

《数据库系统》上机作业

姓名：王振华

学号：20009200821

一、需求分析

1. 录入一位学生，应包含学号、姓名、性别、出生年月、班级等信息

分析：通过 **insert** 关键字插入对应的 **values** 即可，最后再查询是否成功插入。

2. 按学号、姓名、专业三种方式查询学生基本信息

分析：通过 **select** 关键字选择学号，姓名，专业来查询学生基本信息。

3. 录入一位学生一门课的成绩

分析：通过 **update** 关键字更新学生的成绩数据。

4. 查询一位学生所修的课程、性质（必修或选修）、学期、学分及成绩；
查询他的必修课平均成绩、所有课程平均成绩（平均成绩应按学分加权）

分析：建立一个学生视图，视图中的属性为查询中所要求的属性，此时查询就变得较为简单，加权的平均成绩使用（课程分数*课程学分）/ 课程学分总和。

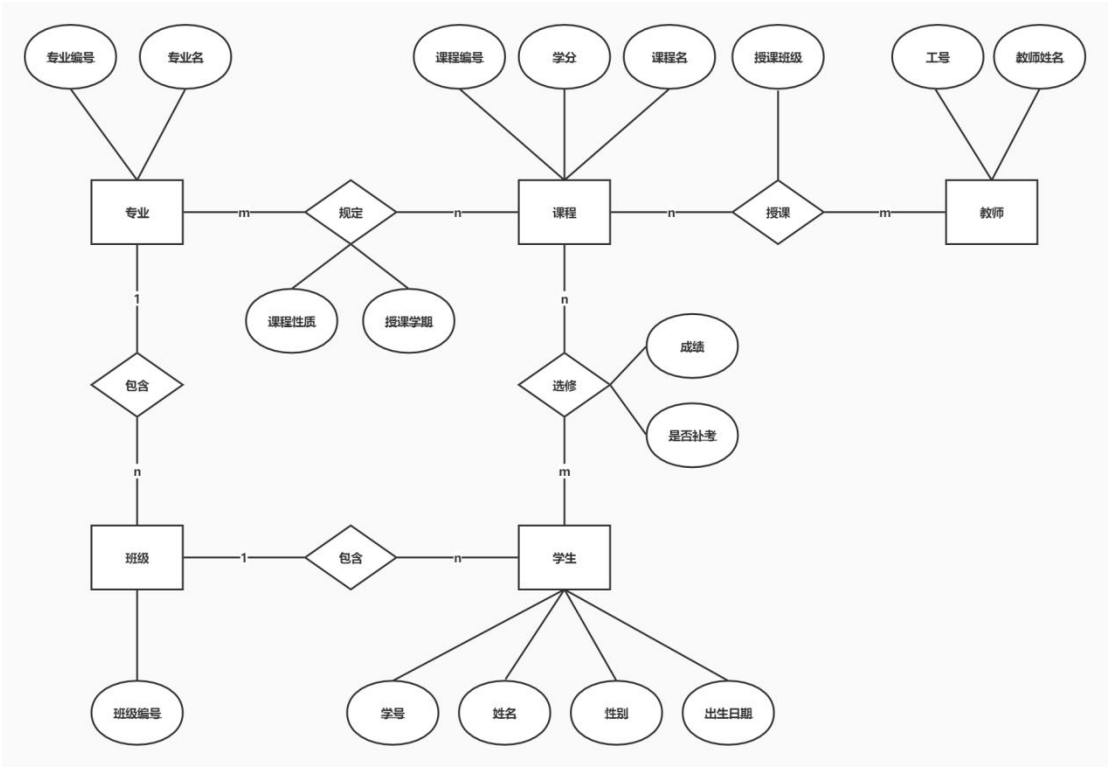
5. 查询一位学生被哪些教师教过课

分析：建立一个老师授课的信息视图，并与学生集合连接查询。

6. 查询快要被开除的学生（距被开除差 3 学分之内）

分析：在前面视图的基础上建立视图，分别筛选出必修课程和选修课程的挂科记录，再进行查询判断该学生是否快要被开除。

二、概念结构设计（ER 图设计）



三、逻辑结构设计（E-R 图转换为关系模型）

由上述的 E-R 图，先建立 5 个基本的关系模型，

即：1.专业 2.班级 3.学生 4.教师 5.课程

再将其中的 3 个多对多的联系转化为下述 3 个关系模型

即：6.规定 7.授课 8.选修

综上：关系模型如下

1. 专业（专业编号， 专业名）

Major(MajorID, MajorName)

(1)主码：MajorID

(2)外码：无

2. 班级（班级编号， 专业编号）

Class(ClassID, MajorID)

(1)主码：ClassID

(2)外码：MajorID

3. 学生（学号， 姓名， 性别， 出生日期， 专业编号， 班级编号）

Student(SNO, SName, Ssex, SBirth, MajorID, ClassID)

(1)主码：SNO

(2)外码：MajorID, ClassID

4. 教师（工号， 教师姓名）

Teacher(TeacherID, TeacherName)

(1)主码：TeacherID

(2)外码：无

5. 课程（课程编号， 课程名， 课程学分）

Course(CourseID, CourseName, CourseCredit)

(1)主码：CourseID

(2)外码：无

6. 规定（专业编号， 课程编号， 课程性质， 授课学期）

Rules(MajorID, ClassID, ClassNature, ClassSchedule)

(1)主码：(MajorID, CourseID)

(2)外码：MajorID, CourseID

7. 授课（课程编号， 工号， 班级编号）

Teach(CourseID, TeacherID, ClassID)

(1)主码：(TeacherID, ClassID)

(2)外码：CourseID, TeacherID, ClassID

8. 选修（学号， 课程编号， 成绩， 是否补考）

StudentGrade(SNO, CourseID, Grade, IsResit)

(1)主码：(SNO, CourseID)

(2)外码：SNO, CourseID

语义 1.学校有若干专业，每个专业每年招若干个班，每个班有若干学生

注：通过表 1 和表 2 和表 3 来实现，其中的属于关系属于 1 对多的关系，通过外码来实现。

语义 2.每个专业有自己的教学计划，规定了该专业相关课程的性质（必修或选修）以及授课学期；例如，数据库课程对计算机专业为必修、在大三上学期，但对数学专业可能为选修、在大三下学期，而中文专业可能不学这门课

注：通过表 6 规定表实现，表中规定了专业的教学课程，课程的性质即为选修还是必修，课程的授课学期。

语义 3.一位教师可以给多个班带课，但不能给一个班带多门课

注：通过表 7 的主码(TeacherID, ClassID)来约束，即主码的唯一性，因此不存在一个老师教授同一个班级一门以上课程的记录。

语义 4.一门课程最多允许学生一次补考；学生达到如下条件之一的被开除：不及格必修课累计达 10 学分、或不及格选修课累计达 15 学分

注：通过视图搜索成绩以及是否补考来判定学生是否已经挂科，在判断该学生的不及格必修课累计达 10 学分、或不及格选修课累计达 15 学分来判断是否需要开除。

语义 5.上述语义未涉及到的事项和细节，可自行做出合理假定

四、功能实现

注：代码中报错为数据库已存在名为“xxx”的对象，运行创建对象后，查看代码自动报错

1. 基本表创建

```
--专业表
CREATE TABLE Major
(
    MajorID varchar(10) not null primary key,
    MajorName varchar(40) not null
);

--班级表
CREATE TABLE Class
(
    ClassID varchar(10) not null primary key,
    MajorID varchar(10) not null references Major(MajorID)
);

--学生表
CREATE TABLE Student
(
    SNO int not null primary key,
    SName varchar(8) not null,
    Ssex varchar(2) not null, check(Ssex='男' or Ssex='女'),
    SBirth varchar(20) not null,
    MajorID varchar(10) not null references Major(MajorID),
    ClassID varchar(10) not null references Class(ClassID)
);

--课程表
CREATE TABLE Course
(
    CourseID varchar(10) not null primary key,
    CourseName varchar(40) not null,
    CourseCredit int not null
);

--教师表
CREATE TABLE Teacher
(
    TeacherID int not null primary key,
    TeacherName varchar(8) not null
);
```

```

--专业规定表
CREATE TABLE Rules
(
    MajorID varchar(10) not null references Major(MajorID),
    CourseID varchar(10) not null references Course(CourseID),
    CourseNature varchar(8) not null, check(CourseNature='选修' or CourseNature='必修'),
    CourseSchedule varchar(20) not null,
    primary key(MajorID, CourseID)
);

--教师课程表
CREATE TABLE Teach
(
    CourseID varchar(10) not null references Course(CourseID),
    TeacherID int not null references Teacher(TeacherID),
    ClassID varchar(10) not null references Class(ClassID)
    primary key(TeacherID, ClassID)
);

--选课成绩表
CREATE TABLE StudentGrade
(
    SNO int not null references Student(SNO),
    CourseID varchar(10) not null references Course(CourseID),
    Grade int not null, check(Grade between 0 and 100),
    IsResit varchar(2) not null, check(IsResit='是' or IsResit='否'),
    primary key(SNO, CourseID)
);

```

2. 视图创建

注：代码中报错为数据库已存在名为“xxx”的对象，运行创建对象后，查看代码自动报错

注：此视图中信息为学生的成绩单，每位学生的课程创建

```

1  GO
2  --建立学生信息视图，便于查看以及搜索
3  CREATE VIEW StudentInfo
4  AS
5  SELECT DISTINCT
6      Student.SNO,
7      SName,
8      MajorID,
9      ClassID,
10     CourseID,
11     Grade,
12     IsResit
13 FROM
14     Student, StudentGrade
15 WHERE
16     Student.SNO = StudentGrade.SNO;
17 GO

```

注：此视图中信息为课程的性质，包含了课程名，后续查询较为方便

```
21 GO
22 --建立课程信息视图，便于查看以及搜索
23 CREATE VIEW CourseInfo
24 AS
25 SELECT DISTINCT
26     Rules.MajorID,
27     Course.CourseID,
28     CourseName,
29     CourseNature,
30     CourseSchedule,
31     CourseCredit
32 FROM
33     Course, Rules
34 WHERE
35     Course.CourseID = Rules.CourseID;
36 GO
```

注：此视图中信息为教师所授课的信息，包含了教师名，后续查询较为方便

```
40 GO
41 --建立教师信息视图
42 --可以直接查询老师教学哪些班级哪些课
43 CREATE VIEW TeacherInfo
44 AS
45 SELECT DISTINCT
46     Teacher.TeacherID,
47     TeacherName,
48     ClassID
49     CourseID,
50     CourseName
51 FROM
52     Teacher, Teach, Course
53 WHERE
54     Teach.TeacherID = Teacher.TeacherID AND
55     Teach.CourseID = Course.CourseID;
56 GO
```

注：此视图中信息为学生选课信息，较视图 1 更加全面，是在上述两个视图的基础上建立的视图

```
60 GO
61 --建立学生选课信息视图，较为全面，即学生的成绩单
62 --此视图基于学生信息视图建立
63 CREATE VIEW Transcript
64 AS
65 SELECT
66     SNO,
67     SName,
68     CourseName,
69     CourseNature,
70     CourseSchedule,
71     CourseCredit,
72     Grade,
73     IsResit
74 FROM
75     StudentInfo, CourseInfo
76 WHERE
77     StudentInfo.CourseID = CourseInfo.CourseID AND
78     StudentInfo.MajorID = CourseInfo.MajorID;
79 GO
```

注：此视图信息为所有学生必修课的挂科记录

```
83 GO
84 --建立学生必修课挂科视图
85 --便于查询
86 CREATE VIEW RequiredCredit
87 AS
88 SELECT
89     SNO,
90     SName,
91     SUM(CourseCredit) 'RequiredFail'
92 FROM
93     Transcript
94 WHERE
95     CourseNature = '必修' AND
96     IsResit = '是' AND
97     Grade < 60
98 GROUP BY SNO, SName;
99 GO
```


注：此视图信息为所有学生必修课的挂科记录

```
103 GO
104 --建立学生选修课挂科视图
105 --便于查询
106 CREATE VIEW ElectiveCredit
107 AS
108 SELECT
109     SNO,
110     SName,
111     SUM(CourseCredit) 'ElectiveFail'
112 FROM
113     Transcript
114 WHERE
115     CourseNature = '选修' AND
116     IsResit = '是' AND
117     Grade < 60
118 GROUP BY SNO, SName;
119 GO
```

3. 数据的生成与导入

(1) 生成数据

通过在 Excel 利用各种函数等生成大量的所需要的测试数据,导出时可以转化为 .csv 后缀文件,该文件自动用逗号分隔,正好符合我们所需要的格式,如下图所示:

文件名(N):	基本表.csv
文件类型(T):	CSV (逗号分隔)(*.csv)

此时将 .csv 的后缀名使用文本文档打开,即 .txt 后缀,发现格式正确,此时可以导入数据库,选择 Flat File Source 即可。

(2) 导入数据

在 SQL Server 中观察是否录入上述 8 个表的信息,下图为其中部分数据截图:

专业表:

	MajorID	MajorNa...
▶	CS001	计算机科学...
	CS002	人工智能
	CS003	电子信息

班级表：

	ClassID	MajorID
▶	202201	CS001
	202202	CS001
	202203	CS002
	202204	CS002
	202205	CS003
	202206	CS003

学生表：

	SNO	SName	Ssex	SBirth	MajorID	ClassID
▶	2022001	许艳芳	女	2001/8/27	CS001	202201
	2022002	林丽娟	女	2001/2/14	CS001	202201
	2022003	魏娜	女	2000/12/31	CS001	202201
	2022004	张亚楠	女	2001/12/4	CS001	202201
	2022005	孙寅秋	女	2001/10/27	CS001	202201
	2022006	袁巍巍	女	2001/9/29	CS001	202201
	2022007	王洁	女	2001/12/31	CS001	202201
	2022008	宋丽	女	2001/5/18	CS001	202201
	2022009	张璐	女	2001/10/18	CS001	202201
	2022010	田旻	女	2001/4/9	CS001	202201
	2022011	吝倩	女	2001/1/3	CS001	202201
	2022012	武杏杏	女	2001/9/6	CS001	202201
	2022013	杨一诺	女	2001/12/29	CS001	202201
	2022014	李霞	女	2001/9/19	CS001	202201
	2022015	郝艳青	男	2001/11/21	CS001	202201
	2022016	谢志宏	男	2001/11/29	CS001	202201
	2022017	苏宇铭	男	2001/9/10	CS001	202201
	2022018	陈杰	男	2001/11/22	CS001	202202
	2022019	张琦	男	2001/3/15	CS001	202202
	2022020	彭良庆	男	2001/1/19	CS001	202202
	2022021	唐国柱	男	2001/5/3	CS001	202202

教师表：

	TeacherID	TeacherN...
▶	10001	韩之柔
	10002	孙名媛
	10003	陶艳
	10004	韩淇
	10005	曹心菲
	10006	秦昭娣
	10007	吕琦
	10008	孙琦
	10009	何丽丽
	10010	孔艳
	10011	卫佳玲
	10012	陈秋菊
	10013	沈虹

课程表：

	CourseID	CourseNa...	CourseCr...
▶	AM1113001	军事理论	2
	CS1121004	计算机导论	2
	CS1121005	计算机导论...	1
	CS1121006	程序设计基础	2
	CS2121001	模拟电子技...	3
	CS2121003	数电、模电...	2
	CS2121007	离散数学	3
	CS2221016	数据结构	3
	CS2221017	数据结构实验	1
	CS2221027	数据结构与...	3
	CS2221028	数据结构与...	1
	CS3121009	计算机组织...	5

规定表：

	MajorID	CourseID	CourseNa...	CourseSc...
▶	CS001	AM1113001	必修	大三上
	CS001	CS1121004	必修	大三下
	CS001	CS1121005	必修	大一上
	CS001	CS1121006	必修	大四下
	CS001	CS2121001	必修	大二上
	CS001	CS2121003	必修	大四下
	CS001	CS2121007	必修	大一上
	CS001	CS2221016	必修	大一上
	CS001	CS2221017	必修	大一上
	CS001	CS2221027	选修	大三上
	CS001	CS2221028	选修	大二下
	CS001	CS3121009	必修	大二上
	CS001	CS3121010	必修	大三上
	CS001	CS3121011	必修	大三上

授课表：

	CourseID	TeacherID	ClassID
▶	HA2113001	10001	202201
	HA2113001	10001	202202
	HA2113001	10001	202203
	HA2113001	10001	202204
	HA2113001	10001	202205
	HA2113001	10001	202206
	HA2113002	10002	202201
	HA2113002	10002	202202
	HA2113002	10002	202203
	HA2113002	10002	202204
	HA2113002	10002	202205
	HA2113002	10002	202206
	HA2113003	10003	202201
	HA2113003	10003	202202

选课表:

	SNO	CourseID	Grade	IsResit
▶	2022001	AM1113001	83	是
	2022001	CS1121004	100	否
	2022001	CS1121005	100	否
	2022001	CS1121006	99	否
	2022001	CS2121001	99	否
	2022001	CS2121003	88	否
	2022001	CS2121007	98	否
	2022001	CS2221016	89	是
	2022001	CS2221017	85	否
	2022001	CS2221027	99	否
	2022001	CS2221028	85	是
	2022001	CS3121009	74	否
	2022001	CS3121010	64	否
	2022001	CS3121011	86	否
	2022001	CS3121012	86	否
	2022001	CS3121013	78	是
	2022001	CS3121014	63	否
	2022001	CS3221018	67	否
	2022001	CS3221020	77	是
	2022001	CS3221022	78	否
	2022001	CS3221023	67	否
	2022001	CS3221024	85	是

4. 功能实现

(1) 录入一位学生，应包含学号、姓名、性别、出生年月、班级等信息
以自己的姓名为例子进行录入并查找，代码及结果如下图：

代码：

```
1  --功能1
2  --录入一位学生，应包含学号、姓名、性别、出生年月、班级等信息
3  --INSERT
4  INTO Student(SNO, SName, Ssex, SBirth, MajorID, ClassID)
5  VALUES (2022999, '王振华', '男', '2002/1/7', 'CS001', '202201');
6
7  --查询该学生是否被成功录入
8  --SELECT *
9  FROM Student
10 WHERE SNO='2022999';
11
12 --删除录入学生
13 --DELETE
14 FROM Student
15 WHERE SNO='2022999';
```

结果：

	SNO	SName	Ssex	SBirth	MajorID	ClassID
1	2022999	王振华	男	2002/1/7	CS001	202201

(2) 按学号、姓名、专业三种方式查询学生基本信息

录入信息后代码及查询结果如下图所示：

代码：

```
1  --功能2
2  --按学号、姓名、专业三种方式查询学生基本信息
3  --先录入一位学生，姓名为我自己，学号为2022999
4  INSERT
5  INTO Student(SNO, SName, Ssex, SBirth, MajorID, ClassID)
6  VALUES (2022999, '王振华', '男', '2002/1/7', 'CS001', '202201');
7
8  --按照学号查询
9  SELECT *
10 FROM Student
11 WHERE SNO='2022999';
12
13 --按照姓名查询
14 SELECT *
15 FROM Student
16 WHERE SNO='2022999';
17
18 --按照专业查询
19 SELECT *
20 FROM Student
21 WHERE MajorID='CS001';
22
23 --删除录入学生
24 delete
25 FROM Student
26 WHERE SNO='2022999';
```

结果：

结果 消息						
	SNO	SName	Ssex	SBirth	MajorID	ClassID
1	2022999	王振华	男	2002/1/7	CS001	202201
	SNO	SName	Ssex	SBirth	MajorID	ClassID
1	2022999	王振华	男	2002/1/7	CS001	202201
	SNO	SName	Ssex	SBirth	MajorID	ClassID
1	2022001	许艳芳	女	2001/8/27	CS001	202201
2	2022002	林丽娟	女	2001/2/14	CS001	202201
3	2022003	魏娜	女	2000/1/...	CS001	202201
4	2022004	张亚楠	女	2001/12/4	CS001	202201
5	2022005	孙黄秋	女	2001/1/...	CS001	202201
6	2022006	袁巍巍	女	2001/9/29	CS001	202201
7	2022007	王洁	女	2001/1/...	CS001	202201
8	2022008	宋丽	女	2001/5/18	CS001	202201
9	2022009	张璐	女	2001/1/...	CS001	202201
10	2022010	田昊	女	2001/4/9	CS001	202201
11	2022011	吝倩	女	2001/1/3	CS001	202201
12	2022012	武杏杏	女	2001/9/6	CS001	202201
13	2022013	杨一诺	女	2001/1/...	CS001	202201
14	2022014	李霞	女	2001/9/19	CS001	202201
15	2022015	郝艳青	男	2001/1/...	CS001	202201
16	2022016	谢志宏	男	2001/1/...	CS001	202201
17	2022017	苏宇铭	男	2001/9/10	CS001	202201
18	2022018	陈杰	男	2001/1/...	CS001	202202
19	2022019	张琦	男	2001/3/15	CS001	202202
20	2022020	彭良庆	男	2001/1/19	CS001	202202
21	2022021	唐国柱	男	2001/5/3	CS001	202202
22	2022022	农飞	男	2001/1/...	CS001	202202
23	2022023	史继宾	男	2001/1/...	CS001	202202
24	2022024	程杰	男	2001/1/...	CS001	202202

(3) 录入一位学生一门课的成绩

代码:

```
1  --功能3
2  --录入一位学生一门课的成绩
3  --先录入一位学生,姓名为我自己,学号为2022999
4  --INSERT
5  INTO Student(SNO, SName, Ssex, SBirth, MajorID, ClassID)
6  VALUES (2022999, '王振华', '男', '2002/1/7', 'CS001', '202201');
7
8
9  --录入一位学生一门课的成绩
10 --INSERT
11 INTO StudentGrade(SNO, CourseID, Grade, IsResit)
12 VALUES (2022999, 'HA2113001', 100, '否');
13
14
15 --查询是否正确录入课程信息
16 --SELECT *
17 FROM Student, StudentGrade
18 WHERE Student.SNO=StudentGrade.SNO and StudentGrade.SNO=2022999;
19
20
21 --删除插入的学生信息
22 --删除学生成绩
23 --DELETE
24 FROM StudentGrade
25 WHERE SNO='2022999';
26
27 --删除录入学生
28 --DELETE
29 FROM Student
30 WHERE SNO='2022999';
```

结果:

此处使用了连接查询检测其他信息是否正确

结果		消息									
	SNO	SName	Ssex	SBirth	MajorID	ClassID	SNO	CourseID	Grade	IsResit	
1	2022999	王振华	男	2002/1/7	CS001	202201	2022999	HA2113001	100	否	

(4) 查询一位学生所修的课程、性质（必修或选修）、学期、学分及成绩；查询他的必修课平均成绩、所有课程平均成绩（平均成绩应按学分加权）

查询一位学生所修的课程、性质（必修或选修）、学期、学分及成绩：

代码：

```
1  --功能4
2  --查询一位学生所修的课程、性质（必修或选修）、学期、学分及成绩；
3  --从视图查询较为方便
4  SELECT
5      SNO,
6      SName,
7      CourseName,
8      CourseNature,
9      CourseSchedule,
10     CourseCredit,
11     Grade,
12     IsResit
13 FROM
14     StudentInfo, CourseInfo
15 WHERE
16     StudentInfo.SNO = '2022001' AND
17     StudentInfo.CourseID = CourseInfo.CourseID AND
18     StudentInfo.MajorID = CourseInfo.MajorID;
```

注：代码中使用了两个视图 StudentInfo 和 CourseInfo 两个视图便于查询以及输出其他有用的信息。

结果：

	SNO	SName	CourseName	CourseNature	CourseSchedule	CourseCredit	Grade	IsResit
1	2022001	许艳芳	军事理论	必修	大三上	2	83	是
2	2022001	许艳芳	计算机导论	必修	大三下	2	100	否
3	2022001	许艳芳	计算机导论实验	必修	大一上	1	100	否
4	2022001	许艳芳	程序设计基础	必修	大四下	2	99	否
5	2022001	许艳芳	模拟电子技术基础	必修	大二上	3	99	否
6	2022001	许艳芳	数电、模电、EDA实验	必修	大四下	2	78	否
7	2022001	许艳芳	离散数学	必修	大一上	3	76	否
8	2022001	许艳芳	数据结构	必修	大一上	2	90	是

查询他的必修课平均成绩（平均成绩应按学分加权）：

代码：

```
21  --查询他的必修课平均成绩、所有课程平均成绩（平均成绩应按学分加权）
22  --必修课平均成绩
23  SELECT
24      SNO,
25      SName,
26      CONVERT(DECIMAL(10, 2), SUM(Grade*CourseCredit)*1.0/SUM(CourseCredit)) '必修课平均成绩'
27 FROM
28     StudentInfo, CourseInfo
29 WHERE
30     StudentInfo.SNO = '2022001' AND
31     CourseNature = '必修' AND
32     StudentInfo.CourseID = CourseInfo.CourseID AND
33     StudentInfo.MajorID = CourseInfo.MajorID
34 GROUP BY SNO, SName;
```

注：代码中使用了 CONVERT 函数，是为了将加权平均成绩从 int 型转换为浮点型，并在被除数后*1.0 进行了格式转换，以便于输出准确的加权平均成绩。

以及 DECIMAL 为精确小数类型，其第一个参数为最大的数值，第二个为保留的小数长度。

结果:

	SNO	SName	必修课平均成绩
1	2022001	许艳芳	77.51

所有课程平均成绩（平均成绩应按学分加权）:

代码:

```
36  --所有课程平均成绩
37  SELECT
38      SNO,
39      SName,
40      CONVERT(DECIMAL(10, 2), SUM(Grade*CourseCredit)*1.0/SUM(CourseCredit)) '平均成绩'
41  FROM
42      StudentInfo, CourseInfo
43  WHERE
44      StudentInfo.SNO = '2022001' AND
45      StudentInfo.CourseID = CourseInfo.CourseID AND
46      StudentInfo.MajorID = CourseInfo.MajorID
47  GROUP BY SNO, SName;
```

结果:

	SNO	SName	平均成绩
1	2022001	许艳芳	77.57

(5) 查询一位学生被哪些教师教过课

代码:

```
1  --功能5
2  --查询一位学生被哪些教师教过课
3  --以查询学号为 2022001 的学生为例
4  SELECT
5      SName '学生姓名',
6      TeacherName '老师姓名',
7      CourseName '课程姓名'
8  FROM
9      Student, TeacherInfo
10 WHERE
11     SNO = '2022001' AND
12     Student.ClassID = TeacherInfo.CourseID;
```

注: 代码中使用了教师信息视图, 是为了在查询学生信息时输出教师的姓名, 这样查询出来的结果更加的直观以及美观。

结果:

	学生姓名	老师姓名	课程姓名
1	许艳芳	韩之柔	毛泽东思想概论
2	许艳芳	孙名媛	马克思主义基本原理
3	许艳芳	陶艳	中国近代史纲要
4	许艳芳	韩淇	思想道德修养与法律基础
5	许艳芳	曹心菲	形势与政策
6	许艳芳	秦昭娣	军事理论
7	许艳芳	吕琦	体育
8	许艳芳	吕琦	大学英语

(6) 查询快要被开除的学生（距被开除差 3 学分之内）

```
1  --功能6
2  --查询快要被开除的学生（距被开除差3学分之内）
3  --先查看挂过科的学生信息
4  SELECT DISTINCT
5      RequiredCredit.SName,
6      RequiredCredit.SNO,
7      RequiredCredit.RequiredFail,
8      ElectiveCredit.ElectiveFail
9  FROM
10     RequiredCredit, ElectiveCredit
11 WHERE
12     RequiredCredit.SNO = ElectiveCredit.SNO
13
14 --进行搜索快要被开除的学生信息
15 SELECT DISTINCT
16     RequiredCredit.SName,
17     RequiredCredit.SNO,
18     RequiredCredit.RequiredFail,
19     ElectiveCredit.ElectiveFail
20 FROM
21     RequiredCredit, ElectiveCredit
22 WHERE
23     RequiredCredit.SNO = ElectiveCredit.SNO AND
24     (
25         RequiredFail >= 7 OR
26         ElectiveFail >= 12
27     );
```

注：代码中使用了两个视图 RequiredCredit 和 ElectiveCredit

RequiredCredit 记录了所有学生的必修课挂科信息

ElectiveCredit 记录了所有学生的选修课挂科信息

然后连接两个视图查询不及格必修课累计达 7 学分、或不及格选修课累计达 12 学分的
学生

结果：

	SName	SNO	RequiredFail	ElectiveFail
1	李霞	2022014	3	4
2	谢...	2022016	5	3
3	彭...	2022020	3	5
4	王...	2022037	5	12
5	张弛	2022044	2	2
6	陈羽	2022057	4	2
7	吕涛	2022069	1	5
8	赵迪	2022083	3	1

	SName	SNO	RequiredFail	ElectiveFail
1	王德海	2022037	5	12
2	王旭	2022099	8	4

上图为挂科学生记录，下图为快要被开除的学生（距被开除差 3 学分之内）

五、遇到的主要问题及解决方法

(1) 生成数据时遇到的问题

数据库的测试需要大量的数据，一个一个编数据显的比较慢，我就放弃了这种想法，然后使用 Excel 表格利用函数以及各种工具生成了大量的数据，并另存为.csv 格式，该格式的分隔符恰好为逗号，将其改为 .txt 后缀后，就可以直接导入至数据库中。

(2) 数据格式所遇到的问题

在编辑学生的属性出生日期时，先是使用了 datetime 格式，但是在实验做到了一半发现该格式的输出如下：

	出生日期
1	2001-08-27 00:00:00.000
2	2001-02-14 00:00:00.000
3	2000-12-31 00:00:00.000
4	2001-12-04 00:00:00.000
5	2001-10-27 00:00:00.000
6	2001-08-08 00:00:00.000

发现其后的格式还输出了具体的小时分钟和秒钟，比较多余，在经过了考虑后还是改为了 varchar 格式。

(3) SQL server 中的函数

代码中使用了 CONVERT 函数，是为了将加权平均成绩从 int 型转换为浮点型，并在被除数后*1.0 进行了格式转换，以便于输出准确的加权平均成绩。

以及 DECIMAL 为精确小数类型，其第一个参数为最大的数值，第二个为保留的小数长度。

六、总结

学习的目的在于将知识能合理顺利的运用，将书本知识化为自己所用，是一个由不知道到知道，了解完善应用的过程，尤其是计算机方面的课程更是如此，必要的上机练习是必不可少的。之前的计算机语言、汇编语言等，都是在上机练习中得到顺利掌握。面对这学期的数据库也是如此的过程，在开始对理论知识的学习，然后进行上机练习，目的在于让我们更好的掌握其知识，熟悉数据库编程语言等。当然上机可也不是轻松是课程，在课前还是应该做些相应的准备。首先在理论知识的学习中必须先打好基础，经过预习、听课、复习、作业四个环节的学习，对于这门课的理论知识和有了一定了解，才便于我们的上机课程，理论与上机的结合这样才有助于我们更好的掌握知识。

此次的数据库上机实验相较于以前所使用的各种语言以及所使用的代码来说，sql 语言所要处理的数据量是远远超出其他。因此需要更加强的逻辑能力，尤其是在写代码之前需要先在纸面上把自己的逻辑捋顺，在写代码前的准备越是充足，写代码时出错的几率就会越小。

总而言之，这次数据库上机加深了我对于数据库这门课的掌握。