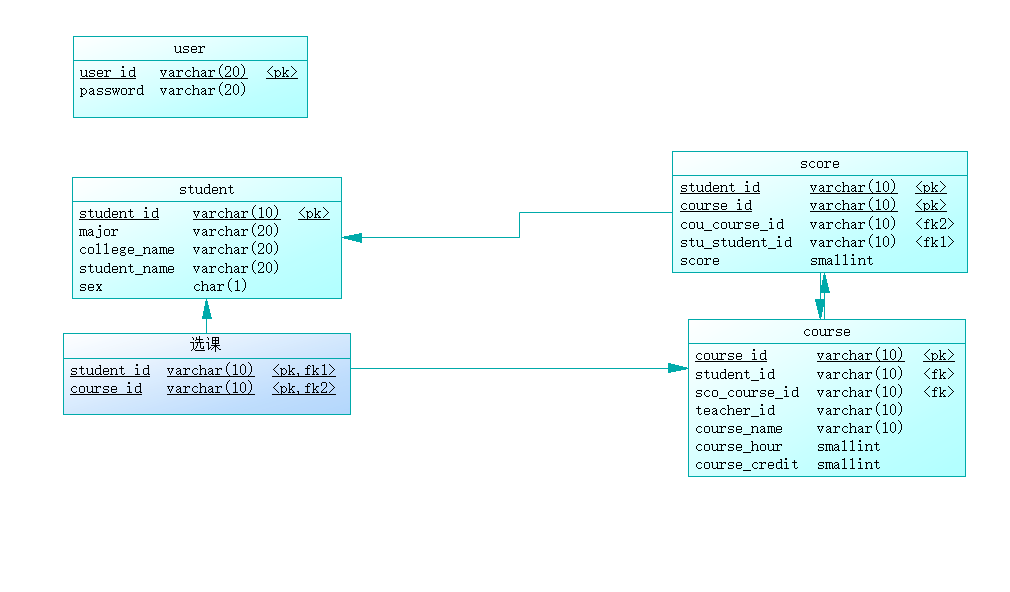
**1、概念结构设计**

系统E-R图

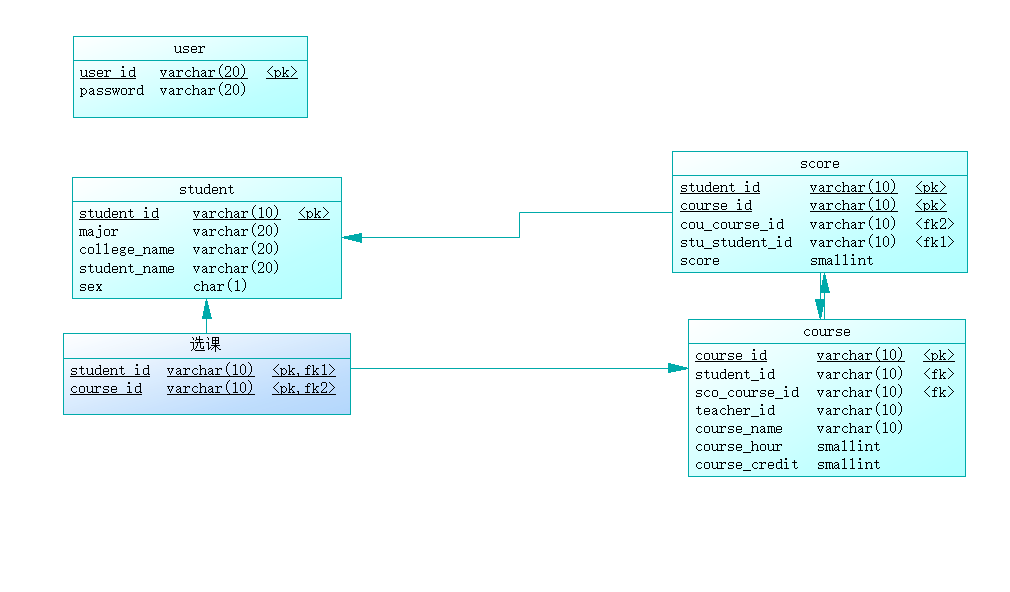


逻辑设计图(CDM)

**2、逻辑结构设计**



**3、系统功能模块图**



**4、其它设计图形工具**

**Xmind、processon**

**四、详细设计**

#初始化

-- 清除已建的表和函数

drop table if exists score;

drop table if exists course;

drop table if exists student;

drop table if exists teacher;

drop table if exists user;

drop function if exists getCredit;

-- 建表

#创建学生信息表

create table student (

student\_id varchar(10) primary key not null,

student\_name varchar(20) not null,

student\_sex char(1) not null check (student\_sex in('男','女')),

student\_college varchar(20) not null,

student\_major varchar(20) not null

);

#创建老师信息表

create table teacher(

teacher\_id varchar(10) primary key not null,

teacher\_name varchar(20) not null

);

#创建课程信息表

create table course(

course\_id varchar(10) primary key not null,

teacher\_id varchar(10) not null,

course\_name varchar(20) not null,

course\_hour int(10) unsigned not null,

course\_credit int(1) unsigned not null,

foreign key(teacher\_id) references teacher(teacher\_id)

);

#创建成绩信息表

create table score (

student\_id varchar ( 10 ) not null,

course\_id varchar ( 10 ) not null,

score int(3) not null,

primary key(student\_id,course\_id),

foreign key(student\_id) references student(student\_id),

foreign key(course\_id) references course(course\_id)

);

#创建用户信息表

create table user (

user\_id varchar ( 10 ) not null,

password varchar ( 10 ) not null,

isAdmin int(1) not null

);

-- 创建触发器

# 创建删除学生之前删除成绩的触发器

create trigger deleteScoreAndUser

before delete

on student

for each row

begin

delete from score where student\_id = old.student\_id;

delete from user where user\_id = old.student\_id;

end;

create trigger deleteUser

before delete

on teacher

for each row

begin

delete from user where user\_id = old.teacher\_id;

end;

# 插入学生和老师信息到user表

create trigger addUserForStudent

after insert

on student

for each row

begin

insert into user values(new.student\_id,new.student\_id,0);

end;

create trigger addUserForTeacher

after insert

on teacher

for each row

begin

insert into user values(new.teacher\_id,new.teacher\_id,1);

end;

-- 插入数据

#向学生信息表插入数据

insert into student values('2019211001','王炬欣','男','计算机学院','计算机科学与技术');

insert into student values('2019211002','王嘉欣','男','计算机学院','计算机科学与技术');

insert into student values('2019211003','吴世浩','男','计算机学院','计算机科学与技术');

insert into student values('2019211004','李志恒','男','计算机学院','计算机科学与技术');

insert into student values('2019211005','张圣杰','男','计算机学院','计算机科学与技术');

#向老师信息表插入数据

insert into teacher values('1664011001','何成春');

insert into teacher values('1664011002','何庚坤');

insert into teacher values('1664011003','刘静');

insert into teacher values('1664011004','牛奥林');

insert into teacher values('1664011005','郑煜');

#向课程信息表插入数据

insert into course values('001','1664011001','离散数学','64','4');

insert into course values('002','1664011002','数据库原理','64','4');

insert into course values('003','1664011003','数据结构','64','4');

insert into course values('004','1664011004','计算机网络','64','4');

insert into course values('005','1664011005','操作系统','64','4');

#向成绩信息表插入数据

insert into score values('2019211001','001','78');

insert into score values('2019211001','002','90');

insert into score values('2019211001','003','84');

insert into score values('2019211001','004','70');

insert into score values('2019211001','005','75');

insert into score values('2019211002','001','66');

insert into score values('2019211002','002','90');

insert into score values('2019211002','003','77');

insert into score values('2019211002','004','66');

insert into score values('2019211002','005','69');

insert into score values('2019211003','001','56');

insert into score values('2019211003','002','61');

insert into score values('2019211003','003','90');

insert into score values('2019211003','004','87');

insert into score values('2019211003','005','67');

insert into score values('2019211004','001','85');

insert into score values('2019211004','002','80');

insert into score values('2019211004','003','69');

insert into score values('2019211004','004','90');

insert into score values('2019211004','005','92');

insert into score values('2019211005','001','37');

insert into score values('2019211005','002','59');

insert into score values('2019211005','003','61');

insert into score values('2019211005','004','70');

insert into score values('2019211005','005','44');

-- 创建函数

# 创建获取学生课程学分的函数

create

function getCredit(credit int,score int)

returns decimal(10,2)

comment '根据成绩获取学分'

begin

declare result decimal(10,2);

if (score >= 90) then

set result = credit \* 1.0;

elseif (score >= 80) then

set result = credit \* 0.8;

elseif (score >= 60) then

set result = credit \* 0.6;

else

set result = credit \* 0.0;

end if;

return result;

end;

# 功能测试

-- 登录/修改密码

# 查询学生密码

select password from user where user\_id = '2019211001';

# 查询是否有管理权限

select isAdmin from user where teacher\_id = '1664011001';

#修改密码

update user set password = '123456' where student\_id = '2019211001';

-- 查询成绩

# 查询全部

select stu.student\_id,

stu.student\_name,

stu.student\_sex,

stu.student\_college,

stu.student\_major,

sum(if(c.course\_name = "离散数学", sc.score, 0)) AS "离散数学",

sum(if(c.course\_name = "数据库原理", sc.score, 0)) AS "数据库原理",

sum(if(c.course\_name = "数据结构", sc.score, 0)) AS "数据结构",

sum(if(c.course\_name = "计算机网络", sc.score, 0)) AS "计算机网络",

sum(if(c.course\_name = "操作系统", sc.score, 0)) AS "操作系统",

round(ifnull(avg(sc.score), 0), 2) AS "平均分",

ifnull(sum(sc.score), 0) AS "总分"

from student stu left join score sc on stu.student\_id = sc.student\_id left join course c on c.course\_id = sc.course\_id

GROUP BY stu.student\_id

order by ifnull(sum(sc.score), 0) DESC ;

# 按学号查询

select stu.student\_id,

stu.student\_name,

stu.student\_sex,

stu.student\_college,

stu.student\_major,

sum(if(c.course\_name = "离散数学", sc.score, 0)) AS "离散数学",

sum(if(c.course\_name = "数据库原理", sc.score, 0)) AS "数据库原理",

sum(if(c.course\_name = "数据结构", sc.score, 0)) AS "数据结构",

sum(if(c.course\_name = "计算机网络", sc.score, 0)) AS "计算机网络",

sum(if(c.course\_name = "操作系统", sc.score, 0)) AS "操作系统",

round(ifnull(avg(sc.score), 0), 2) AS "平均分",

ifnull(sum(sc.score), 0) AS "总分"

from student stu left join score sc on stu.student\_id = sc.student\_id left join course c on c.course\_id = sc.course\_id

where stu.student\_id = '2019211001';

# 按学院查询

select stu.student\_id,

stu.student\_name,

stu.student\_sex,

stu.student\_college,

stu.student\_major,

sum(if(c.course\_name = "离散数学", sc.score, 0)) AS "离散数学",

sum(if(c.course\_name = "数据库原理", sc.score, 0)) AS "数据库原理",

sum(if(c.course\_name = "数据结构", sc.score, 0)) AS "数据结构",

sum(if(c.course\_name = "计算机网络", sc.score, 0)) AS "计算机网络",

sum(if(c.course\_name = "操作系统", sc.score, 0)) AS "操作系统",

round(ifnull(avg(sc.score), 0), 2) AS "平均分",

ifnull(sum(sc.score), 0) AS "总分"

from student stu left join score sc on stu.student\_id = sc.student\_id left join course c on c.course\_id = sc.course\_id

where stu.student\_college = '计算机学院'

GROUP BY stu.student\_id;

order by ifnull(sum(sc.score), 0) DESC ;

# 按专业查询

select stu.student\_id,

stu.student\_name,

stu.student\_sex,

stu.student\_college,

stu.student\_major,

sum(if(c.course\_name = "离散数学", sc.score, 0)) AS "离散数学",

sum(if(c.course\_name = "数据库原理", sc.score, 0)) AS "数据库原理",

sum(if(c.course\_name = "数据结构", sc.score, 0)) AS "数据结构",

sum(if(c.course\_name = "计算机网络", sc.score, 0)) AS "计算机网络",

sum(if(c.course\_name = "操作系统", sc.score, 0)) AS "操作系统",

round(ifnull(avg(sc.score), 0), 2) AS "平均分",

ifnull(sum(sc.score), 0) AS "总分"

from student stu left join score sc on stu.student\_id = sc.student\_id left join course c on c.course\_id = sc.course\_id

where stu.student\_major = '计算机科学与技术'

GROUP BY stu.student\_id

order by ifnull(sum(sc.score), 0) DESC ;

# 按课程查询

select stu.student\_id,

stu.student\_name,

sc.score

from student stu left join score sc on stu.student\_id = sc.student\_id left join course c on c.course\_id = sc.course\_id

where c.course\_name = '数据库原理'

order by sc.score DESC ;

-- 修改成绩

update score set score = '60' where student\_id = '2019211001' and course\_id = '001';

-- 删除成绩

delete from score where student\_id = '2019211001' and course\_id = '001';

-- 功能

# 查询每门课程不及格人数

select course\_name '课程名',count(case when score <60 then 1 else null end) '不及格人数' from score,course where score.course\_id = course.course\_id GROUP BY score.course\_id;

# 查询每门课程良好人数

select course\_name '课程名',count(case when score >= 60 and score < 90 then 1 else null end) '良好人数' from score,course where score.course\_id = course.course\_id GROUP BY score.course\_id;

# 查询每门课程优秀人数

select course\_name '课程名',count(case when score > 90 then 1 else null end) '优秀人数' from score,course where score.course\_id = course.course\_id GROUP BY score.course\_id;

# 老师查询任课信息

select course\_name from course where teacher\_id = (select teacher\_id from teacher where teacher\_name = '牛奥林');

# 查询学生某门课程获得学分

select getCredit(course\_credit,score) '学分' from score,course where score.course\_id = course.course\_id and course\_name='离散数学' and student\_id='2019211001';

**五、系统实现与测试**

**1、开发平台和工具选择**

Powerdesign ： 方便地设计E-R图，并转换物理模型和sql语句

Xmind：方便地画出功能结构图

Processon：方便地画出数据流图

Navicat 15 for MySQL ：mysql图形化界面，方便检查mysql语句功能

**2、系统测试**

**查询全部学生成绩**



按学号查询



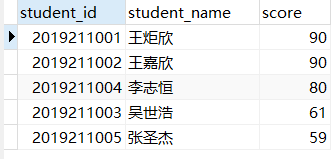
按学院查询



按专业查询



按课程查询



查询每门课程不及格/良好/优秀人数







查询老师任课信息



查询学生某门课程获得学分

