**创建表**

create table student

(学号 char(10) primary key,

姓名 varchar(8),

性别 char(2),

出生日期 datetime,

民族 varchar(20),

身高 tinyint,

体重 decimal(5,2),

奖学金 smallmoney,

手机号 char(11),

备注 text,

系编号 char(2))

1.文本类型：

Char：记录中该列长度相同或差别不大。

学号，手机号，性别，邮政编码，……

Varchar：记录中该列长度差别比较大。

家庭住址，院系名，公司名，部门名，……

Text：长度无法确定，或者长度太长。

备注，评语，简介，……

2.数值类型：表示数量。

（1）整型：只能输入整数。

Tinyint：0-255

Smallint：正负3万

Int：正负20亿

（2）精确类型：需要指定精度和小数位数。

5=3+2

（3）货币类型：

Money：正负900万亿

Smallmoney：正负20万

3.日期类型：表示日期。

Datetime：1753——9999

Smalldatetime：1900——2079

需要指定长度的：char，Varchar

需要指定精度和小数位数：decimal，numeric

其他类型：既不需要指定长度，也不需要指定小数位数。

**创建约束类型**

1.创建表同时创建约束。

Create table 系

(系编号 char(2) constraint pk\_系 primary key,

系名,…)

Create table 选修

(学号,课程编号,成绩,

constraint pk\_系 Primary key(学号,课程编号))

2.先创建表, 然后添加约束.

Create table 选修

(学号 char(11) not null,课程编号 char(3) not null,成绩)

Alter table 选修

add constraint pk\_选修 Primary key(学号,课程编号)

Create table 系

(系编号 char(2) not null, 系名,…)

Alter table 系

add constraint pk\_系 Primary key(系编号)

3.修改约束:先删除,再添加.

Alter table 系 drop constraint pk\_系

Alter table 系

add constraint pk\_系 Primary key(系编号)

**约束代码**

alter table 系 add constraint pk\_系 primary key(系编号)

alter table 学生 add constraint pk\_学生 primary key(学号)

alter table 课程 add constraint pk\_课程 primary key(课程编号)

alter table 选课 add constraint pk\_选课 primary key(学号,课程编号)

alter table 学生 add constraint ck\_学生\_性别 check(性别='男' or 性别='女')

alter table 学生 add constraint ck\_学生\_奖学金 check(奖学金>=0 and 奖学金<=10000)

alter table 课程 add constraint ck\_课程\_学分 check(学分>=1 and 学分<=16)

alter table 选课 add constraint ck\_选课\_成绩 check(成绩>=0 and 成绩<=100)

alter table 学生 add constraint fk\_学生\_系

foreign key(系编号) references 系(系编号)

alter table 选课 add constraint fk\_选课\_学生

foreign key(学号) references 学生(学号)

alter table 选课 add constraint fk\_选课\_课程

foreign key(课程编号) references 课程(课程编号)

create table 系

(系编号 char(2) constraint pk\_系 primary key,

系名 varchar(20),系主任 varchar(8))

create table 学生

(学号 char(11) constraint pk\_学生 primary key,

姓名 varchar(8),

性别 char(2) con straint ck\_学生\_性别 check(性别='男' or 性别='女'),

出生日期 smalldatetime,

奖学金 smallmoney constraint ck\_学生\_奖学金 check(奖学金>=0 and 奖学金<=1000),

手机号 char(11),

系编号 char(2) constraint fk\_学生\_系 foreign key references 系(系编号))

create table 课程

(课程编号 char(3) constraint pk\_课程 primary key,

课程名 varchar(20),

学分 tinyint constraint ck\_课程\_学分 check(学分>=1 and 学分<=16))

create table 选课

(学号 char(11) constraint fk\_选课\_学生 foreign key references 学生(学号),

课程编号 char(3) constraint fk\_选课\_课程 foreign key references 课程(课程编号),

成绩 tinyint constraint ck\_选课\_成绩 check(成绩>=0 and 成绩<=100),

constraint pk\_选课 primary key(学号,课程编号))

**连接**

select \* from 学生,系,课程,选修

where 学生.系编号=系.系编号 and 选修.课程编号=课程.课程编号 and 学生.学号=选修.学号

select 学生.学号,姓名,性别,出生日期,奖学金,系.系编号,系名,课程.课程编号,成绩 from 学生,系,课程,选修

where 学生.系编号=系.系编号 and 选修.课程编号=课程.课程编号 and 学生.学号=选修.学号

select 学生.学号,姓名,year(getdate())-year(出生日期) as 年龄,奖学金,系.系编号,系名,课程.课程编号,成绩 from 学生,系,课程,选修

where 学生.系编号=系.系编号 and 选修.课程编号=课程.课程编号 and 学生.学号=选修.学号

select \* from 学生,系,课程,选修

where 学生.系编号=系.系编号 and 选修.课程编号=课程.课程编号 and 学生.学号=选修.学号

select \* from 学生,系 where 系名='管理系'

select \* from 学生 where 政治面貌='团员'

select count(\*),sum(奖学金) from 学生 where 性别='男'

select count(\*),sum(奖学金),AVG(奖学金) from 学生 where 性别='男'

select 性别,count(\*),sum(奖学金),AVG(奖学金) from 学生 group by 性别

select 性别,count(\*),sum(奖学金),AVG(奖学金) from 学生 where 政治面貌='团员' group by 性别

select 系名,count(\*),sum(奖学金),AVG(奖学金) from 学生,系 where 学生.系编号=系.系编号 and 奖学金>800 group by 系名 having AVG(奖学金)>1300

select 系名,count(\*),sum(奖学金),AVG(奖学金) from 学生,系 where 政治面貌='团员' and 学生.系编号=系.系编号 group by 系名 having AVG(奖学金)>1000

select \* from 学生

**默认值和索引**

alter table 学生 add 民族 varchar(20) constraint def\_学生\_民族 default '汉族'

alter table 学生 add 政治面貌 char(4) not null default '团员'

alter table 学生 add 政治面貌 char(4) not null constraint def\_学生\_政治面貌 default '团员'

create index ix\_学生\_系编号\_出生日期 on 学生(系编号,出生日期)

**查询**

use 学生管理

select \* from 学生

select \* from 系

select \* from 选修

select \* from 课程

select left(学号,4)+'级' as 级别 from 学生

select cast(datediff(yy,出生日期,getdate()) as char(2))+'岁' as 年龄 from 学生

select \* from 学生 where (datediff(yy,出生日期,getdate()))>18

select \* into 学生04 from 学生 where abs(year(出生日期)-year(getdate()))>19

select \* into 学生03 from 学生 where 性别='男'

select distinct 系编号 from 学生

select distinct 课程编号 from 选修

select all 系编号 from 学生

select \* from 学生 where 奖学金!=1200

select \* from 学生 where 系编号 in('01','02','03')

select \* from 学生 where 奖学金 in(1200,1500)

select \* from 学生 where 奖学金 between 500 and 1800

select \* from 学生 where 姓名 like '王%'

select \* from 学生 where 姓名 like '王\_'

select 学号,姓名,性别 from 学生 where 性别='男'

select \* from 学生 where 姓名 like '%旭%'

select \* from 学生 where abs(year(出生日期)-year(getdate()))>19

select abs(datediff(yy,getdate(),出生日期)) as 年龄 from 学生

select year(getdate())-year(出生日期) as 年龄 from 学生

select substring('410426199501192521',7,4)+'年'

+substring('410426199501192521',11,2)+'月'

+substring('410426199501192521',13,2)+'日'

select left(学号,4) as 级段 from 学生

**考试题**

一、用命令创建“销售管理”数据库，并把创建数据库的代码保存到“First.TXT”文件中，具体要求如下表：（20分）

| 数据库名称：销售管理 | | |
| --- | --- | --- |
| 主要数据文件 | 逻辑文件名 | 销售\_Data1 |
| 物理文件名 | D:\销售管理\销售\_Data1.mdf |
| 文件组 | Primary |
| 初始大小 | 1MB |
| 文件大小上限 | 50MB |
| 每次增长大小 | 1MB |
| 次要数据文件 | 逻辑文件名 | 销售\_Data2 |
| 物理文件名 | D:\销售管理\销售\_Data2.ndf |
| 文件组 | 销售\_group |
| 初始大小 | 1MB |
| 文件大小上限 | 50MB |
| 每次增长大小 | 10% |
| 日志文件 | 逻辑文件名 | 销售\_Log |
| 物理文件名 | D:\销售管理\销售\_log.ldf |
| 初始大小 | 1MB |
| 文件大小上限 | 50MB |
| 每次增长大小 | 2MB |

二、用命令在“销售管理”数据库中创建以下表，并把创建表的代码保存到“Second.TXT”文件中：（20分）

1.产品（创建“产品号”为主键）（5分）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 小数  位数 | 是否允许  NULL值 | 说明 |
| 产品号 | CHAR | 5 |  | NO |  |
| 产品名 | VARCHAR | 20 |  | NO |  |
| 单位 | VARCHAR | 6 |  | YES | 默认值为：台 |
| 产地 | VARCHAR | 50 |  | YES |  |
| 等级 | Char | 1 |  | YES | A/B/C |

2.客户（创建“客户编号”为主键）（5分）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 小数  位数 | 是否允许  NULL值 | 说明 |
| 客户编号 | CHAR | 5 |  | NO |  |
| 客户名称 | VARCHAR | 50 |  | NO |  |
| 地址 | VARCHAR | 50 |  | YES |  |
| 联系电话 | VARCHAR | 12 |  | YES |  |
| 银行帐号 | VARCHAR | 30 |  | YES |  |

3.订购（创建“产品号，客户编号”为主键，“产品号”为外键，和“产品”表对应，“客户编号”为外键，和“客户”表对应）（10分）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 小数  位数 | 是否允许  NULL值 | 说明 |
| 产品号 | CHAR | 5 |  | NO |  |
| 客户编号 | CHAR | 5 |  | NO |  |
| 数量 | DECIMAL | 7 | 2 | YES |  |
| 单价 | Decimal（money） | 8(8) | 3(0) | YES | （范围1～10000） |
| 交货日期 | DATETIME | 8 |  | YES |  |

三、用命令在每个表中插入5条以上的记录，并把插入数据的代码保存到“Third.TXT”文件中：（15分，每个表5分）

四、用命令按以下要求对表进行修改，并把代码保存到“Fourth.TXT”文件中：（20分，每题4分）

1.为“产品”表添加一个字段“等级”，类型为固定长度1的字符型。

2.把“客户”表中的“联系电话”字段的长度设置为15。

3.为“产品”表的“等级”字段设置其取值范围为“A”、“B”和“C”。

4.把“客户”表中“联系电话”的字段名更改为“电话”。

5.在“客户”表中创建一个索引，索引字段为“客户名称”，索引名为“IX\_客户名称”。

五、利用上述的“客户”、“产品”和“订购”三个表进行如下查询操作，写出相应的查询语句，并保存到“Fifth.SQL”文件中。（25分，每题5分）

1.利用“产品”表查询所有产品的“产品编号”和“产品名”。

2.利用“客户”表查询“客户名称”的最后两个字是 “公司”的客户的所有信息。

3.利用“客户”、“产品”和“订购”表查询客户“唐僧”订购的产品的“产品名”。

4.利用“产品”和“订购”表查询已到“交货日期”的“产品名”。

5.利用“客户”和“订购”表查询每个客户的“客户编号，客户名称和订货金额”。