教务管理系统（选课、成绩）

一、概念模型设计

实体： 学生 教师 课程 班级

属性： 学生（学号，姓名，年龄，性别，籍贯） 成绩

教师（教师工号，姓名）

课程（课程号，课程名，课时，学分）

班级（班级号，班级名称，人数）

班主任

管理

教师

1 1

班级

n

教授

1

属于

m

选修

课程

学生

n

m n

二、逻辑模型设计

**第一步**

**学生（学号，姓名，年龄，性别，籍贯）**

**教师（教师工号，姓名）**

**课程（课程号，课程名，课时，学分）**

**班级（班级号，班级名称，人数，教师号）**

**选修（学号，课程号，成绩）**

**属于（班级号，学号）**

**管理（班级号，教师号）**

**教授（教师号，课程号）**

**第二步**

**学生（学号，姓名，年龄，性别，籍贯，班级号）从表**

**教师（教师工号，姓名）**

**课程（课程号，课程名，课时，学分）**

**班级（班级号，班级名称，人数，教师号）主表**

**选修（学号，课程号，成绩）**

**教授（教师号，课程号）**

**1、12，计算机，计算机导论 修改1条记录**

**修改异常 主键 学号 ，课程号**

**插入异常**

**删除异常**

**1 ER图中信息不能丢失**

**2信息不能冗余**

借阅

图书

学生

m n

借阅（图书编号，学号，借书时间，还书时间）

Create database 数据库名；

Show databases;

Alter database 数据库名

Drop database数据库名；

**借书时间 datetime**

三、数据库表的设计

Int I;double b;char c;String s;

32 64 8

**学生（学号，姓名，出生日期，性别，家庭住址，班级号） student**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **Sno** | **学号** | **Char（10）** | **主键** |
| **Sname** | **姓名** | **Char（10）** | **非空** |
| **Birthday** | **出生日期** | **Date** |  |
| **Sex** | **性别** | **Char(2)** | **女** |
| **Address** | **家庭住址** | **Char(40)** |  |
| **Phone** | **联系电话** | **Char(11)** |  |
| **Email** | **电子邮件** | **Char(20)** |  |
| **Clno** | **班级号** | **Char(4)** | **外键** |

**create table student(sno char(10) primary key ,sname char(8),birthday date,sex char(2) default ‘女’,address varchar(40),phone char(11),email char(20),classno char(4));**

**create table student(sno char(10) ,sname char(8) not null,birthday date,sex char(2) default “女” ,address varchar(40),phone char(11) unique,email char(20),classno char(4),**

**Constraint primary key(sno),**

**Constraint foreign key (classno) references class(clno));**

**show tables;**

**describe student;**

**教师（教师工号，姓名）teacher**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **Tno** | **教工号** | **Char（10）** |  |
| **Tname** | **教师姓名** | **Char（10）** |  |

**课程（课程号，课程名，课时，学分） course**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **Cno** | **课程号** | **Char（10）** |  |
| **Cname** | **课程名** | **Char（10）** |  |
| **Ch** | **课时** | **Tinyint** |  |
| **Credit** | **学分** | **tinyint** |  |

**班级（班级号，班级名称，人数，教师号）class**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **clno** | **班级号** | **Char（4）** |  |
| **clname** | **班级名称** | **Char（10）** |  |
| **Clnum** | **人数** | **Tinyint** |  |
| **Tno** | **教师号** | **Char(10)** |  |

**Create table class (clno char(4) primary key,clname char(10),clnum tinyint,tno char(10))**

**选修（学号，课程号，成绩）**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **Sno** | **学号** | **Char（10）** |  |
| **Cno** | **课程号** | **Char（10）** |  |
| **Score** | **成绩** | **tinyint** |  |

**教授（教师号，课程号）**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **说明** | **数据类型** | **备注** |
| **Tno** | **教工号** | **Char（10）** |  |
| **Cno** | **课程号** | **Char（10）** |  |
|  |  |  |  |

实体 表

实例 记录

关键属性 主键 不能重复 ，不能为空

外键 另外一个表中主键，可以为空 参考 references

1. 主键
2. 设计的时候添加
3. 设计后添加

Alter table class add constraint fk\_tno foreign key (tno) references teacher(tno);

Create table teacher(tno char(10) primary key,tname char(10))

--添加数据

insert into course values

('8545005','数据库原理及应用',64,4),

('8545001','java程序设计',64,4),

('8545002','线性代数',64,4),

('8545003','网页设计',64,4),

('8545004','高等数学',64,4);

insert into student(sno,sname,birthday,address,classno)values

('2017098521','王五','1999-5-6','广西玉林','计科17'),

('2017098515','赵六','1998-5-6','广西玉林','计科17'),

('2017098516','李三','1999-7-6','广西玉林','计科17'),

('2017098517','赵毅','1999-8-6','广西玉林','计科17'),

('2017098518','李四','1999-5-20','广西玉林','计科17'),

('2017098519','张晓华','1999-9-2','广西玉林','计科17'),

('2017098520','魏大伟','1999-12-30','广西玉林','计科17');

insert into class values

('1701','计科17-1',36,'9807008'),

('1702','计科17-2',38,'98070010');

insert into teacher values

('9807008','张老师'),

('9807009','何老师'),

('9807010','李老师'),

('9807011','王老师');

insert into result values

('2017098515','8545005',98),

('2017098516','8545005',98),

('2017098517','8545005',68),

('2017098518','8545005',92),

('2017098519','8545005',58),

('2017098515','8545001',98),

('2017098515','8545002',98),

('2017098515','8545003',48),

('2017098515','8545004',92),

('2017098516','8545004',38),

('2017098518','8545004',91);

insert into teach values

('9807008','8545004'),

('9807009','8545005'),

('9807009','8545003'),

('9807011','8545001');

--修改

update student set sname='王五',sex='女' where sno='2017098515';

--删除

delete from student where sno='2017098515';

--全部删除数据，数据表回到初始状态

truncate from student;

--简单的查询

select \* from student;

select sname,address from student where sname='王五';

select sno,cno,score-5 from result;

--1.查询成绩不及格的学生，学号，课程号，成绩

select sno,con,score from result where score<60;

--2.查询课程表中，数据库原理及应用，Java程序设计

select cno,cname from course where cname in('数据库原理及应用','Java程序设计');

select cno,cname from course where cname ='数据库原理及应用' or cname ='Java程序设计';

select cno,cname from course where cname ='数据库原理及应用' union select cno,cname from course where cname ='Java程序设计';

--3.查询成绩在60-80之间的学号，课程号，成绩

select cno,sno,score from result where score between 60 and 80;

select cno,sno,score from result where score>=60 and score<=80;

select cno,sno,score from result where not (score<60 or score>80);

--4.查询电话号码非空的学生信息

select \* from student where phone is not null;

--5.查询有那些同学选了课

select distinct sno from result;

--6.查询姓张的教师信息

select \* from teacher where tname like '张%';

--7.查询带有设计

select \* from course where cname like '%设计%';

--8.名字两个字，姓李

select \* from student where sname like '李\_';

--查询姓不姓张的教师信息

select \* from teacher where tname not like '张%';

--查询每个学生的平均成绩avg 最高成绩 max，最低成绩 min，sum 个数 count

select sno, avg(score) from result group by sno;

update result set score=68 where sno='2017098515' and cno='8545005';

--查询平均成绩大于70分的课程号，平均成绩,

select cno,avg(score) from result group by cno having avg(score)>70;

--查询成绩大于70分的课程的平均成绩，课程号

select cno,avg(score) from result where score>70 group by cno;

--查询没有选课的学生sno

方法一

1）查询选课的学生信息

select distinct sno from result;

2）

select sno from student where sno not in(select distinct sno from result);

方法二左外连接

select student.sno,sname from student left outer join result on student.s

no=result.sno where score is null;

选课的学生学号

方法三 =any 相当与in

select sno,sname from student where sno =any(select distinct sno from result);

--查询没有选数据库原理及应用的学生信息

select sno from student where sno not in(select distinct sno from result where cno=(

select cno from course where cname='数据库原理及应用'));

--

select sno from student where sno not in(select sno from result join course on result.cno=course.cno where cname='数据库原理及应用');

--左外连接

select student.sno,sname,rst.sno,cno,score from student left outer join

(select \* from result where cno=(select cno from course where cname='数据库原理及英语')) rst

on student.sno=rst.sno where score is null;

--查询数据库原理应用的成绩信息

--方法一

1）select cno from course where cname ='数据库原理及应用';

2）select sno,score from result where cno=(select cno from course where cname ='数据库原理及应用');

--方法二

select sno,score from result,course where result.cno=course.cno and cname='数据库原理及应用';

--查询教师张老师教授课程的学生成绩

思考步骤

1)查询张老师的教工号

select tno from teacher where tname='张老师';

2)根据教工号查询课程号

select cno from teach where tno=();

3)根据课程号查询成绩信息

select score,sno from result where cno=

(select cno from teach where tno=

(select tno from teacher where tname='张老师'));

--查询每位老师的教授的课程门数，教工号,课程门数

select tno,count(\*) from teach group by tno

--查询上两门课的教师姓名

select tname from teacher where tno=

(select tno from teach group by tno having count(\*)=2);

--查询没有选课的学生人数

select count(sno) from student where sno not in (select distinct sno from

result);

--查询张老师教授的学生信息

select \* from student where sno in(

select sno from result where cno in(

select cno from teach where tno=

(select tno from teacher where tname='张老师')));

--查询李三同学的成绩信息

--方法1

select cno,score from result where sno=(select sno from student where sname='李三');

--方法2--查询李三同学的成绩信息，学号，姓名，课程名，课程号，成绩

select result.sno,sname,result.cno,cname,score from result,student,course

where result.sno=student.sno and course.cno=result.cno and sname='李三';

--查询各个地方的学生人数

select address,count(sno) from student group by address;

--查询结果（一个字段名） 单值 多值 数值 集合

select count(sno) from student where sno not in

(select distinct sno from result);

select score,sno from result where cno=

(select cno from teach where tno=

(select tno from teacher where tname='张老师'));

--查询结果（多个字段名） 表

mysql> select \* from (select sno,sname,sex from student) s;

--什么情况使用单表--什么情况使用多表

最后要求查询的字段名在一个表内，两种方法都可以，要求查询的字段在不同的表中

--多表一定注意表间关联，在条件里写出关联

select \* from course,student,result

where course.cno=result.cno and result.sno=student.sno;

--国际标准内连接

select result.cno,cname,score from course inner join result

on course.cno=result.cno

where score>=60;

--国际标准外连接

left outer 左外连接 左表 中的数据全部显示出来，右表中没有相配，留空

select student.sno,sname,result.sno,cno,score from student left outer join result on student.sno=result.sno;

right outer 右外连接 右表 中的数据全部显示出来，左表中没有相配，留空

select student.sno,sname,result.sno,cno,score from result right outer join student on student.sno=result.sno;

full outer 全外连接 左、右表中的数据全部显示出来

查询 和赵毅同学同乡的同学

Select sname from student where address=(Select address from student where sname=’赵毅’);

自身连接

mysql> select sd.sname,sd.address from student s inner join student sd

ess=sd.address where s.sname='赵毅';

*笛卡尔运算 非同构*

*集合运算 同构*

*并 Aub union*

*交 anB intersect*

*差 a-b minus*

--查询课程表中，数据库原理及应用，Java程序设计

select cno,cname from course where cname ='数据库原理及应用' union select cno,cname from course where cname ='Java程序设计';

--查询成绩在60-80之间的学号，课程号，成绩

Select cno,score from result where score<=80 intersect select cno,score from result where score>=60

--查询每门课的平均成绩

课程号，课程名，平均成绩 result course 按照课程号分组

Select result.cno 课程号,cname 课程名,avg(score) 平均成绩 from result,course where result.cno=course.cno group by result.cno,cname

--查询每个学生的平均成绩 学号,姓名,班级,平均成绩

--any 任意一个SOME =any(1,2,3) >=any(1,2,3) <=any(1,2,3) <>any(1,2,3)

--all 所有 >=all(1,2,3) <=all(1,2,3)

--查询成绩表中的最高分

select sno,score from result where score=(select max(score) from result);

select sno,score from result where score>=all(select score from result);

select sno,score from result where score<=all(select score from result);

select sno,sname from student where sno =any(select distinct sno from result);

--exists 存在

不相关子查询 先内后外

select \* from result exists(select \* from result where score=100 )

相关子查询 先外后内，外内交替

--查询=98分的学生的学号，姓名

select sno,sname from student where sno in(select sno from result where score=98);

--连接

Select student.sno,sname from student,result where student.sno=result.sno and score=98;

--相关子查询

Select sno,sname from student where exists (select \* from result where sno=student.sno and score=98);

--函数 数学函数，日期函数，字符函数

If(判断条件,值1,值1)

select if(score>=60,if(score>=70, if(score>=80, if(score>=90, '优', '良'), '中'), '及格'),'不及格') 成绩 from result where cno='8545005';

select sno,case cno when '8545005' then '数据库原理及应用' when '8545004' then 'Java程序设计' when '8545003' then '线性代数' when '8545002' then '网页设计' else '高等数学' end ,score from result;

查询

按照下面格式

学号 数据库原理，java程序设计，线性代数，网页设计 平均分 排名

0001 98 98 23 86 86.2

select sno,sum(if(cno='8545005',score,0)) 数据库原理, sum(if(cno='8545004',score,0)) Java程序设计 from result group by sno;

select student.sno 学号, sname 姓名, sum(case result.cno when '8545005' then score else 0 end) 数据库原理, sum(if(result.cno='8545004',score,0)) Java程序设计,avg(score) 平均成绩 from result,student where result.sno=student.sno group by student.sno,sname;

事务：

|  |  |
| --- | --- |
|  |  |
| A | 200 |
| B | 200 |

A向 B账号转账100元

1. update a=a-100

数据库出现问题

1. update b=b+100 没有成功

a=100 b=200

A向 B账号转账100元

开始事务 ◎start transaction a=200 b=200 旧的一致状态

{update a=a-100

有问题

update b=b+100}

如果出现问题 回滚 rollback◎

如果没有问题 提交 commit a=100 b=300 新的一致状态

ACID

隔离性：并发处理

|  |  |  |
| --- | --- | --- |
| A | T1 | T2 |
| A=1000 b=1000 |  |  |
| A=900 | update a=a-100 |  |
| B=1100 | update b=b+100 |  |
|  |  | Select \* from account b=1100 |
| A=1000 b=1000 | rollback |  |
|  |  |  |

读取未提交的数据

隔离等级

1. Read uncommitted 读取未提交的数据 脏数据

2 Read committed 读取提交的数据

1. repeatable read 可以重复读 默认等级

4 serializable

(1)default-storage-engine=innodb 修改my.ini 重新启动数据库服务器

(2)set autocommit=0 设置非自动提交

create table account(

id int primary key auto\_increment,

name varchar(10),money float);

insert into account(name,money)values ('a',1000),

('b',1000);

set session transaction isolation level read uncommitted;

set session transaction isolation level read committed;

不可重复读，（同一个事务内，读取的值应该一致的）

|  |  |  |  |
| --- | --- | --- | --- |
| A | T1 | T3 | T2（查询B的值） |
| A=1000 b=1000 |  |  |  |
| A=900 | update a=a-100 |  |  |
| B=1100 | update b=b+100 |  |  |
| A=900 b=1100 | Commit |  | Select \* from account b=1100 |
| A=900 b=1100 |  | update a=a-100 |  |
|  |  | update b=b+100 |  |
| A=800 b=1200 |  | Commit |  |
|  |  |  | Select \* from account  B=1200 |

幻读

|  |  |  |
| --- | --- | --- |
| A | T1 | T2（查询B的值） |
|  |  |  |
|  |  | Select \* from account |
|  | Insert into account values(‘c’,1000) |  |
|  | Commit |  |
|  |  | Select \* from account |
|  |  |  |
|  |  |  |
|  |  |  |

丢失修改

|  |  |  |
| --- | --- | --- |
| A | T1 | T2（查询B的值） |
| A=1000 b=1000 | update a=a-100 |  |
|  | update b=b+100 |  |
|  |  |  |
| A=1000 b=1000 |  | update a=a-100 |
|  |  | update b=b+100 |
| A=900 b=1100 | Commit |  |
| A=900 b=1100 |  | commit |
|  |  |  |

总结：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 问题 | 原因 | 解决方法 |
| 读脏数据（数据行） | 读了未提交的数据 | 写读冲突 | 写数据的时候，不能读取，直到写的事务结束 |
| 不可重复读 | 读的过程中，有人修改数据 | 读写冲突 | 读数据的过程中，其他事务不能写数据，直到读的事务结束 |
| 幻读（表） |  | 写读冲突 | 写（添加数据）数据的时候，不能读取，直到写的事务结束 |
| 丢失修改 |  | 写写冲突 | 写数据时候，不允许其他事务写数据，直到事务提交 |
| 封锁协议 LOCK |  |  |  |
|  | S | X |  |
| S | TRUE | FALSE |  |
| X | FALSE | FALSE | REPEATALBE READ |

--查询所有学生的选课情况，学号，姓名，课程号，成绩，没有选课的同学，成绩留空

create procedure proc\_allStudent()

begin

select student.sno,sname,result.sno,cno,score from student left outer join result on student.sno=result.sno;

end //

--查询选修某位老师课程的学生学号和成绩

delimiter //

create procedure proc\_StudentByteacher(in p\_tname char(10))

begin

select score,sno from result where cno=

(select cno from teach where tno=

(select tno from teacher where tname=p\_tname));

end //

--添加数据 insert update delete select

create procedure proc\_insert\_course(in p\_cno char(10),

in p\_cname char(10),in p\_ch tinyint,in p\_credit tinyint)

begin

insert into course values

(p\_cno,p\_cname,p\_ch,p\_credit);

end //

--查询没有选课的学生人数()

create procedure proc\_getUnselectcount(out num tinyint)

begin

select count(sno) into num from student where sno not in

(select distinct sno from result);

end //

--修改成绩，给每科成绩加5分，如果成绩大于100，按100分计算，如果小于100分，按实际分数计算

--找到成绩，

delimiter //

create procedure proc\_updatescore(p\_sno char(10),p\_cno char(10))

begin

declare s\_score tinyint;

select score into s\_score from result where sno=p\_sno and cno=p\_cno;

set s\_score=s\_score+5;

if s\_score>=100 then

update result set score=100 where sno=p\_sno and cno=p\_cno;

else

update result set score=s\_score where sno=p\_sno and cno=p\_cno;

end if;

end //

--修改成绩，给数据库原理课程减3分,网页设计减2分,java程序设计减1分，

create procedure proc\_updatescore2(p\_cname char(10))

begin

declare s\_up tinyint;

declare s\_cno char(10);

select cno into s\_cno from course where cname=p\_cname;

case s\_cno

when '8545005' then set s\_up=3;

when '8545003' then set s\_up=2;

else set s\_up=0;

end case;

update result set score=score-s\_up where cno=s\_cno;

end //

--计算1...100之和

delimiter //

alter procedure proc\_getSum6()

begin

declare s\_i tinyint default 0;

declare s\_sum int default 0;

lop:loop

set s\_sum=s\_sum+s\_i;

set s\_i=s\_i+1;

if s\_i>100 then

leave lop;

end if;

end loop lop;

select s\_sum;

end //;

--127 0-127

create procedure proc\_getSum1()

begin

declare s\_i tinyint default 0;

declare s\_sum int default 0;

repeat

set s\_i=s\_i+1;

set s\_sum=s\_sum+s\_i;

until s\_i>100 end repeat;

end //;

create procedure proc\_getSum3(out s\_sum int)

begin

declare s\_i tinyint default 0;

while s\_i<=100

do

set s\_i=s\_i+1;

set s\_sum=s\_sum+s\_i;

end while;

select s\_sum;

end //

-- continue exit undo

create procedure proc\_updatescore2(p\_cname char(10))

begin

declare s\_up tinyint;

declare s\_cno char(10);

declare no\_such\_table condition for 1146 ;

declare continue handler for no\_such\_table set @info='no such table!';

select cno into s\_cno from course1 where cname=p\_cname;

case s\_cno

when '8545005' then set s\_up=3;

when '8545003' then set s\_up=2;

else set s\_up=0;

end case;

update results set score=score-s\_up where cno=s\_cno;

end //

create procedure proc\_curor2()

begin

declare p\_sname char(10);

declare p\_i int default 0;

declare p\_sno char(10);

declare cursor\_stu cursor for select sname,sno from student;

open cursor\_stu;

repeat

fetch cursor\_stu into p\_sname,p\_sno;

select p\_sname,p\_sno;

set p\_i=p\_i+1;

until p\_i>=5 end repeat;

close cursor\_stu;

end //

create PROCEDURE addcount4(out count INT)

BEGIN

DECLARE itmp INT;

DECLARE cur\_id CURSOR FOR SELECT id FROM stu;

DECLARE EXIT HANDLER FOR NOT FOUND CLOSE cur\_id;

SELECT count(\*) INTO count FROM stu;

SET @sum=0;

OPEN cur\_id;

REPEAT

FETCH cur\_id INTO itmp;

select itmp;

IF itmp<10

THEN SET @sum= @sum+itmp;

END IF;

UNTIL itmp=0 END REPEAT;

CLOSE cur\_id;

END //

视图 1 复杂性隐藏 2安全 3独立性

存储过程 1 复杂性隐藏 2 一次编译 3网络传输

--学号，姓名，性别，地址

create or replace view v\_student(v\_sno,v\_sname,v\_sex,v\_address)

as

select sno,sname,sex,address from student where classno='1701';

--学生姓名 课程名 成绩 教师姓名

create or replace view v\_stu(学生姓名,课程名,成绩,教师姓名)

as

select sname,cname,score,tname from student,result,course,teacher,teach

where student.sno=result.sno and result.cno=course.cno

and course.cno=teach.cno and teach.tno=teacher.tno and score<60;

--1701班的全体学生成绩，学生姓名 课程名 成绩

--1701班学生成绩的平均分，最高分，最低分

--没有选修够两门课 1 0的同学 学号，姓名，班级

create view v\_st(学号,姓名,班级,选课门数)

as

select student.sno ,sname,classno,count(result.sno) from student left join result

on student.sno=result.sno

group by student.sno ,sname,classno

having count(result.sno)<2;