

第1章 数据库设计

我们学的 SQL 和 Oracle，之所以称之为关系型数据库，是因为数据库中的表是由关系的，呵呵，像是废话。

很多同学在学 SQL 语句时，觉得非常困难，那是因为你在学一个你根本不了解的数据库，数据库中的表不是你设计的，表与表之间的关系你不明白。因此在学 SQL 语句之前，先介绍一下数据库设计。

我已经忘记了数据库设计的几个范式的严格定义，但我会数据库设计。

下面举例说明数据库设计：

学校需要开发一个系统记录有学生、课程和成绩信息。数据库如何设计？

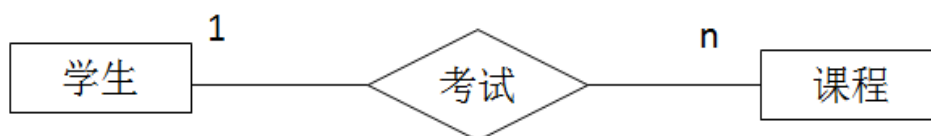
这里面涉及到两个实体，学生表、课程，这些表为实体表。

这些表之间有什么关系呢？教师给某个班级的学生上课，成绩记录在成绩表。

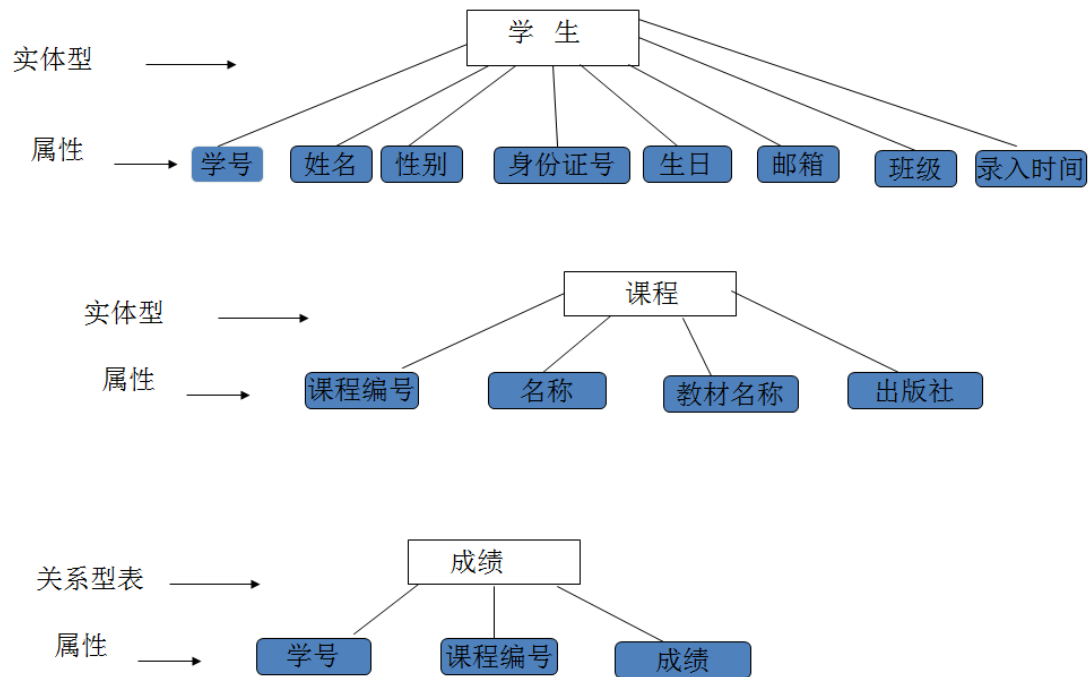
一个学生可以参加多门课程，关系是 1 对多。

1.1 数据库设计

E-R图的设计



设计表



1.2 准备本章学习环境

--第章准备学习用到的表和数据

--将以下语句拷贝到 SQL manager，分部执行

--创建数据库

```
create database schoolDB
```

```
go
```

```
use schoolDB
```

```
go
```

--创建学生表

```
create TABLE TStudent (
```

```
    StudentID varchar(10) NOT NULL,
```

```
    Sname varchar(10) DEFAULT NULL,
```

```
    sex char(2) DEFAULT NULL,
```

```
    cardID varchar(20) DEFAULT NULL,
```

```
    Birthday datetime DEFAULT NULL,

    Email varchar(40) DEFAULT NULL,

    Class varchar(20) DEFAULT NULL,

    enterTime datetime DEFAULT NULL

)

go

--创建课程表

create table TSubject

(

    subJectID nvarchar(4),

    subJectName nvarchar(30),

    BookName nvarchar(30),

    Publisher nvarchar(20)

)

go

--创建分数表

create table TScore

(

    StudentID nvarchar(10),

    subJectID nvarchar(4),

    mark decimal

)

go

--插入课程信息

insert into TSubject values ('0001','网络管理','奠基计算机网络','清华大学出版社')
```

```
'');  
insert into TSubject values ('0002','软件测试','功能测试','人邮出版社');  
insert into TSubject values ('0003','软件开发','企业级开发','人邮出版社')  
  
go
```

--创建函数该函数能够产生学生姓名

```
create function create_name(@s float)
```

```
RETURNS varchar(10)
```

```
begin
```

```
DECLARE @LN VARCHAR(300);
```

```
DECLARE @MN VARCHAR(200);
```

```
DECLARE @FN VARCHAR(200);
```

```
DECLARE @LN_N INT;
```

```
DECLARE @MN_N INT;
```

```
DECLARE @FN_N INT;
```

```
SET @LN='李王张刘陈杨黄赵周吴徐孙朱马胡郭林何高梁郑罗宋谢唐韩曹许邓萧冯曾程  
蔡彭潘袁于董余苏叶吕魏蒋田杜丁沈姜范江傅钟卢汪戴崔任陆廖姚方金邱夏谭韦贾邹  
石熊孟秦阎薛侯雷白龙段郝孔邵史毛常万顾赖武康贺严尹钱施牛洪龚';
```

```
SET @MN='伟刚勇春菊毅俊峰强军平保东文辉力明永健世广志瑗琰韵融园艺咏卿聪澜  
纯毓悦昭冰爽琬茗羽希宁欣飘育滢馥新利筠柔竹霭凝晓欢霄枫芸菲寒伊亚宜可姬舒义  
兴良海山仁波宁贵福生龙元全国胜学祥亮政谦亨奇固之岚苑富顺信子杰涛昌成康星光  
天达安岩中茂进林有坚和彪博诚先敬震振壮会思群豪清飞彬娜静淑惠珠翠雅芝妍茜秋  
珊莎锦黛青倩婷姣婉娴瑾颖露瑶怡婵雁蓓纨仪荷丹蓉眉君琴蕊薇菁梦素伟刚勇毅俊峰  
强军平保东文辉力明永健世广志义兴良海山仁波宁贵福生龙元全国胜学祥才发武新利  
清飞彬富顺信子杰涛昌成康星光天达安岩中茂进林有坚和彪博诚先敬震振壮会思群豪  
心邦承乐绍功松善厚庆磊民友裕河哲江超浩亮政谦亨奇固之翰朗伯宏言若鸣朋斌梁  
栋维启克伦翔旭鹏泽晨辰士以建家致树炎德行时泰盛雄琛钧冠策腾楠榕风航弘';
```

```
SET @FN='伟刚勇毅俊云莲真环雪荣爱妹霞香月莺媛艳瑞凡佳嘉琬勤珍贞莉桂娣叶璧才  
发武丽琳轮翰朗伯宏言若鸣朋斌梁栋维启克伦翔旭鹏泽晨辰士以建家致树炎德河哲江  
超浩璐娅琦晶裕华慧巧美婕馨影荔枝思心邦承乐绍功松善厚庆磊民友玉萍红娥玲芬芳  
燕彩兰凤洁梅秀娟英行时泰盛雄琛钧冠策腾楠榕风航弘峰强军平保东文辉力明永健世  
广志义兴良海山仁波宁贵福生龙元全国胜学祥才发武新利清飞彬富顺信子杰涛昌成康
```

星光天达安岩中茂进林有坚和彪博诚先敬震振壮会思群豪心邦承乐绍功松善厚庆磊民
友裕河哲江超浩亮政谦亨奇固之轮翰朗伯宏言若鸣朋斌梁栋维启克伦翔旭鹏泽晨辰士
以建家致树炎德行时泰盛雄琛钧冠策腾楠榕风航弘';

```
SET @LN_N=len(@LN)*@s;
```

```
SET @MN_N=len(@MN)*@s;
```

```
SET @FN_N=len(@FN)*@s;
```

```
return substring(@LN,@LN_N,1)+substring(@MN,@MN_N,1)+substring(@FN,@FN_N,1);
```

```
end
```

```
go
```

```
--测试是否能产生新的姓名
```

```
select dbo.create_name(rand())
```

```
go
```

```
--创建汉字转拼音的函数，用来产生用户的邮箱
```

```
create function dbo.fn_GetPinyin(@words nvarchar(2000))
```

```
returns varchar(8000)
```

```
as
```

```
begin
```

```
declare @word nchar(1)
```

```
declare @pinyin varchar(8000)
```

```
declare @i int
```

```
declare @words_len int
```

```
declare @unicode int
```

```
set @i = 1
```

```
set @words = ltrim(rtrim(@words))
```

```
set @words_len = len(@words)
```

```
while (@i <= @words_len) --循环取字符

begin

set @word = substring(@words, @i, 1)

set @unicode = unicode(@word)

set @pinyin = ISNULL(@pinyin +SPACE(0),'')+

(case when unicode(@word) between 19968 and 19968+20901 then

(select top 1 py from (

select 'a' as py,N'厖' as word

union all select 'ai',N'霰'

union all select 'an',N'黯'

union all select 'ang',N'醯'

union all select 'ao',N'鵠'

union all select 'ba',N'霸'

union all select 'bai',N'陌' --鞞鞞陌

union all select 'ban',N'瓣'

union all select 'bang',N'鎗'

union all select 'bao',N'鏹'

union all select 'bei',N'璧'

union all select 'ben',N'輻'

union all select 'beng',N'鏹'

union all select 'bi',N'鼃'

union all select 'bian',N'變'

union all select 'biao',N'鰲'

union all select 'bie',N'譬'

union all select 'bin',N'鬢'
```

union all select 'bing',N'靐'

union all select 'bo',N'蔔'

union all select 'bu',N'簿'

union all select 'ca',N'囍'

union all select 'cai',N'盜' --縹盜

union all select 'can',N'爇'

union all select 'cang',N'贍'

union all select 'cao',N'鏹'

union all select 'ce',N'箒'

union all select 'cen',N'岑'

union all select 'ceng',N'甞' --砵砵岵猋甞

union all select 'cha',N'詫'

union all select 'chai',N'囁'

union all select 'chan',N'顫'

union all select 'chang',N'輶'

union all select 'chao',N'舳'

union all select 'che',N'爇'

union all select 'chen',N'讖'

union all select 'cheng',N'秤'

union all select 'chi',N'驚'

union all select 'chong',N'銃'

union all select 'chou',N'殍'

union all select 'chu',N'矗'

union all select 'chuai',N'踹'

union all select 'chuan',N'鶡'

union all select 'chuang',N'愴'

union all select 'chui',N'頤'

union all select 'chun',N'蠢'

union all select 'chuo',N'綏'

union all select 'ci',N'髡' --賜髡

union all select 'cong',N'聰'

union all select 'cou',N'輶'

union all select 'cu',N'顱'

union all select 'cuan',N'爨'

union all select 'cui',N'睪'

union all select 'cun',N'村'

union all select 'cuo',N'錯'

union all select 'da',N'撻'

union all select 'dai',N'隸'

union all select 'dan',N'餐'

union all select 'dang',N'閭'

union all select 'dao',N'燾'

union all select 'de',N'的'

union all select 'den',N'鈍'

union all select 'deng',N'鐙'

union all select 'di',N'蟬'

union all select 'dia',N'嗲'

union all select 'dian',N'驪'

union all select 'diao',N'鐃'

union all select 'die',N'黠' --咥黠

union all select 'ding',N'顛'

union all select 'diu',N'鈺'

union all select 'dong',N'霽'

union all select 'dou',N'鬪'

union all select 'du',N'蠹'

union all select 'duan',N'𪛗' -- 𪛗 𪛗

union all select 'dui',N'藹'

union all select 'dun',N'踰'

union all select 'duo',N'鷄'

union all select 'e',N'鰓'

union all select 'en',N'摠'

union all select 'eng',N'鞞'

union all select 'er',N'榘'

union all select 'fa',N'髮'

union all select 'fan',N'纂'

union all select 'fang',N'放'

union all select 'fei',N'贐'

union all select 'fen',N'鱻'

union all select 'feng',N'勑'

union all select 'fo',N'拂'

union all select 'fou',N'鴆'

union all select 'fu',N'獼' -- 獼 獼

union all select 'ga',N'尅'

union all select 'gai',N'璫'

union all select 'gan',N'瀨'

union all select 'gang',N'戇'

union all select 'gao',N'鎔'

union all select 'ge',N'獫'

union all select 'gei',N'給'

union all select 'gen',N'拒'

union all select 'geng',N'垠' --互垠厖噤咽

union all select 'gong',N'羴' --煩顛尠羴

union all select 'gou',N'購'

union all select 'gu',N'顧'

union all select 'gua',N'詿'

union all select 'guai',N'恠'

union all select 'guan',N'鑰'

union all select 'guang',N'擴'

union all select 'gui',N'鯢'

union all select 'gun',N'覬'

union all select 'guo',N'腠'

union all select 'ha',N'哈'

union all select 'hai',N'饕'

union all select 'han',N'鞞'

union all select 'hang',N'沆'

union all select 'hao',N'尅'

union all select 'he',N'霫'

union all select 'hei',N'嬾'

union all select 'hen',N'恨'

union all select 'heng',N'堊' --堊囍

union all select 'hong',N'閏'

union all select 'hou',N'黉'

union all select 'hu',N'驢'

union all select 'hua',N'薤'

union all select 'huai',N'瓊'

union all select 'huan',N'鰓'

union all select 'huang',N'銑'

union all select 'hui',N'顙'

union all select 'hun',N'譚'

union all select 'huo',N'忝'

union all select 'ji',N'驥'

union all select 'jia',N'𦍋'

union all select 'jian',N'鑣'

union all select 'jiang',N'譽'

union all select 'jiao',N'醕'

union all select 'jie',N'緝'

union all select 'jin',N'𪔐'

union all select 'jing',N'競'

union all select 'jiong',N'藟'

union all select 'jiu',N'櫓'

union all select 'ju',N'燿'

union all select 'juan',N'絹'

union all select 'jue',N'鑊'

union all select 'jun',N'攢'

union all select 'ka',N'鈇'

union all select 'kai',N'𪔐' --𪔐

union all select 'kan',N'𪔑'

union all select 'kang',N'𪔒'

union all select 'kao',N'𪔓'

union all select 'ke',N'𪔔'

union all select 'ken',N'𪔕'

union all select 'keng',N'𪔖' --𪔖

union all select 'kong',N'𪔗'

union all select 'kou',N'𪔘'

union all select 'ku',N'𪔙'

union all select 'kua',N'𪔚'

union all select 'kuai',N'𪔛'

union all select 'kuan',N'𪔜'

union all select 'kuang',N'𪔝'

union all select 'kui',N'𪔞'

union all select 'kun',N'𪔟'

union all select 'kuo',N'𪔠'

union all select 'la',N'𪔡'

union all select 'lai',N'𪔢'

union all select 'lan',N'𪔣'

union all select 'lang',N'𪔤'

union all select 'lao',N'𪔥'

union all select 'le',N'𪔦'

union all select 'lei',N'𪔧' --𪔧

union all select 'leng',N'𪔨'

union all select 'li',N'璿'

union all select 'lia',N'倆'

union all select 'lian',N'戀'

union all select 'liang',N'鯨'

union all select 'liao',N'瞭'

union all select 'lie',N'鱉'

union all select 'lin',N'轡' --轡拎

union all select 'ling',N'伶'

union all select 'liu',N'咯' --硌硌嚙咯

union all select 'long',N'隴'

union all select 'lou',N'鏤'

union all select 'lu',N'魯'

union all select 'lv',N'鑢'

union all select 'luan',N'亂'

union all select 'lue',N'撾'

union all select 'lun',N'論'

union all select 'luo',N'鑠'

union all select 'ma',N'嘛'

union all select 'mai',N'霰'

union all select 'man',N'縵'

union all select 'mang',N'蟒'

union all select 'mao',N'毳'

union all select 'me',N'痲' --痲攸

union all select 'mei',N'欸'

union all select 'men',N'們'

union all select 'meng',N'蒙' -- 蒙跖

union all select 'mi',N'冪'

union all select 'mian',N'麵'

union all select 'miao',N'廟'

union all select 'mie',N'𩶛' -- 𩶛𩶛

union all select 'min',N'𩶛'

union all select 'ming',N'詔'

union all select 'miu',N'謬'

union all select 'mo',N'𩶛' -- 𩶛𩶛

union all select 'mou',N'𩶛' -- 𩶛𩶛

union all select 'mu',N'旆'

union all select 'na',N'𩶛'

union all select 'nai',N'𩶛'

union all select 'nan',N'𩶛'

union all select 'nang',N'𩶛'

union all select 'nao',N'𩶛'

union all select 'ne',N'呢'

union all select 'nei',N'𩶛' -- 𩶛𩶛

union all select 'nen',N'𩶛'

union all select 'neng',N'能' -- 𩶛𩶛𩶛𩶛

union all select 'ni',N'𩶛'

union all select 'nian',N'𩶛'

union all select 'niang',N'𩶛'

union all select 'niao',N'𩶛'

union all select 'nie',N'𩶛'

union all select 'nin',N'拑'

union all select 'ning',N'寧'

union all select 'niu',N'紐'

union all select 'nong',N'農'

union all select 'nou',N'譎'

union all select 'nu',N'擣'

union all select 'nv',N'衄'

union all select 'nue',N'瘡'

union all select 'nuan',N'煨' --磳磨煨煨

union all select 'nuo',N'桼'

union all select 'o',N'鞫' --撝毚𠵹鞫

union all select 'ou',N'漚'

union all select 'pa',N'裱'

union all select 'pai',N'磚' --鉈磚

union all select 'pan',N'鑠'

union all select 'pang',N'胖'

union all select 'pao',N'礲'

union all select 'pei',N'轡'

union all select 'pen',N'噉'

union all select 'peng',N'毘' --洝毘闐芑毘

union all select 'pi',N'鷗'

union all select 'pian',N'騙'

union all select 'piao',N'標'

union all select 'pie',N'嫫'

union all select 'pin',N'聘'

union all select 'ping',N'蘋'

union all select 'po',N'魄'

union all select 'pou',N'𪔐' -- 𪔐

union all select 'pu',N'曝'

union all select 'qi',N'螫'

union all select 'qia',N'髀'

union all select 'qian',N'繹'

union all select 'qiang',N'𪔐' -- 𪔐

union all select 'qiao',N'𪔐'

union all select 'qie',N'𪔐'

union all select 'qin',N'蕤'

union all select 'qing',N'櫟'

union all select 'qiong',N'璫'

union all select 'qiu',N'糗'

union all select 'qu',N'覷'

union all select 'quan',N'勸'

union all select 'que',N'確'

union all select 'qun',N'𪔐'

union all select 'ran',N'𪔐'

union all select 'rang',N'讓'

union all select 'rao',N'繞'

union all select 're',N'熱'

union all select 'ren',N'飪'

union all select 'reng',N'𪔐'

union all select 'ri',N'𪔐'

union all select 'rong',N'榕'

union all select 'rou',N'嶠'

union all select 'ru',N'孺'

union all select 'ruan',N'孺'

union all select 'rui',N'睿'

union all select 'run',N'欄' --欄授

union all select 'ruo',N'鵠'

union all select 'sa',N'柂' --檣柂

union all select 'sai',N'籩' --籩籩

union all select 'san',N'閭'

union all select 'sang',N'喪'

union all select 'sao',N'皐'

union all select 'se',N'颶' --衺聾

union all select 'sen',N'簪'

union all select 'seng',N'綰' --閭綰

union all select 'sha',N'霎'

union all select 'shai',N'曬'

union all select 'shan',N'鱔'

union all select 'shang',N'綯'

union all select 'shao',N'梢'

union all select 'she',N'櫛'

union all select 'shen',N'疹'

union all select 'sheng',N'賸'

union all select 'shi',N'𪔐' --鯨𪔐𪔐𪔐

union all select 'shou',N'鍬'

union all select 'shu',N'黠'

union all select 'shua',N'諛'

union all select 'shuai',N'擧'

union all select 'shuan',N'喘'

union all select 'shuang',N'瀟'

union all select 'shui',N'睡'

union all select 'shun',N'髡'

union all select 'shuo',N'鑠'

union all select 'si',N'𪚩' -- 𪚩 𪚩 𪚩

union all select 'song',N'鋤'

union all select 'sou',N'瘡'

union all select 'su',N'鷗'

union all select 'suan',N'算'

union all select 'sui',N'鑿'

union all select 'sun',N'𪚩'

union all select 'suo',N'蟀'

union all select 'ta',N'𪚩' -- 𪚩 𪚩

union all select 'tai',N'燭'

union all select 'tan',N'𪚩'

union all select 'tang',N'燙'

union all select 'tao',N'𪚩' -- 𪚩 𪚩

union all select 'te',N'𪚩'

union all select 'teng',N'𪚩' -- 𪚩 𪚩

union all select 'ti',N'𪚩'

union all select 'tian',N'𪚩'

union all select 'tiao',N'耀'

union all select 'tie',N'餐'

union all select 'ting',N'烹' --滷烹

union all select 'tong',N'勳'

union all select 'tou',N'透'

union all select 'tu',N'鵠'

union all select 'tuan',N'緣'

union all select 'tui',N'駢'

union all select 'tun',N'埤'

union all select 'tuo',N'籜'

union all select 'wa',N'轆'

union all select 'wai',N'頹'

union all select 'wan',N'謁'

union all select 'wang',N'望'

union all select 'wei',N'衛'

union all select 'wen',N'豐'

union all select 'weng',N'鶯'

union all select 'wo',N'齧'

union all select 'wu',N'鵠'

union all select 'xi',N'晝'

union all select 'xia',N'鋤'

union all select 'xian',N'鱗'

union all select 'xiang',N'鯨'

union all select 'xiao',N'敦'

union all select 'xie',N'蹊'

union all select 'xin',N'𣎵'

union all select 'xing',N'𣎵'

union all select 'xiong',N'𣎵'

union all select 'xiu',N'𣎵'

union all select 'xu',N'𣎵'

union all select 'xuan',N'𣎵'

union all select 'xue',N'𣎵'

union all select 'xun',N'𣎵'

union all select 'ya',N'𣎵'

union all select 'yan',N'𣎵'

union all select 'yang',N'𣎵'

union all select 'yao',N'𣎵'

union all select 'ye',N'𣎵' -- 𣎵

union all select 'yi',N'𣎵'

union all select 'yin',N'𣎵'

union all select 'ying',N'𣎵'

union all select 'yo',N'𣎵'

union all select 'yong',N'𣎵'

union all select 'you',N'𣎵'

union all select 'yu',N'𣎵'

union all select 'yuan',N'𣎵'

union all select 'yue',N'𣎵'

union all select 'yun',N'𣎵'

union all select 'za',N'𣎵'

union all select 'zai',N'𣎵'

union all select 'zan',N'贛'

union all select 'zang',N'臧'

union all select 'zao',N'竈'

union all select 'ze',N'稜'

union all select 'zei',N'鰓'

union all select 'zen',N'噉'

union all select 'zeng',N'贈'

union all select 'zha',N'醺'

union all select 'zhai',N'瘵'

union all select 'zhan',N'躡'

union all select 'zhang',N'障'

union all select 'zhao',N'繹'

union all select 'zhe',N'鷗'

union all select 'zhen',N'黠'

union all select 'zheng',N'證'

union all select 'zhi',N'豈'

union all select 'zhong',N'鍾'

union all select 'zhou',N'驟'

union all select 'zhu',N'鑄'

union all select 'zhua',N'爪'

union all select 'zhuai',N'跬'

union all select 'zhuan',N'簞'

union all select 'zhuang',N'戇'

union all select 'zhui',N'鐫'

union all select 'zhun',N'稕'

```
union all select 'zhuo',N'籀'

union all select 'zi',N'漬' --漬咽

union all select 'zong',N'縱'

union all select 'zou',N'嫫'

union all select 'zu',N'誼'

union all select 'zuan',N'攢'

union all select 'zui',N'檮'

union all select 'zun',N'鎔'

union all select 'zuo',N'咗') t

where word >= @word collate Chinese_PRC_CS_AS_KS_WS

order by word collate Chinese_PRC_CS_AS_KS_WS ASC) else @word end)

set @i = @i + 1

end

return @pinyin

END

go

--执行以下命令查看函数效果

select dbo.fn_GetPinyin('韩立刚')

go

--创建添加学生的存储过程

create procedure addStudent @num int

as

begin

declare @i int;
```

```
set @i=1;

delete TStudent;

while @num>=@i

begin

declare @sname varchar(40)

declare @sub int

set @sname=dbo.create_name(rand())

set @sub=rand()*4

insert TStudent (StudentID,Sname,sex,cardID,Birthday,Email,Class,enterTime)values (

right(replicate('0',10)+ltrim(@i),10),

@sname,

case when(rand())>0.5) then '男' else '女' end,

convert(nvarchar(16),convert(bigint,rand()*10000000000000000)),

'198'+convert(char(1),convert(int,rand()*9)+1)+'-'+convert(char(2),convert(int,rand()*11+1))

+'-'+convert(char(2),convert(int,rand()*27+1)),

dbo.fn_GetPinyin(@sname)+'@91xueit.com',

case when (@sub<1) then '网络班' when (@sub>=1 and @sub<2) then '测试班' else

'开发班' end,

getdate()

)

set @i=@i+1;

end

end
```

--添加个学生

exec dbo.addStudent 1900

--查看添加的学生

select * from dbo.TStudent

go

--创建插入学生成绩的存储过程学生成绩在 50-100 分之间

create procedure fillScore

as

DECLARE @St_Num INT;

DECLARE @Sb_Num INT;

DECLARE @i1 INT;

DECLARE @i2 INT;

set @i1=1;

set @i2=1;

delete TScore;

select @St_Num=count(*) from TStudent;

select @Sb_Num=count(*) from TSubject;

while @St_Num>=@i1

begin

set @i2=1;

while @Sb_Num>=@i2


```
begin

insert TScore values

(right(replicate('0',10)+ltrim(@i1),10),right(replicate('0',4)+ltrim(@i2),4),50+rand()*50)

set @i2=@i2+1

end

set @i1=@i1+1

end

--插入成绩

--调用存储过程插入学生成绩

exec fillSore

select * from dbo.TScore

--查看三张表连接的结果

select a.*,b.*,c.* from TStudent a join TScore b on a.StudentID=b.StudentID join TSubject c
on b.subjectID=c.subjectID
```

第2章 Transact-SQL 概述

概述 Structured Query Language

2.1 Transact-SQL 编程语言

美国国家标准协会（ANSI）和国际标准组织（ISO）为 SQL 定义了标准，微软通过用 Transact-SQL 和 ANSI—SQL 兼容，Transact-SQL 还包含了几种能够增强性能的扩展。

2.1.1 Ttansact-SQL 语句的类型

- 数据定义语言（Date Definition Language）语句

DDL 语句通过生成数据库、表、和用户自定义的数据库

create object_name

alter object_name

drop object_name

use schoolDB

create table teacher

(cust_id int ,company varchar(40),contact varchar(30),phone char(12))

go

alter table teacher add age int default 30

- 数据控制语言（Date Control Language）语句

用于改变某个用户或角色相关联的权限

grant

deny

revoke

use schoolDB

grant select on products to public

go

- 数据操作语言（Date Manipulation Language）语句

操作数据库中的数据。你可以更改数据库中的数据或查询数据库中的信息。

Select

Insert

Delecte

Update

2.1.2 Transact-SQL 语法要素

➤ 批处理

go

一个批处理命令通知 SQLServer 分析并运行一个批处理的所有指令

实际上不是 Transact-SQL 语句，只是描述一个批处理。局部变量作用范围局限在一个批内，必须独占一行。

Exec

用于执行用户自定义的函数，系统存储过程，用户自定义的存储过程或扩展存储过程

➤ 注释语句

作为注释或在测试时是一些语句无效

嵌于行内的注释语句

USE schoolDB

go

select * from dbo.TScore--从那张表查找

where mark+9<60

GO

2.2 GO

块注释语句

/*

```
/*
```

给学生提 5 分

看看还有多少不及格

```
*/
```

```
USE schoolDB
```

```
go
```

```
select * from dbo.TScore where mark+9<60
```

```
GO
```

2.3 标识符

标准标识符

第一个字符是 a—z 或 A—Z

第一个字符后可以是数字，字母，或各种符号@,\$, _

第一个字母是符号时表示有特殊的用途

@代表局部变量或参数

#代表临时表和存储过程

##代表一个全局临时表

限定表示符

当对象名包含空格时,当用保留关键字被用作对象的名字时，必须使用括号和引号把限定标识符括起来

```
Use schoolDB
```

```
create table [order detail]
```

```
(
```

```
OrderID nvarchar(10),
```

```
OrderTime datetime
```

)

```
select * from [order detail]
```

打开双引号界定符

```
set quoted_identifier on
```

```
set quoted_identifier off
```

```
select * from "order detail"
```

标识符的命名原则

短

有意义

使用简单明了的命名习惯

能够区分对象的类型的标识符 view table

保持对象名和用户名的唯一性 sales 用户 sales 表

2.4 数据类型

数字型 代表数字 int tinyint smallint bigint 十进制小数 money smallmoney
decimal 浮点数和 real

日期型 datetime 可以精确到 0.333 毫秒 small

字符型 包括 char 和 nchar 也包含变长字符类型 varchar 和 nvarchar

定长字符 char(20)

变长字符 varchar(20)

Char 适合存放英文 一个字符占用 1 个字节

Nchar 适合存放中文 一个字符占用 2 个字节

二进制型 Binary 和 varbinary, bit 代表一位的值 0 或 1, rowversion 代表数据库中唯一的 8 位二进制。

唯一标识 代表一个全局特殊标识符 (GUID), 是一个 16 位 16 进制的值。

SQL 变量 包括 SQL 所支持的各种数据类型, 但不包括 text, ntext, rowversion 和 sql_variant 这几种数据类

位图和文本（image&text） 属于大型二进制对象结构

表 这种数据类型代表一个表结构，将一个表保存在一个字段中

游标 这种数据类型用于存储过程的编程

用户自定义数据类型 由数据库管理员生成，它基于系统数据类型，当多个表需要在一个字段存储同一类型数据时，这些字段具有相同的数据类型、长度和可控属性时，选择实用用户自定义的数据类型。

2.5 变量

使用 **Declare** 定义局部变量，@局部变量作用域仅限于一个批处理中，@@而全局变量在整个会话有效

```
use schoolDB
```

```
go
```

```
declare @sname nvarchar(11),@studentid nvarchar(20)
```

```
set @studentid ='0000000022'
```

```
select @sname=Sname from dbo.TStudent where StudentID=@studentid
```

```
select @sname as 姓名
```

2.6 系统函数

聚集函数---对于一个集合中的值进行运算，返回一个单一的，汇总的值。

```
use northwind
```

```
select avg(unitprice) as AvgPrice from products
```

```
go
```

标量函数---只对单个值进行运算并返回单一的值，只要表达式正确，就可以使用这些函数

```
use northwind
```

```
select DB_name() as 'database'
```

```
go
```

2.6.1 系统函数范例

1. 把时间转化成其他各式

```
select 'ansi:',convert(varchar(30),getdate(),102) as style

union

select 'Japanese',convert(varchar(30),getdate(),111)

union

select 'European',convert(varchar(30),getdate(),113)

go
```

2. 算数运算符

+ - * / %

比较运算符 = < > >=

字符串运算符 + 空字符不等于空值

逻辑运算符 and or not

3. 表达式

表达式是各种符号和对单个数据进行操作

```
select mark+5 from dbo.TScore where mark<60
```

4. 时间函数

select dateadd(yy,4,(getdate())) 表示对当时时间加 4 年

select dateadd(qq,2,(getdate())) 季度

select dateadd(mm,2,(getdate())) 月

select dateadd(dd,2,(getdate())) 日

select dateadd(wk,2,(getdate())) 周

select dateadd(hh,2,(getdate())) 小时

select dateadd(mi,2,(getdate())) 分钟

select dateadd(ss,2,(getdate())) 秒

select dateadd(dy,3,getdate()) 一年中的第几天

select datediff(day,'2005-3-4',getdate()) 返回时间差

select datepart(day,getdate())

5. 流控制语言

语句集

Begin....end

If....else

while

使用 if else 案例

declare @score decimal

set @score=rand()*100

if (@score>60) print '****考试及格****'

else print '****考试不及格****'

case 命令的使用

declare @score int

set @score=(50+rand()*50)/10

select case @score when 5 then '***不及格***'

when 6 then '***及格***'

when 7 then '***良好***'

else '***优秀***'

end

在查询语句中的应用

```
select StudentID, subJectID, case when mark>60 then '及格' else '不及格' end from dbo.TScore
```

使用 while 的案例

```
declare @times int,@string nvarchar(1000)
```

```
set @times=1
```

```
set @string='*'
```

```
while (@times<100)
```

```
begin
```

```
print @string;
```

```
set @times=@times+1;
```

```
set @string=@string+'*';
```

```
end
```

6. 保留关键字----在 T-SQL 语句的任何地方都不能使用已经定义的保留关键字

```
create table order
```

```
(
```

```
orderid int,
```

```
ordertime datetime
```

```
)
```

2.7 试验 A 使用 SQL Server 联机帮助

附：字符串函数

下面是日期型转化成变长字符串，再将变长字符串转化成日期型

```
declare @ch varchar(19)
```

```
declare @t datetime
```

```
set @t=getdate()
```

```
select @t
```

```
set
```

```
@ch=convert(varchar(19),year(@t))+ '-' +convert(varchar(19),month(@t))+ '-' +convert(varchar(19),day(@t))
```

```
select @ch
```

```
select convert(datetime,@ch)
```

使用 Waitfor delay or Waitfor time 知道指定的时间

```
begin tran
```

```
update authors set au_lname=au_lname
```

```
waitfor delay '00:00:10'
```

```
update titles set title=title
```

```
rollback tran
```

```
begin tran
```

```
update titles set title=title
```

```
update authors set au_lname=au_lname
```

```
rollback tran
```

第3章 使用 Transact-SQL 的查询工具

3.1 SQL 查询分析器

一个使用方便的文本编辑器

代码颜色转换

带有网格或文本输出的多重查询窗口

执行脚本一部分

查询执行的信息

3.1.1 使用 SQL 查询分析器中的对象浏览工具

对象浏览器有如下功能

为对象生成脚本---生成创建对象的脚本，`select insert update`

执行存储过程---执行存储过程，当执行有参数的存储过程时，对象浏览器提示你输入参数的值

打开表---显示查询结果，可以编辑、插入或删除行

改变数据库中的对象

使用 Transact-SQL 模板

生成数据库，表，视图，索引，存储过程，统计数字和函数

管理扩展属性、连接的服务器、登陆帐号、角色和用户

声明及使用游标

定制脚本

执行 Transact-SQL 语句

3.1.2 执行 Transact-SQL 语句

在运行时动态构造并执行 Transact-SQL 语句

使用批组织一起要执行的语句

使用脚本文件保存批语句，以供日后使用

```
declare @dbname varchar(30),@tblname varchar(30)
```

```
set @dbname='northwind'
```

```
set @tblname='products'
```

```
execute
```

```
('use '+@dbname+' select * from '+@tblname)
```

```
Go
```

使用批

提交一个语句或同时提交多个语句，只有在 SQL 查询分析器或 osql 工具接受这个语句，在 ODBC 或 ODBC API 的应用程序使用 go 将会出现错误。SQL Server 优化、编译并执行批中的语句，用户自定义的变量有效范围局限于一个批内部。

对象生成语句必须使用单独的批，批中不能同时出现的语句

Create procedure

Create view

Create trigger

Create rule as

Create default

使用脚本

使用 SQL 查询分析器或写字板之类的任何其它文本编辑器，编写并保存脚本，扩展名.sql

1. 最佳实践

使用 SQL 查询分析器可以建立和 SQL Server 的多个连接，可为对象生成脚本的功能

使用对象浏览工具来查找表和字段名，或为他们生成脚本

使用模板生成对象

运行批文件或执行重复性任务时，使用 osql

把经常使用的 Transact-SQL 语句保存在文件中，可以以后使用。

2. 试验 A 创建并执行 Transact—SQL 脚本

第4章 数据查询

4.1 使用 select 语句查询数据

4.1.1 指定列

查询所有行

```
select * from dbo.TStudent
```

使用 where 子句指定行

```
select Sname,sex,Email from dbo.TStudent where Sname='田育朋'
```

4.1.2 过滤数据

使用比较操作符 = > < >= <= <>

```
select * from dbo.TScore where mark<=60
```

使用字符比较符 like

% 0 个或多个字符串

_ 任何单个的字符

[]在指定区域或集合内的任何单个字符

[^]不在指定区域或集合内的任何单个字符

```
select * from dbo.TStudent where sname like '高%'
```

```
select * from dbo.TStudent where sname like '_[明,育]_'
```

```
select * from dbo.TStudent where sname like '_[^明,育]_'
```

4.1.3 使用逻辑操作符

Or and not

```
select * from dbo.TStudent where Sname like '高%' and sex='男' or  
StudentID='0000000112'
```

查找不姓高的学生

```
select * from dbo.TStudent where Sname not like '高%'
```

4.1.4 查询在一定范围内的值

Between 70 and 80 包括 70 和 80

```
select * from dbo.TScore where mark between 70 and 80
```

等价于

```
select * from dbo.TScore where mark>=70 and mark<=80
```

go

```
mark>70 and mark<80
```

不包括 70 和 80

尽量使用 **between** 而不使用 **and** 和比较操作符表示的表达式

如果想返回不在指定区域的行时，使用 **not between** 。这样会降低数据查询的速度。

```
select * from dbo.TScore where mark not between 70 and 80
```

指定时间范围

```
select * from dbo.TStudent where Birthday between '1983-01-01' and  
'1984-01-01'
```

4.1.5 使用值的列表作为搜索条件

```
select * from dbo.TStudent where Class in ('开发','网络')
```

等价于

```
select * from dbo.TStudent where Class='开发' or Class='网络'
```

使用 **in** 或使用由 **or** 操作符连接起来的一系列比较操作符，**SQLServer** 在处理他们的方式相同。不要在搜索条件结尾使用 **NULL**，这将返回意想不到的结果。可以使用 **not in**，降低数据查询速度。

4.1.6 查询未知的值 空值

```
insert dbo.TStudent (StudentID,Sname,sex) values ('0000001901','韩立刚',  
'男')
```

查找班级为空的学生

```
select * from dbo.TStudent where Class is not null
```

查找班级不为空的学生

```
select * from dbo.TStudent where Class is not null
```

```
is null
```

使用 **is not null** 来查询指定列中非空的行

4.2 格式化结果集

4.2.1 给数据排序

```
select StudentID, subJectID, mark from dbo.TScore order by 2, 3 desc  
select StudentID, subJectID, mark from dbo.TScore order by subJectID, mark  
desc
```

asc 升序

desc 降序

4.2.2 消除重复的行

distinct

```
select distinct Class from dbo.TStudent
```

4.2.3 改变字段的名字

```
select StudentID as '学号', Sname as '姓名', sex as '性别', cardID as '身份证号',  
Birthday as '生日', Email as '邮件', Class as '专业', enterTime as '录入时间' from dbo.Tstudent
```

等价于

```
select StudentID '学号', Sname '姓名', sex '性别', cardID '身份证号',  
Birthday '生日', Email '邮件', Class '专业', enterTime '录入时间' from  
dbo.TStudent
```

4.2.4 使用符号

符号可能是字母，数字或标识，在结果集中，他们被用作特定的值，以增加结果集的可读性。

```
select StudentID '学号', Sname '姓名', '性别', sex '性别' from dbo.Tstudent
```

4.2.5 计算列

年龄是计算列

```
select StudentID as '学号', Sname as '姓名', sex as '性别', cardID as '身份证号',  
Birthday as '生日', Email as '邮件', Class as '专业', enterTime as '录入时间',  
Datediff(yy, Birthday, getdate()) as '年龄' from dbo.TStudent
```

4.3 查询的工作原理

非缓存的查询

所有查询在执行前，都需要经过一下步骤：解释、解析、优化和编译

解析----检查语句的语法是否正确

解析----校验语句中出现的对象名称是否有效，同时检查对象的所有权的权限

优化----检查是否能够使用索引并决定联合（join）策略

编译----把查询翻译为一个可执行的表（from）

执行----把编译过程的查询要求提交并进行处理

缓存的查询

为了提高性能 SQLServer 能够保存编译过的查询计划供以后使用，查询计划是经过优化的指令，他指定了如何处理插叙并访问数据。

4.3.1 查询是如何被自动缓存的

被缓存的查询保存在内存中一个叫做过程缓存的地方，在下列两种情况下，查询将被自动缓存—特定的批和自动参数化。

特殊的批-----SQL Server 将为特殊的批保存查询计划，如果接下来的批和上一个批类似，SQLServer 将使用缓存的计划。脚本文本必须匹配。

```
select * from products where unitprice = $ 12.5
```

```
select * from products where unitprice = 12.5
```

```
select * from products where unitprice = $ 13.5
```

```
go
```

查询语句 1 和 3 可以使用一个查询计划，但语句 2 必须使用另外的查询计划。

自动参数化

```
use library
```

```
select * from member where member_no=7890
```

```
select * from member where member_no=1234
```

```
select * from member where member_no=7890
```


go

4.3.2 影响性能的注意事项

尽量使用正逻辑而不是非逻辑，非逻辑操作（no between 、 not in 和 not null）可能会降低查询速度，因为它要检索数据表中的所有行；

如果能够使用一个更确定的查询，就尽量避免使用关键字 LIKE，使用 LIKE 查询，数据查询速度可能会降低；

只要有可能，尽量在搜索条件中使用精确的比较或值的域；

使用 Order by 子句可能会降低数据查询速度，

7.试验 A 查询数据并操作结果集

计算数据，然后返回计算出的值，并使用字段别名

通过使用字符串函数对结果集进行格式化

拆分姓名为两列

```
select left(Sname,1) '姓',right(Sname,2) '名' from dbo.TStudentgo
```

使用系统函数

如何得到服务器的进程 ID

sp_who 显示服务器上所有正在产生的活动

select @@Spid 如果想知道那些活动是你的

exec sp_who 56 将显示和你的服务器进程号有关的所有活动

查询运行环境信息

```
select user_name()
```

```
select DB_name()
```

```
select @@servername
```

第5章 多表连接查询

创建多表查询的学习环境

学生表

学号	姓名	性别
1	韩立刚	男
2	王景正	男
3	郭淑丽	女
4	韩旭	女
5	孟小飞	男

成绩表

学号	学科	成绩
1	英语	89
1	数学	98
2	英语	87
2	数学	78
3	英语	97
3	数学	86
6	英语	90
6	数学	93

```
--创建学生表 student
Create table student
(
    studentid int,
    sname nvarchar(10),
    sex nchar(1)
)
--插入学生
insert student values (1, '韩立刚', '男')
insert student values (2, '王景正', '男')
insert student values (3, '郭淑丽', '女')
insert student values (4, '韩旭', '女')
insert student values (5, '孟小飞', '男')
--创建成绩表
create table score
(
    studentid int,
    subjectname nvarchar(20),
    score decimal
)
--插入成绩
insert score values (1, '英语', 89)
insert score values (1, '数学', 98)
insert score values (2, '英语', 87)
insert score values (2, '数学', 78)
insert score values (3, '英语', 97)
insert score values (3, '数学', 86)
insert score values (6, '英语', 90)
insert score values (6, '数学', 93)
```

1. 查询所有学生的成绩

```
select a.*,b.* from student a join score b on a.studentid=b.studentid
```

```
select sname,subjectname,score from student a join score b on  
a.studentid=b.studentid
```

```
select a.studentid,sname,subjectname,score from student a join score b  
on a.studentid=b.studentid
```

2. 从多个表中合并数据

使用内连接

```
select a.*,b.* from student a join score b on a.studentid=b.studentid  
等价于
```

```
select a.*,b.* from student a inner join score b on a.studentid=b.studentid
```

使用外连接

左外连接

```
select a.studentid,sname,subjectname,score from dbo.student a left join  
dbo.score b on a.studentid=b.studentid
```

右外连接

```
select a.*,b.* from dbo.student a right join dbo.score b on  
a.studentid=b.studentid
```

练习:

1. 查找不及格同学 姓名和学科 分数

```
select sname,subjectname,score from dbo.student a join dbo.score b on  
a.studentid=b.studentid where score<60
```

自连接

查找到重名的学生

```
select a.*,b.* from dbo.student a join dbo.student b on a.sname=b.sname  
where a.studentid<>b.studentid
```

3. 合并多个结果集

数据类型相同，SQLServer 将自动删除结果集中的重复记录，如果指定了 all 结果集将返回所有的记录，第一个 select 语句中指定列名，如果想排序，在 union 子句中提拔

```
select sname,sex from dbo.student where sex='男'
union
select sname,sex from dbo.student where sex='女'
order by sex asc
```

4 最佳实践

进行表连接时，在主键或外部键的字段上连接

当表连接操作发生在多个字段组成的主键或外部键上时，on 子句中一定要引用所有组成键的字段

尽量限制连接的表的个数，连接的表越多，SQL Server 用于处理这个查询的时间越长。

5 试验 A 查询多个表

练习 1 表连接

练习 2 用操作符 union 合并结果集

第6章 数据分组和汇总

6.1 用 TOP n 列出前 n 个记录

结合 Order by 找出满足条件的前几条记录

1. 年龄最大的前 5 名学生

```
select top 5 * from dbo.TStudent order by Birthday
```

2. 年龄最小的前 5 名学生

```
select top 5 * from dbo.TStudent order by Birthday desc
```

3. 查找网络班年龄最小的前 5 名学生

```
select top 5 * from dbo.Tstudent where Class='网络班' order by Birthday desc
```

将 order by 排序相等的所有的记录显示出来。

6.1.1 使用聚集函数

可以在 Select 语句中单独使用聚集函数，也可以与语句 group by 联合使用

除了 count (*) 函数，如果没有满足 where 子句的记录，则所有的聚集函数都将返回空值，Count (*) 返回 0

Count(*) Count (列) Sum Min max Avg

统计表中有多少行

```
select count(*) from dbo.TStudent
```

统计表中 class 列 不为空的记录数量

```
insert dbo.TStudent (studentID,Sname,sex) values ('0000001901','王敬正','男')
```

```
select count(Class) from dbo.TStudent
```

求平均值

```
select avg(mark) from dbo.TScore
```

求最大值和最小值

```
select max(mark) from dbo.TScore
```

```
select min(mark) from dbo.TScore
```

求总和

```
select sum(mark) from dbo.TScore
```

使用含空值字段的聚集函数

6.2 Group by 使用基础

如果使用聚集函数，则将对表中的所有记录的某个字段进行汇总，然后生成单个的值。如果想生成多个汇总值，同时使用聚集函数和 group by 语句，联合使用 having 和 group by 子句能够使结果集只包含满足条件的记录。

联合使用 group by 子句与 having 子句

分组汇总

1. 统计每个班有多少个学生

```
select Class,count(*) from dbo.TStudent group by Class
```

2. 统计男生女生数量

```
select sex,count(*) from dbo.TStudent group by sex
```

3. 统计每科平均分

```
select subjectName,avg(mark) from dbo.TSubject a join dbo.TScore b on  
a.subjectID=b.subjectID  
  
group by subjectName
```

4. 统计每个学生的平均分

```
select a.StudentID,avg(mark) from dbo.TStudent a join dbo.TScore b on  
a.StudentID=b.StudentID  
  
group by a.StudentID order by avg(mark)
```

5. 查找平均分大于 95 的学生

```
select a.StudentID,avg(mark) from dbo.TStudent a join dbo.TScore b on  
a.StudentID=b.StudentID  
  
group by a.StudentID having avg(mark)>95
```

Having 相当于条件

6. 多列汇总

学科 3 班级 3 每科 每个班 总成绩

```
select subjectName,Class,sum(mark) 总分 from dbo.TStudent a join  
dbo.TScore b on a.StudentID=b.StudentID join  
dbo.TSubject c on b.subjectID=c.subjectID  
group by subjectName,Class  
order by 总分 desc
```

	subjectName	Class	总分
1	软件测试	开发班	71184
2	软件开发	开发班	70024
3	网络管理	开发班	69864
4	软件测试	测试班	38750
5	网络管理	测试班	38424
6	软件开发	测试班	38039
7	软件测试	网络班	34005
8	软件开发	网络班	33748
9	网络管理	网络班	33700

联合使用 group by 子句和操作符 rollup，将两列的详细信息和分组汇总

```
select subjectName,Class,sum(mark) from  
TStudent a join TScore b on a.StudentID=b.StudentID join TSubject c on  
b.subjectID=c.subjectID
```

```
group by subJectName,Class with rollup
```

	subjectName	Class	总分
1	软件测试	测试班	38750
2	软件测试	开发班	71184
3	软件测试	网络班	34005
4	软件测试	NULL	143939
5	软件开发	测试班	38039
6	软件开发	开发班	70024
7	软件开发	网络班	33748
8	软件开发	NULL	141811
9	网络管理	测试班	38424
10	网络管理	开发班	69864
11	网络管理	网络班	33700
12	网络管理	NULL	141988
13	NULL	NULL	427738

联合使用 cube，将对列出的 subJectName,Class 两列进行汇总

```
select subJectName,Class,sum(mark) 总分 from
TStudent a join TScore b on a.StudentID=b.StudentID join TSubject c on
b.subJectID=c.subJectID
group by subJectName,Class with cube
```

	subjectName	Class	总分
1	软件测试	测试班	38750
2	软件测试	开发班	71184
3	软件测试	网络班	34005
4	软件测试	NULL	143939
5	软件开发	测试班	38039
6	软件开发	开发班	70024
7	软件开发	网络班	33748
8	软件开发	NULL	141811
9	网络管理	测试班	38424
10	网络管理	开发班	69864
11	网络管理	网络班	33700
12	网络管理	NULL	141988
13	NULL	NULL	427738
14	NULL	测试班	115213
15	NULL	开发班	211072
16	NULL	网络班	101453

使用 grouping 函数，可以看出哪一行是汇总值

```
select
subJectName,grouping(subJectName),Class,grouping(Class),sum(mark)
from
```

```
TStudent a join TScore b on a.StudentID=b.StudentID join TSubject c on
b.subJectID=c.subJectID
group by subJectName,Class with cube
```

	subjectName	(无列名)	Class	(无列名)	总分
1	软件测试	0	测试班	0	38750
2	软件测试	0	开发班	0	71184
3	软件测试	0	网络班	0	34005
4	软件测试	0	NULL	1	143939
5	软件开发	0	测试班	0	38039
6	软件开发	0	开发班	0	70024
7	软件开发	0	网络班	0	33748
8	软件开发	0	NULL	1	141811
9	网络管理	0	测试班	0	38424
10	网络管理	0	开发班	0	69864
11	网络管理	0	网络班	0	33700
12	网络管理	0	NULL	1	141988
13	NULL	1	NULL	1	427738
14	NULL	1	测试班	0	115213
15	NULL	1	开发班	0	211072
16	NULL	1	网络班	0	101453

4. 最佳实践

为了改善查询操作的性能，在进行汇总操作的列上进行索引

尽量避免在含有空值的字段上使用聚集函数，因为此时你的到的结果集可能并不能代表你的数据

用 `order by` 子句为结果集中的记录排序。SQL Server 并不保证结果集中记录的顺序

如果可能，尽量使用 `rollup` 操作符，因为它比操作符 `cube` 的效率高得多

5. 试验 A 数据分组和汇总

练习 1 使用 `top n` 关键字

练习 2 使用 `group by` 和 `having` 子句

练习 3 使用 `rollup` 和 `cube` 操作符

第7章 用子查询

7.1 子查询介绍

为什么使用子查询---子查询可以把一个复杂的查询分解成一系列逻辑步骤，这样就可以用一个单个的语句解决复杂的查询问题。

为什么使用连接而不使用子查询---执行效力差不多，子查询可能要求查询优化器执行额外的操作，比如排序，而这些操作将会影响查询的处理策略。

7.2 把子查询用作表达式

查看每个学生成绩与平均分的差距

```
select StudentID, subJectID, mark - (select avg(mark) from dbo.TScore) 差距
from dbo.TScore
```

```
select StudentID, subJectID, case when (mark - (select avg(mark) from
dbo.TScore)) > 0 then '高于平均分' else '低于平均分' end
from dbo.TScore
```

查找高于均分 23 分的学生

```
select StudentID, mark from dbo.TScore where mark - (select avg(mark) from
dbo.TScore) > 23
```

7.3 用子查询关联数据

查找网络管理课程的学生成绩

逻辑更强

```
select * from dbo.TScore where subJectID = (select subJectID from
dbo.TSubject where subJectName = '网络管理')
等价于多表查询 多表查询效率更高
select a.* from dbo.TScore a join dbo.TSubject b on
a.subJectID = b.subJectID where subJectName = '网络管理'
```

7.4 把子查询用作派生的表

找到平均分低于 60 分的学生学号 姓名

```
select StudentID, Sname from dbo.TStudent where StudentID in (select
StudentID from
```

```
(select StudentID, avg(mark) 平均分 from dbo.TScore group by StudentID  
having avg(mark)<60) as t1)
```

第8章 数据修改

8.1 插入数据

```
insert dbo.TStudent values ('0000001902', '李维伟', '男'  
, '132302197506055634', '1984-3-4', 'liweiwei@91xueit.com', '网络班'  
, getdate())
```

8.2 插入部分数据

插入学生 学号 ;姓名 性别 其他列为空

```
insert dbo.TStudent (StudentID, sex, Sname) values ('0000001903', '男', '  
张国强')
```

8.3 将查询的数据插入现有表

将学生表中网络班的学生 查询出来 插入现有表中

```
insert dbo.TNetwork select * from dbo.TStudent where Class='网络班'
```

8.4 将查询记录插创建的新表

使用 Select into 创建表。

将学生表中 开发班 的从学生 查询到一个新表

```
select StudentID, Sname, sex, Email into TDe from dbo.TStudent where Class='  
开发班'
```

8.5 删除数据

删除学号是 0000000020 的学生

```
delete dbo.TStudent where StudentID='0000000020'
```

删除 1982 年以前出生的学生

```
delete dbo.TStudent where Birthday<'1982-1-1' and class='网络班'
```

8.6 删除基于其他表的行

下面的例子使用一个带 delete 语句的连接操作，删除不及格学生

Delete 条件使用子查询

```
delete dbo.TStudent where StudentID in (select distinct StudentID from
dbo.TScore where mark<60)
```

Delete 后面多表连接

```
delete dbo.TStudent from dbo.TStudent a join dbo.TScore b on
a.StudentID=b.StudentID where mark<60
```

8.7 更新数据

更改学生学号是 0000000569 的学生姓名为 韩立刚 性别 改成 男

```
update dbo.TStudent set Sname='韩立刚',sex='男' where
studentid='0000000569'
```

将网络班的学生邮箱更改为 姓名拼音@network.com

```
update dbo.TStudent set Email=left(Email,Len(email)-12)+'@network.com'
where Class='网络班'
```

基于其他表更新数据行

给软件测试 分数低于 70 分的学生加 5 分

使用子查询

```
update dbo.TScore set mark=mark+5 where mark<70 and subJectID=
(select subJectID from dbo.TSubject where subJectName='软件测试')
```

给软件测试 分数低于 80 分的学生加 5 分

多表连接

```
update dbo.TScore set mark=mark+5 from dbo.TScore a join dbo.TSubject b
on a.subJectID=b.subJectID
where b.subJectName='软件测试' and mark<80
```

8.8 性能因素

给事务中的数据加锁可阻止其他事务或查询的进行直到事务的结束

修改表可以修改数据的物理存储，同时导致事务中发生数据页的分配

如果正在被修改的数据列创建了索引，这些列的索引作为事务的一部分也被修改

数据修改语句 **where** 子句中的列的索引可以改进性能

7.最佳实践

在修改数据前先 **Select** 一下，察看修改前的数据

把列名改为别名或使用文字替换结果的值来提高计算结果的可读性。

如果不加条件 **delete** 和 **update** 将会删除或更新所有行。

8.试验 A 修改数据

练习 1 使用 **insert** 语句

练习 2 使用带 **default** 关键字的 **insert** 语句

练习 3 使用带 **default values** 关键字的 **Insert** 语句

练习 4 使用 **delete** 语句

练习 5 使用 **update** 语句

练习 6 基于其他表的数据修改表