1.创建上述表，并定义相应的完整性约束。

create table 系表

(系号 char(4) PRIMARY KEY,

系名 nvarchar(50) UNIQUE,

系主任 nvarchar(10))

create table 学生表

(学号 char(10) PRIMARY KEY,

姓名 nvarchar(10),

性别 char(2) CHECK(性别 IN ('男','女')),

年龄 int CHECK(年龄 between 10 and 50 ),

入学年份 nvarchar(10),

籍贯 nvarchar(100),

系号 char(4),

班长学号 char(10),

FOREIGN KEY(班长学号) REFERENCES 学生表(学号),

FOREIGN KEY(系号) REFERENCES 系表(系号))

create table 课程表

(课程号 char(10) PRIMARY KEY,

课程名 nvarchar(100) UNIQUE,

先修课 char(10) FOREIGN KEY REFERENCES 课程表(课程号),

学分 int CHECK(学分 between 0 and 5))

create table 选课表

(学号 char(10),

课程号 char(10),

成绩 int CHECK(成绩 between 0 and 100 ) default 0,

PRIMARY KEY(学号,课程号),

FOREIGN KEY(学号) REFERENCES 学生表(学号),

FOREIGN KEY(课程号) REFERENCES 课程表(课程号))

create table 学分计算表

(最低成绩 real CHECK(最低成绩 between 0 and 100),

最高成绩 real CHECK(最高成绩 between 0 and 100),

计算比率 real,

CHECK(最高成绩>= 最低成绩))

2.将提供的上述各表的数据导入SQL Server。如果导入过程存在错误，请说明可能是什么原因。

答：存在错误，由于源表没有手机号码，而手机号码必须唯一，因此会导致错误，所以去除手机号码字段。

3.在学生表中插入学号为26，姓名为'李四'，性别为'女'，年龄为20，入学年份为2008，籍贯为'广东'，手机号码为10010001000，班长学号为10的一条记录。

insert into 数据库练习1.dbo.学生表

values('26','李四','女',20,'2008','广东', NULL,'0110')

1. 删除上述记录。

delete from 数据库练习1.dbo.学生表 where  学号='26'

1. 将学生表中的姓名字段的长度改为6个汉字。

alter table 数据库练习1.dbo.学生表 ALTER COLUMN 姓名 nvarchar(6)

1. 为学生表增加一个字段电子邮件，20个字符。

ALTER TABLE 数据库练习1.dbo.学生表

ADD 电子邮件 varchar(20) NULL

1. 对课程表的学分字段上的完整性约束进行修改，使其在0到6之间取值。

ALTER TABLE 数据库练习1.dbo.课程表

DROP constraint CK\_\_课程表\_\_学分\_\_05D8E0BE

1. 为学生表在学号列上创建cluster索引。

CREATE CLUSTERED

INDEX PK\_学生表\_学号 ON 数据库练习1.dbo.学生表(学号)

1. 创建一个视图，计算每门课的最高分。

CREATE VIEW view\_每门课的最高分

AS

SELECT  课程号,MAX(成绩) as 每门课的最高分

FROM 数据库练习1.dbo.选课表

group by 课程号

1. 查找每个学生的学号、姓名、总成绩和平均分。

select 学生表.学号,姓名,sum(成绩) as总成绩,avg(成绩) as 平均分

from 数据库练习1.dbo.学生表 LEFT OUTER JOIN 数据库练习1.dbo.选课表 ON (学生表.学号=选课表.学号)

GROUP BY 学生表.学号,姓名

1. 将6系所有学生的年龄，改为7系学生的平均年龄。

(由于没有7系，改成8系)

update 数据库练习1.dbo.学生表

set 年龄=(select avg(年龄)

from  学生表

where 系号='8')

where 系号='6'

1. 将’曹洪’同学操作系统课程的成绩改为62分。

update 数据库练习1.dbo.选课表

set 成绩=62

where 学号 in

(select 学号

from 数据库练习1.dbo.学生表

where 姓名='曹洪')

AND 课程号 in

(select 课程号

from 数据库练习1.dbo.课程表

where 课程名='操作系统')

13.查找所有学生的姓名、入学年份和籍贯。

select 姓名,入学年份,籍贯

from 数据库练习1.dbo.学生表

1. 列出籍贯为'山东'的同学的所有属性。

select \* from 学生表

where 籍贯='山东'

1. 查找年龄最小的学生的学号和姓名。

select 学号,姓名

from 学生表

where 年龄=(

select min(年龄)

from 学生表)

1. 查找选修了'数据库'的学生的学号。

select 学号

from 选课表 join 课程表

on 选课表.课程号= 课程表.课程号

where 课程名='数据库'

1. 查找选修了'编译技术'的女学生的学号和姓名。

select 学生表.学号,姓名

from 学生表 join 选课表

on 学生表.学号=选课表.学号

join 课程表

on 选课表.课程号= 课程表.课程号

where 课程名='编译技术' and 性别='女'

1. 查找'典韦'同学的班长所选修的课程的课程号。

select 课程号

from 选课表 join 学生表

on 学生表.班长学号 = 选课表.学号

where 姓名='典韦'

1. 查找名字中倒数第二字为'侯'的学生的学号、姓名和所在系的系名。

select 学号,姓名,系号

from 学生表

where 姓名 like '%侯\_'

1. 查找名字以P打头，倒数第三字为L的课程的名字。

select 课程名

from 课程表

where 课程名 like 'p%l\_\_'

1. 查找'甘宁'同学所有选修课程的总分。

select sum(成绩) as 总分

from 选课表 join 学生表

on 选课表.学号=学生表.学号

where 姓名='甘宁'

1. 查找既选修了'数据库'，也选修了'操作系统'的同学。

select a.学号 from

选课表 a join 选课表 b

on a.学号=b.学号

join 课程表 c

on a.课程号=c.课程号

join 课程表 d

on b.课程号=d.课程号

where c.课程名='数据库' and d.课程名='编译技术'

1. 查找没有选修'数据库'课程的学生的学号和姓名。

select 学号,姓名

from 学生表

where 学号 not in

(select 学号

from 选课表

where 课程号 in

(select 课程号

from 课程表

where 课程名='数据库'))

1. 查找'数据库'课程及格了，但'编译技术'没有及格的学生的学号和姓名。

select 学生表.学号,姓名

from 选课表 a join 选课表 b

on a.学号=b.学号

join 课程表 c

on a.课程号=c.课程号

join 课程表 d

on b.课程号=d.课程号

join 学生表

on a.学号=学生表.学号

where c.课程名='数据库' and d.课程名='编译技术' and a.成绩>=60 and b.成绩<60

1. 查找数据库成绩低于数据库课平均成绩的同学的学号和姓名。

select 学号,姓名

from  学生表

where 学号 in

(select 学号

from 选课表 join 课程表

on 选课表.课程号=课程表.课程号

where 课程名='数据库' and 成绩<

(select avg(成绩)

from 选课表 join 课程表

on 选课表.课程号=课程表.课程号

where 课程名='数据库'))

1. 查找与'貂蝉'同学选修课程完全相同的学生的学号和姓名（不能多选也不能少选）。

select 学号,姓名

from  学生表 a

where not exists --貂蝉的课都有

(select \*

from 学生表 b join 选课表 c

on b.学号=c.学号

where  姓名='貂蝉' and not exists

(select \*

from 选课表 d

where d.学号= a.学号 and d.课程号= c.课程号))

and 学号 not in --貂蝉没选的都没有

(select 学号

from 选课表

where 课程号 not in

(select 课程号

from 选课表

where 学号 in

(select 学号

from 学生表

where 姓名='貂蝉')))

1. 查找不仅选修了'貂蝉'同学选修的课程，而且还选修了其他课程的同学。

select 学号,姓名

from  学生表 a

where not exists --貂蝉的课都有

(select \*

from 学生表 b join 选课表 c

on b.学号=c.学号

where  姓名='貂蝉' and not exists

(select \*

from 选课表 d

where d.学号= a.学号 and d.课程号= c.课程号))

and 学号 in --貂蝉没选的也有

(select 学号

from 选课表

where 课程号 not in

(select 课程号

from 选课表

where 学号 in

(select 学号

from 学生表

where 姓名='貂蝉')))

1. 查找'高等数学'平均成绩最高的系的系名。

select 系名

from 选课表 join 学生表

on 学生表.学号=选课表.学号

join 课程表

on 选课表.课程号=课程表.课程号

join 系表

on 系表.系号=学生表.系号

where 课程名='数据库'

group by 系名

having avg(成绩)>= all

(select avg(成绩)

from 选课表

join 学生表

on 学生表.学号=选课表.学号

join 课程表

on 选课表.课程号=课程表.课程号

join 系表

on 系表.系号=学生表.系号

where 课程名='数据库'

group by 系名)

1. 查找至少有一个籍贯为'四川'同学所选修的课程的课程名。

(由于没有四川人，改为浙江)

select distinct 课程名

from 课程表 join 选课表

on 课程表.课程号=选课表.课程号

join 学生表

on 学生表.学号=选课表.学号

where 籍贯 = '浙江'

1. 编写一个函数，能够根据学生获得的成绩计算其获得的学分。并利用该函数检索选修了'数据库'课程的学生的学号和获得的学分。

CREATE FUNCTION func()

RETURNS @table TABLE

(

学号 char(4) primary key not null,

所获学分 real null

)

AS

BEGIN

INSERT INTO @table

select 学号,学分\*计算比率 as 获得学分

from 选课表 join 课程表

on 选课表.课程号 = 课程表.课程号

join 学分计算表

on 选课表.成绩 between 学分计算表.最低成绩 and 学分计算表.最高成绩

where 课程名='数据库'

RETURN

END

SELECT \* FROM dbo.func()

1. 编写一个存储过程，能够根据输入的学号，计算出该学生所在系的学生人数，平均的选课门数和平均成绩，以及该学生在该系的排名。

create proc StuProc

@学号 varchar(100)

as

begin

select 系学生数,

cast(系选课数 as real)/系学生数 as 系平均选课数,

cast(系总成绩 as real)/系学生数 as 系平均成绩,

排名

from

(select COUNT(\*) as 系学生数

from 学生表

where 系号=

(select distinct 系号

from 学生表

where 学号 = @学号))a

join

(select count(\*) as 系选课数

from 学生表 join 选课表

on 学生表.学号=选课表.学号

where 系号=

(select distinct 系号

from 学生表

where 学号 = @学号))b

on 1=1

join

(select sum(成绩) as 系总成绩

from 学生表 join 选课表

on 学生表.学号=选课表.学号

where 系号=

(select distinct 系号

from 学生表

where 学号 = @学号))c

on 1=1

join

(select rank as 排名 from(

select 学号,rank() over(order by 学生总成绩 desc) as rank

from

(select 学生表.学号,sum(成绩) as 学生总成绩

from 学生表 join 选课表

on 学生表.学号=选课表.学号

where 系号=

(select distinct 系号

from 学生表

where 学号 = @学号)

group by 学生表.学号)a)b

where 学号=@学号)d

on 1=1

end

go

exec dbo.StuProc'0101'

回答问题：

1. SQL语言的特点。

① 综合统一，功能一体化，语言风格统一，可以随时修改模式

② 高度非过程化，

③ 面向集合的操作方式

④ 以同一种语法结构提供多种使用方法

⑤ 语言简洁，易学易用

1. 基本表和视图有什么区别？

视图：视图是外模式一级数据结构的基本单位。它是从一个或几个基本表（视图）中导出的表，是从现有基本表中抽取若干子集组成用户的虚表

基本表：基本表的定义指建立基本关系模式,而变更则是指对数据库中已存在的基本表进行删除与修改

区别：

①视图是已经编译好的sql语句，表不是

②视图是窗口，表是物理空间内容

③视图是外模式，表是内模式

1. 创建一个数据库，需要创建几个文件，它们分别是做什么用的？它们对应于三级模式中的哪一级？创建的表存储在什么地方？它们对应于三级模式中的哪一级？

① 主数据文件mdf：primary data file，包含数据库的启动信息，并指向数据库中的其他文件。用户数据和对象可存储在此文件中，也可以存储在次要数据文件中

② 辅助数据文件ndf：Secondary data files，次要数据文件是可选的，由用户定义并存储用户数据。通过将每个文件放在不同的磁盘驱动器上，次要文件可用于将数据分散到多个磁盘上。另外，如果数据库超过了单个 Windows 文件的最大大小，可以使用次要数据文件，这样数据库就能继续增长。

③ 事务日志文件ldf：事务日志文件保存用于恢复数据库的日志信息。每个数据库必须至少有一个日志文件

对应内模式

表存储在mdf或者ndf中 对应模式

1. 可以为表定义哪些完整性约束？它们各自的作用是什么？

① 域完整性，字段取值的合理性，包括检查（CHECK）、默认值（DEFAULT）、不为空（NOT NULL）、外键（FOREIGN KEY）等

② 实体完整性，关系的主关键字不能重复也不能取NULL

③ 参照完整性，主键与外键引用的约束条件，要么与被参照关系中某元组主关键字的值相同，要么取空值

1. 自然连接和等值连接有什么差别？

连接运算符是“=”的为等值连接。

自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉。

1. 子查询分为哪几种？它们之间有什么区别？

① 相关子查询，子查询的查询条件依赖于父查询

② 不相关子查询，子查询的查询条件不依赖于父查询

1. 索引有什么作用和缺点？

作用：加快数据的检索速度，提高系统的性能

缺点：占据物理空间，需要动态的维护

1. 视图有什么优点？什么样的视图是可以更新的？

优点：视图是数据库数据的特定子集，可以禁止用户访问数据库表，而要求用户只能通过视图操作数据，可以简化用户操作，提供一定的数据库逻辑独立性，提供数据安全保护。

一般，行列子集视图是可以更新的