**《Oracle数据库应用》课程设计报告书**

**题目：选课系统**

**姓名：杨应行**

**学号：20172212**

**组长：杨应行**

**组员：缪求辉、陈锐、周帅、王浩天**

**专业：软件工程**

**班级：软件171班**

**学院：计算机**

**分工：**

**杨应行（20172212）：e-r图制作、功能模块图、查询更新数据、创建表空间、创建三个角色**

**缪求辉（20172071）：e-r图设计、索引、字段、存储过程、模型设计、逻辑和物理结构设计**

**陈 锐（20172097）：创表、加入数据、函数过程、Word文档初期制作**

**周 帅（20172115）：视图、序列、触发器、Word文档后期制作**

**王浩天（2017）：导出表和数据、导入**

**1 概述**

### 1.1.1 问题的提出

随着科技的发展，基本上所有的具有一定数量数据的机构都开始使用计算机数据库来做管理。几乎所有学校也都已经在使用计算机管理数据的机制，大大减少了学校学生成绩管理的工作量。系统概述：通过对学校日常教学管理中的课程、选课、学生、教师、成绩等相关内容进行分析，完成具有学生管理、成绩管理、课程管理等相关功能的小型数据库管理应用系统。基本要求：完成学生、班级、课程表的维护。 完成成绩表的维护、浏览查找 。 按班级、按课程计算平均成绩、最高成绩、最低成绩、及格率。统计某学生、某学期的所有课程的平均成绩。 完成用户管理功能。 完成数据备份与恢复功能，

### 1.1.2 本课题的意义

通过该课程设计，可以达到把数据库理论知识更加的巩固加深，加强动手能力与实践能力，学以致用，与现实生活中的应用充分的结合起来。

1.2设计环境

①  Oracle 11G

② Oracle SQL

**2需求分析**

2.1 系统功能要求设计

此系统实现如下系统功能：

**（**1）使得学生的成绩管理工作更加清晰、条理化、自动化。

（2）通过用户名和密码登录系统，查询课程基本资料，学生所选课程成绩，修改用户密码等功能。 容易地完成学生信息的查询操作。

(3) 设计人机友好界面，功能安排合理，操作使用方便，并且进一步考虑系统在安全性，完整性，并发控制，备份和恢复等方面的功能要求。

2.2 系统模块设计

成绩管理系统大体可以分成二大模块如， 一是学生的基本信息模块，里面应该包含学生的各方面的基本信息；再者便是课程管理模块， 在该模块中应该包含有对学生成绩信息的查询和处理，如平均成绩、最好成绩、最差成绩以及不及格学生的统计等功能模块；再其次还有教师、课程等相关信息的模块；

2.3 数据字典

数据项是数据库的关系中不可再分的数据单位，下表分别列出了数据的名

称、数据类型、长度、取值能否为空。利用SQL Server 2000建立“学生选课”

数据库，其基本表清单及表结构描述如下：

数据库中用到的表：

|  |  |  |
| --- | --- | --- |
| 数据库表名 | 关系模式名称 | 备注 |
| S | 学生表 | 学生学籍信息表 |
| Course | 课程表 | 课程基本信息表 |
| Teacher | 教师表 | 教师基本信息 |
| SC | 选课表 | 学生选课信息 |
| DDept | 学院 | 学生学院信息表 |

S基本情况数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | Not Null | 说明 |
| Sno | char | Primary key | 学号 |
| Sname | varchar | Not Null | 学生姓名 |
| Sdept | varchar |  | 学院 |
| Sage | varchar |  | 年龄 |
| Ssex | char |  | 性别 |
| Dno | char | 外键 | 学院号 |

\

Teacher基本情况数据表，结构如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | Not Null | 说明 |
| Tno | char | Primary key | 教师号 |
| Tname | varchar | Not Null | 教师姓名 |
| Tsex | char |  | 性别 |
| Tdept | varchar |  | 学院 |
| Dno | Char | 外键 | 学院号 |
| Cno | varchar | 外键 | 课程号 |

Course数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| Cno | varchar | 主键（primary key） | 课程号 |
| Cname | varchar | not null | 课程名称 |
| Ctime | char | not null | 课时 |
| Ccredit | varchar | not null | 学分 |

SC情况数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| Cno | char | 外键 | 课程号 |
| Sno | varchar | 外键 | 学号 |
| Grade | char | not null | 成绩 |

Dept情况数据表，结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 约束控制 | 说明 |
| Dno | char | Not null | 学院号 |
| Dname | varchar | Not null | 学院名 |

**3 概念模型设计**

由需求分析的结果可知，本系统设计的实体包括：

（1）学生基本信息：学号，姓名，学院，班级，年龄，性别。

（2）课程基本信息： 课程名，课程号，学时，学分。

（3）教师基本信息：教师号，教师姓名，性别，学院

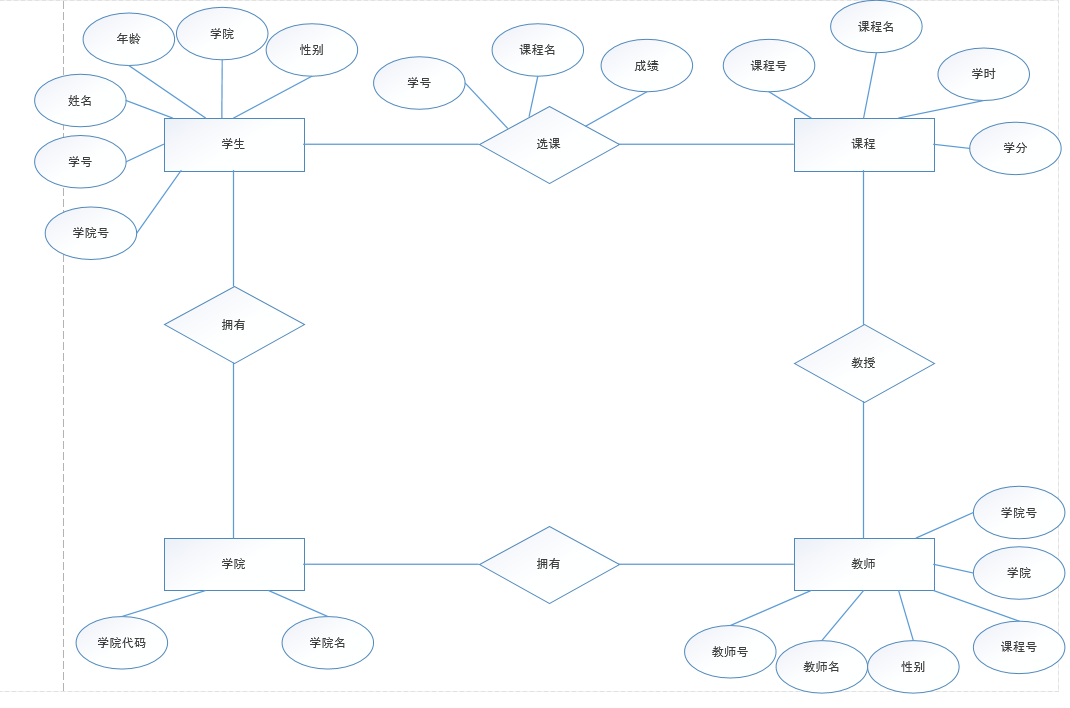
（4）学院基本信息：学院名，学院号

（5）选课基本信息：学号，课程号，成绩

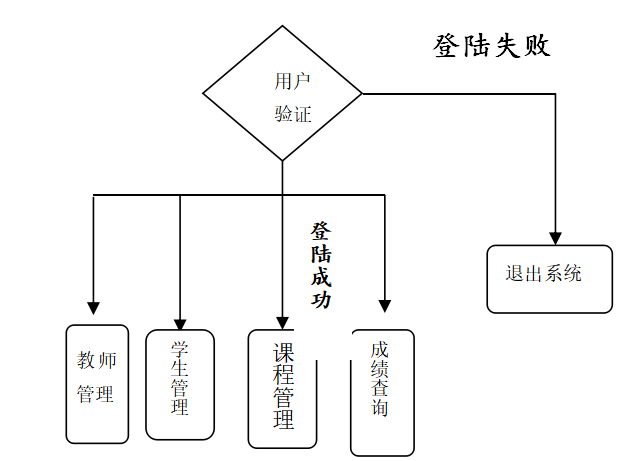
这些实体间的联系包括：

1. 每位学生可以学习多门课程，每门课程可供多位学生学习。
2. 每门课可以由多个老师教，每个老师可以教多门课程
3. 学生每选一门课就可以得到一个成绩，不选此课就不能取得该课程成绩

由上述分析可得到系统的E—R图：

****

**4 设计功能模块-最少4个功能模块**



**5 根据E-R图创建模式对象：创建表、索引、视图、序列 等**

**5.1创表**

**create table S(**

**Sno char(10) not null primary key,**

**Sname varchar(20) not null,**

**Ssex char(10) not null,**

**Sage varchar(5),**

**Sdept varchar(10),**

**Dno char(10) references DDept(Dno)**

**) 学生表**

**create table SC(**

**Grade char(10) not null,**

**Sno char(10) references S(Sno),**

**Cno varchar(20) references Course(Cno)**

**)**

**选课表**

**create table Course(**

**Cno varchar(20) not null primary key,**

**Cname varchar(20) not null,**

**Ctime char(10) not null,**

**Ccredit varchar(5)**

**)**

**课程表**

**create table Teacher(**

**Tno char(10) not null primary key,**

**Tname varchar(20) not null,**

**Tsex char(10) not null,**

**Tdept varchar(10),**

**Cno varchar(20) references Course(Cno),**

**Dno char(10) references DDept(Dno)**

**)**

**教师表**

**create table DDept(**

**Dno char(10) not null primary key,**

**Dname varchar(20) not null**

**)**

**学院表**

**5.2索引**

**create unique index Stusno on S(Sno);**

**create unique index SCsno on SC(Sno);**

**create unique index Coucno on Course(Cno);**

**create unique index Teatno on Teacher(Tno);**

**create unique index DDepdno on DDept(Dno);**

**5.3视图**

**create view S\_G(Sno,Gavg)**

**as**

**select Sno,AVG(Grade)**

**from SC**

**GROUP by sno;**

**create view IS\_student**

**as**

**select Sno,Sname,Sage**

**from S**

**grant create view to scott;权限**

**5.5序列**

**create SEQUENCE student\_seq**

**start with 20170000**

**increment by 1**

**maxvalue 20171000**

**cache 10;**

**6 往数据库加入数据（组员的信息）**

INSERT INTO S VALUES('20172212','杨应行','男',21,'计算机','JS');

INSERT INTO S VALUES('20172071','缪求辉','男',20,'法学','SX');

INSERT INTO S VALUES('20172097','陈锐','男',21,'英语','WY');

INSERT INTO S VALUES('20172115','周帅','男',20,'电信','DX');

INSERT INTO S VALUES('20162159','王浩天','男',21,'数学','SX');

INSERT INTO SC VALUES('20172212','001',77.5);

INSERT INTO SC VALUES('20172071','003',89);

INSERT INTO SC VALUES('20172097','004',86);

INSERT INTO SC VALUES('20172115','005',82);

INSERT INTO Course VALUES('001','数学分析',44,'3');

INSERT INTO Course VALUES('002','普通物理',44,'3');

INSERT INTO Course VALUES('003','微机原理',72,'3');

INSERT INTO Course VALUES('004','数据结构',72,'3');

INSERT INTO Course VALUES('005','操作系统',64,'3');

INSERT INTO Course VALUES('006','数据库原理',64,'3');

INSERT INTO Course VALUES('007','DB\_Design',48,'3');

INSERT INTO Course VALUES('008','程序设计',56,'3');

INSERT INTO Teacher VALUES('9401','王成钢','男','计算机','001','JS');

INSERT INTO Teacher VALUES('9402','李正科','男','计算机','002','JS');

INSERT INTO Teacher VALUES('9403','严敏','女','数信','003','SX');

INSERT INTO Teacher VALUES('9404','赵高','男','电信','004','DX');

INSERT INTO Teacher VALUES('9405','李正科','男','电信','005','DX');

INSERT INTO Teacher VALUES('9406','李玉兰','女','法学院','006','FX');

INSERT INTO Teacher VALUES('9407','王成钢','男','外语','007','WY');

INSERT INTO Teacher VALUES('9408','马悦','女','计算机','008','JS');

INSERT INTO Teacher VALUES('9409','王成钢','男','数信','003','SX');

INSERT INTO Ddept VALUES('JS','计算机');

INSERT INTO Ddept VALUES('SX','数信');

INSERT INTO Ddept VALUES('WY','外国语');

INSERT INTO Ddept VALUES('DX','电信');

INSERT INTO Ddept VALUES('FX','法学院');

7 **根据功能模块完成数据操作语句-如更新或查询某个数据**

**7.1查询全体学生的学号、姓名和年龄**

Select sno,sname,ssex from S;

**7.2查询年龄不在21-23岁之间的学生姓名、系别和年龄**

Select Sname,Sdept,Sage from S where Sage not between 21 and 23;

**7.3将所有学生的年龄增加一岁**

Update S set sage=sage+1;

**8 根据功能模块及第五步完成的数据操作语句，通过PL/SQL完成业务逻辑程序**

**8.1存储过程**

**8.1.1以课程号为参数，查询并输出改课程的平均成绩，以及比改课程平均成绩高的学生信息。**

Create or replace procedure p\_grade(

p\_Cno SC.Cno%TYPE)

as

p\_grade SC.Grade%TYPE;

BEGIN

select avg(Grade) into p\_grade from SC

where SC.Cno=p\_Cno;

DBMS\_OUTPUT.PUT\_LINE(p\_Cno||' '||'average grade is: '||p\_grade);

FOR v\_s in(

select \* from sc where SC.Cno=p\_Cno and grade >p\_grade)LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_s.Cno||' '||v\_s.Sno);

end loop;

EXCEPTION

when NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('The course doesn''t exists');

end p\_grade;

**8.1.2某学生本学期选课情况**

Create or replace procedure p\_course(

p\_sno s.sno%TYPE)

as

p\_cno sc.cno%TYPE;

BEGIN

select cno into p\_cno from sc where sc.sno=p\_sno;

FOR v\_course in(

select \* from course where course.cno=p\_Cno )LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_course.cno||' '||v\_course.cname||' '||v\_course.ccredit);

end loop;

exception

when NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('The course doesn''t exists');

end p\_course;

**8.1.3.对教师表进行更新操作的过程**

create or replace PROCEDURE p\_update(p\_tno teacher.tno%type,

p\_tname teacher.tname%type,

p\_tsex teacher.tsex%type,

p\_tdept teacher.tdept%type

)

AS

Begin

update teacher set tname=p\_tname,tsex=p\_tsex,tdept=p\_tdept where tno=p\_tno;

Exception

When NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('the deptno is invalid');

End;

**8.1.4.以课程号为参数进行删除过程**

create or replace PROCEDURE t\_course(p\_cno course.cno%type)

AS

Begin

delete from course where cno=p\_cno;

Exception

When NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('the deptno is invalid');

End;

**8.2 函数**

**8.2.1输入学号查总成绩**

**create or replace function get\_sum(**

**s\_no1 S.Sno%type)**

**return SC.Grade%type**

**as**

**s\_sumgrade SC.Grade%type;**

**begin**

**select sum(grade) into s\_sumgrade from SC**

**where Sno=s\_no1;**

**return s\_sumgrade;**

**end get\_sum;**

**8.2.2输入课程号查哪些学生选了这节课**

**create or replace function stu\_name(**

**sc\_no1 SC.Cno%type)**

**return S.Sname%type**

**as**

**s\_name S.Sname%type;**

**begin**

**select Sname into s\_name from S**

**where SC.Cno=sc\_no1%type;**

**return s\_name;**

**end stu\_name;**

**8.2.3删除学生的信息**

**create or replace function s\_delete(**

**s\_del S.Sno%type)**

**return S.Sname%type**

**as**

**s\_name S.Sname%type;**

**begin**

**delete from S where Sno=S\_del;**

**exception**

**when no\_data\_found then**

**dbms\_output.put\_line('delete fail');**

**end s\_delete;**

**8.2.4输入学号查平均成绩**

**create or replace function get\_aug(**

**s\_no2 S.Sno%type)**

**return SC.Grade%type**

**as**

**s\_auggrade SC.Grade%type;**

**begin**

**select aug(grade) into s\_auggrade from SC**

**where Sno=s\_no2;**

**return s\_auggrade;**

**end get\_aug;**

**9 根据功能模块增加触发器**

**9.1创建一个触发器:当S表中有数据插入时候，我们在控制台输入你成功插入一条数据**

**create or replace trigger tir\_S\_insert\_data\_success**

**after**

**insert**

**on S**

**declare**

**begin**

**dbms\_output.put\_line('你成功插入一条数据');**

**end;**

**9.2创建一个触发器增加后的年龄大于之前的年龄**

**create or replace trigger checkage**

**before update**

**on S**

**for each row**

**declare**

**begin**

**if:new.Sage<:old.Sage then**

**raise\_application\_error(-2002,'增加后的年龄不能小于增加前的 年龄'||:old.Sage||'||增后:'||:new.Sage);**

**end if;**

**end checkage;**

**10 创建一个新的表空间（TEST）用于存储选课系统数据；创建三个级别的角色（一般用户：有创建表、视图等模式对象的创建修改权限，管理员：有数据备份、启动关闭权限， 维护用户：只有模式对象的查询权限）， 创建三个用户分别分配不同的角色，这三个用户都使用相关表空间。**

**10.1创建一个新的表空间（TEST）用于存储选课系统数据**

create tablespace Test1 datafile

'C:\oracle\Test1.dbf'

SIZE 10M;

grant create tablespace to scott;权限

**10.2创建三个级别的角色**

**create user user1 identified by user1;**

**create user user2 identified by user2;**

**create user user3 identified by user3;**

**grant create user to scott;权限**

**create role role1 identified by firstRole;**

**create role role2 identified by secondRole;**

**create role role3 identified by thirdRole;**

**grant create table,create view to role1;**

**grant DBA to role2;**

**grant select on scott.S to role3;**

**alter user user1 default tablespace Test1;**

**alter user user2 default tablespace Test1;**

**alter user user3 default tablespace Test1;**

**11 导出选课系统数据库里的所有表和数据，导入到以上新建的用户下**

**1) sqlplus system/manager**

**2) 在C盘创建文件夹**

**3)create or replace directory dumpdir as 'C:\oracle';**

**4)grant read, write on directory dumpdir to scott;**

**grant exp\_full\_database, imp\_full\_database to scott;**

**另一个框导出**

**expdp scott/tiger directory=dumpdir dumpfile=S.dmp logfile=scott.log tables=S job\_name=S\_job parallel=3**

**5) show user 要在system**

**6) create user user4 identified by user4;**

**7) grant dba to user4;**

**8) conn user4/user4;**

**9) create table**

**10)create table s as select \*from scott.s;**

**create table sc as select \*from scott.sc;**

**11)conn system/manager**

**grant read, write on directory dumpdir to user4;**

**grant exp\_full\_database, imp\_full\_database to user4;**

**另一个框导入（不能换行，一口气敲完）**

**impdp user4/user4 directory=dumpdir dumpfile=S.dmp remap\_schema=scott:user4 nologfile=yes content=data\_only logfile=import.log**

**12 逻辑和物理结构设计**

11.1由系统E—R图转化而得到的关系模式如下：

（1）学生（学号，姓名，性别，年龄，学院，班级），其主关键字为学号;

（2）课程（课程名，课程号，学时，学分），其中主关键字为课程号；

（3）教师（教师号，教师姓名，性别，学院）

（4）选课（课程号，学号，成绩）其中主关键字为学号和课程号

（5）学院（学院代码，学院名）其中主关键字为学号和课程号。

12.2 确定关系模型的存取方法

在将概念模型转换成物理模型之后，我们可以对物理模型进行设计，双击物理模型的关系，可以对该关系的名称、注释等信息进行查询。可对该关系的属性列进行设计，可分别设置其名称、码、数据类型以及主码、是否为空等。在实际设计中最常用的存取方法是索引发，使用索引可以大大减少数据的查询时间，在建立索引时应遵循：在经常需要搜索的列上建立索引；  在主关键字上建立索引；在经常用于连接的列上建立索引，即在外键上建立索引；在经常需要根据范围进行搜索的列上创建索引，因为索引已经排序，其指定的范围是连续的等规则。才能充分利用索引的作用避免因索引引起的负面作用。

12.3 确定数据库的存储结构

确定数据库的存储结构主要指确定数据的存放位置和存储结构，包括确定关系、索引、日志、备份等的存储安排及存储结构，以及确定系统存储参数的配置。因为该成绩管理系统的数据量小，，所以我们只用把数据存储在使用的电脑硬盘上，不用作更多的安排。