苏 州 大 学 实 验 报 告

院、系	计算机学院	年级专业	19 计科图灵	姓名	张昊	学号	1927405160
课程名称	数据库课程实践					成绩	95
指导教师	赵朋朋	同组实验者	无		实验日期	2021 年 3 月 21 日	

实 验 名 称　SQL 语言实验 5-8

试验五 索引

一、实验目的

掌握索引的建立、删除的方法。

二、实验内容

（一） 创建索引

1 建 student 的索引

为姓名建立索引，索引名：Ix_student_sname

为系科建立索引，索引名：Ix_student_sdept

2 SC 的索引

为课程号建立索引： ix_sc_cno

3 Course 的索引

为课程名建立唯一性索引 ：Ix_course_cname

4 如何 SP_HELP 查看索引刚才建立的索引？

如何在企业管理器中查看索引？

（二） 删除索引 course 表的索引 IX_course_cname

（三） 思考：如何把索引 IX_student_sname 修改为唯一性索引？

（四） *思考建立索引的目的

1 输入下列存储过程，该程序生成大量数据供测试：

```
create procedure usp_makedata as
declare @nCnt int , @sNo varchar(6) , @sName varchar(8)
set @nCnt =12000    --计数器
while @nCnt<999999
begin
        set    @nCnt = @nCnt + 1
        set @sNo = convert(varchar(6) ,@nCnt)
        set @sName = '张'+@sNo
        insert into student (Sno,Sname,Ssex,Sage) values ( @sNo,@sName,'男',20)
end
return
```

2 exec usp_makedata --生成测试数据

3 输入下述测试程序：

```
create procedure usp_test as
declare @nCount int ,@data int
set @nCount=0
while @nCount<100
```

```
begin
        select @data=count(*) from student where Sname <'张 3800' or Sname>'张 8800'
        set @nCount =@nCount + 1
end
```

4 测试

1）建立姓名的索引，查看运行时间(8 秒).

```
create index Ix_student_sname on student(Sname)   --建立索引
        exec usp_test
```

2) 删除姓名索引，查看运行时间（2 分 11 秒），比较与 1）的时间长短。

```
drop index student.Ix_student_sname      --删除索引
        exec usp_test
```

三、实验结果

创建索引：

```
-- 建 student 的索引
CREATE INDEX Ix_student_sanme ON student(Sname)
CREATE INDEX Ix_student_sdept ON student(Sdept)
-- SC 的索引
CREATE INDEX Ix_sc_cno ON sc(Cno)
-- Course 的索引
CREATE INDEX Ix_course_cname ON course(Cname)
```

SP_HELP 查看索引：

```
SP_HELP student
```

100 %

结果 消息

	Name	Owner	Type	Created_datetime
1	student	dbo	user table	2021-03-21 15:55:25.200

	Column_name	Type	Computed	Length	Prec	Scale	Nullable
1	Sno	char	no	6			no
2	Sname	varchar	no	8			no
3	Ssex	char	no	2			no
4	Sage	smallint	no	2	5	0	yes
5	Sdept	varchar	no	15			no

	Identity	Seed	Increment	Not For Replicati
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	Ix_student_sanme	nonclustered located on PRIMARY	Sname
2	Ix_student_sdept	nonclustered located on PRIMARY	Sdept
3	PK__student__...	clustered, unique, primary k...	Sno
4	UQ__student__...	nonclustered, unique, unique...	Sname

```
SP_HELP sc
```

100 %

结果 消息

	Name	Owner	Type	Created_datetime
1	sc	dbo	user table	2021-03-21 15:55:25.210

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenN
1	Sno	char	no	6			no	no	no
2	Cno	char	no	4			no	no	no
3	Grade	decimal	no	9	12	1	yes	(n/a)	(n/a)

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	Ix_sc_cno	nonclustered located on PRIMARY	Cno
2	PK_SC	clustered, unique, primary key located on PRIMARY	Sno, Cno

```
SP_HELP course
```

100 %

结果 消息

	Name	Owner	Type	Created_datetime
1	course	dbo	user table	2021-03-21 15:55:25.207

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenN
1	Cno	char	no	4			no	no	no
2	Cname	varchar	no	20			yes	no	yes
3	Cpno	char	no	4			yes	no	yes
4	Ccredit	tinyint	no	1	3	0	yes	(n/a)	(n/a)

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	Ix_course_cname	nonclustered, unique located on PRIMARY	Cname
2	PK__course__C1FE637307E218B1	clustered, unique, primary key located on PRIMARY	Cno

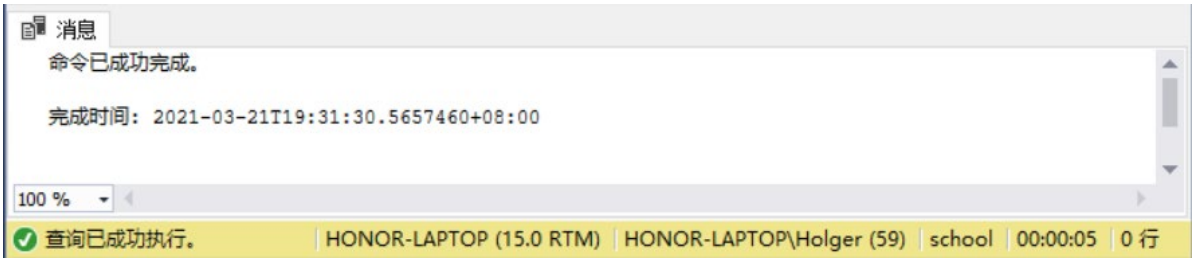
企业管理器查看索引：

删除索引：

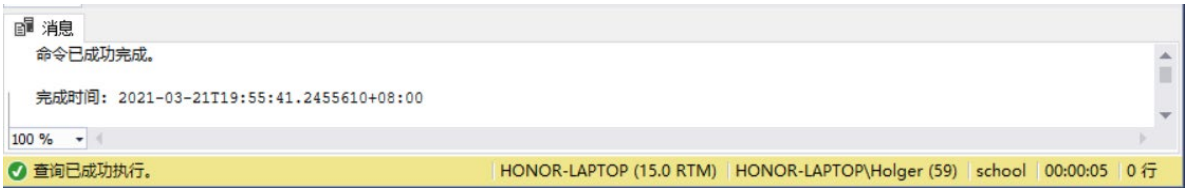
```
DROP INDEX course.Ix_course_cname
```

修改为唯一性索引：使用企业管理器，或者先删除再建立新的索引

建立姓名的索引，查看运行时间：5s



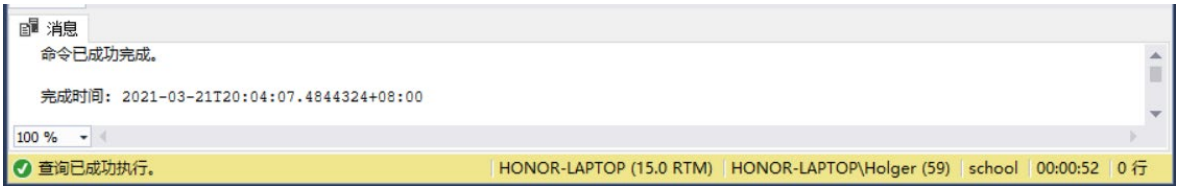
删除姓名索引，查看运行时间，比较与 1）的时间长短：5s



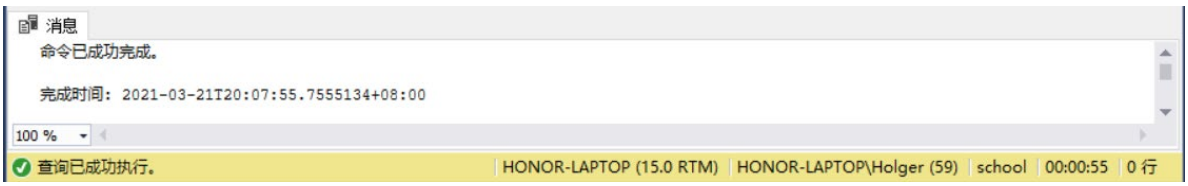
两者时间相差不大

个人推测是查询数据量的问题，将数据量改为 1000 后

建立索引所用时间为：52s



删除索引所用时间为：55s



删除索引后查询的时间变慢了。可见，索引在某些情况下可以在大批量的查询中加快速度的。

四、实验总结

通过实验，我掌握了索引的建立、删除的方法，理解了建立索引的作用。

试验六 更新数据

一、实验目的

掌握 insert,update ,delete 语句的使用。

二、实验内容

（一） insert

1 写出把下述学生的信息添加到 student 表中的命令。

学号	姓名	性别	年龄	系科
4001	赵茵	男	20	SX
4002	杨华	女	21	

2 批量插入数据

1） 建立一个新表 sc_name ，有属性 sno , sname , ssex , cno , grade 。

2） 把 SX 系学生的 sno,sname,ssex, cno , grade 插入到表 sc_name 中。

3） 察看 sc_name 表的数据

（二） Update

1 修改 0001 学生的系科为: JSJ

2 把陈小明的年龄加 1 岁，性别改为女。

3 修改李文庆的 1001 课程的成绩为 93 分

4 把“数据库原理”课的成绩减去 1 分

（三） Delete

1 删除所有 JSJ 系的男生

2 删除“数据库原理”的课的选课纪录

思考：修改数据的命令与修改表结构的命令有何区别？

三、实验结果

insert：

```
INSERT INTO student(Sno, Sname, Ssex, Sage, Sdept)
   VALUES (4001, '赵茵', '男', 20, 'SX')
INSERT INTO student(Sno, Sname, Ssex, Sage)
   VALUES (4002, '杨华', '女', 21)
```

批量插入数据

建立一个新表 sc_name：

```
CREATE TABLE sc_name (
   Sno    char(6),
   Sname  varchar(8),
   Ssex   char(2),
   Cno    char(4),
   Grade  decimal(12,1)
)
```

➢ 添加数据：(实验报告中没有，为了后续操作自己添加的)

```
INSERT INTO course  VALUES (1088, 'Java', NULL, 5)
INSERT INTO sc      VALUES (4001, 1088, 90)
INSERT INTO sc      VALUES (4002, 1088, 86)
```

把 SX 系学生的 sno,sname,ssex, cno , grade 插入到表 sc_name 中：

```
INSERT INTO sc_name(Sno, Sname, Ssex, Cno, Grade)
SELECT student.Sno, student.Sname, student.Ssex, sc.Cno, sc.Grade
FROM   student, sc
WHERE  student.Sno = sc.Sno AND Sdept = 'SX'
```

察看 sc_name 表的数据：

```
SELECT * FROM sc_name
```

结果 消息

	Sno	Sname	Ssex	Cno	Grade
1	4001	赵茵	男	1088	90.0

update：

初始数据：

```
SELECT * FROM student
SELECT * FROM course
SELECT * FROM sc
```

100 %

结果 消息

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	SX
2	4001	赵茵	男	20	SX
3	4002	杨华	女	21	JSJ
4	4003	陈小明	男	21	JSJ
5	4004	李四	女	19	JSJ
6	4005	李文庆	男	20	SX
7	4006	王五	男	21	JSJ

	Cno	Cname	Cpno	Ccredit
1	1001	C++	NULL	4
2	1002	数据库原理	NULL	4
3	1088	Java	NULL	5

	Sno	Cno	Grade
1	4001	1088	90.0
2	4002	1002	95.0
3	4002	1088	86.0
4	4003	1002	92.0
5	4004	1002	89.0
6	4005	1001	95.0
7	4006	1001	90.0
8	4006	1002	87.0

学生 0001 转专业：

```
UPDATE student SET Sdept = 'JSJ' WHERE Sno = '0001'
```

陈小明长大一岁，性别变成女：

```
UPDATE student SET Sage = Sage + 1, Ssex = '女' WHERE Sname = '陈小明'
```

李文庆课程 1001 改分数：

```
UPDATE sc      SET Grade = 93
WHERE Sno IN (SELECT Sno FROM student WHERE Sname = '李文庆')
```

全体数据库课程成绩-1：

```
UPDATE sc      SET Grade = Grade - 1
WHERE Cno IN (SELECT Cno FROM course WHERE Cname = '数据库原理')
```

更新后数据：

```
SELECT * FROM student
SELECT * FROM course
SELECT * FROM sc
```

100 %

结果 消息

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	JSJ
2	4001	赵茵	男	20	SX
3	4002	杨华	女	21	JSJ
4	4003	陈小明	女	22	JSJ
5	4004	李四	女	19	JSJ
6	4005	李文庆	男	20	SX
7	4006	王五	男	21	JSJ

	Cno	Cname	Cpno	Ccredit
1	1001	C++	NULL	4
2	1002	数据库原理	NULL	4
3	1088	Java	NULL	5

	Sno	Cno	Grade
1	4001	1088	90.0
2	4002	1002	94.0
3	4002	1088	86.0
4	4003	1002	91.0
5	4004	1002	88.0
6	4005	1001	93.0
7	4006	1001	90.0
8	4006	1002	86.0

delete：

初始数据：

```
SELECT * FROM student
SELECT * FROM course
SELECT * FROM sc
```

100 %

结果 消息

	Sno	Sname	Ssex	Sage	Sdept
1	0001	张三	男	20	JSJ
2	4001	赵茵	男	20	SX
3	4002	杨华	女	21	JSJ
4	4003	陈小明	女	22	JSJ
5	4004	李四	女	19	JSJ
6	4005	李文庆	男	20	SX
7	4006	王五	男	21	JSJ

	Cno	Cname	Cpno	Ccredit
1	1001	C++	NULL	4
2	1002	数据库原理	NULL	4
3	1088	Java	NULL	5

	Sno	Cno	Grade
1	4001	1088	90.0
2	4002	1002	94.0
3	4002	1088	86.0
4	4003	1002	91.0
5	4004	1002	88.0
6	4005	1001	93.0
7	4006	1001	90.0
8	4006	1002	86.0

删除全体 JSJ 系的男生（需要先将 sc 表引用 student 的外键改为级联删除）：

```
ALTER TABLE sc DROP FK__sc__Sno__2E1BDC42
ALTER TABLE sc ADD FOREIGN KEY (Sno)
REFERENCES student(Sno) ON DELETE CASCADE  -- 级联删除
```

```
DELETE FROM student WHERE Sdept = 'JSJ' AND Ssex = '男'
```

删除数据库选课记录：

```
DELETE FROM sc
WHERE Cno IN (SELECT Cno FROM course WHERE Cname = '数据库原理')
```

删除后数据：

```
SELECT * FROM student
SELECT * FROM course
SELECT * FROM sc
```

100 %

结果 消息

	Sno	Sname	Ssex	Sage	Sdept
1	4001	赵茵	男	20	SX
2	4002	杨华	女	21	JSJ
3	4003	陈小明	女	22	JSJ
4	4004	李四	女	19	JSJ
5	4005	李文庆	男	20	SX

	Cno	Cname	Cpno	Ccredit
1	1001	C++	NULL	4
2	1002	数据库原理	NULL	4
3	1088	Java	NULL	5

	Sno	Cno	Grade
1	4001	1088	90.0
2	4002	1088	86.0
3	4005	1001	93.0

修改数据的命令（使用 INSERT、UPDATE、DELETE 语句）：

```
INSERT INTO 表名 VALUES (所有属性值)  -- 插入（所有属性）
INSERT INTO 表名(部分属性名) VALUES (部分属性值)  -- 插入（部分属性）
UPDATE 表名 SET 属性名1=属性值1, 属性名2=属性值2, ... WHERE 条件  -- 更新
DELETE FROM 表名 WHERE 条件  -- 删除
```

修改表结构的命令（使用 ALTER TABLE 语句）：

```
ALTER TABLE 表名 DROP 键名  -- 删除主键/外键
ALTER TABLE 表名 ADD CONSTRAINT 主键名 PRIMARY KEY 属性  -- 添加主键
ALTER TABLE 主表名 ADD FOREIGN KEY (属性) REFERENCES 从表名(属性) -- 添加外键
ALTER TABLE 表名 ADD CHECK 约束条件  -- 添加约束
```

四、实验总结

通过实验，我掌握了使用 insert,update ,delete 语句来更新数据。

试验七 Sql 查询语句

一、实验目的

掌握 Select 查询语句。

二、实验内容

（一） 单表

1 查询年龄在 19 至 21 岁之间的女生的学号,姓名,年龄,按年龄从大到小排列。

2 查询姓名中第 2 个字为“明”字的学生学号、性别。

3 查询 1001 课程没有成绩的学生学号、课程号

4 查询 JSJ 、SX、WL 系的年龄大于 25 岁的学生学号,姓名，结果按系及学号排列

5 按 10 分制查询学生的 sno,cno,10 分制成绩

（1-10 分 为 1 ，11-20 分为 2 ，30-39 分为 3，。。。90-100 为 10）

6 查询 student 表中的学生共分布在那几个系中。（distinct）

7 查询 0001 号学生 1001，1002 课程的成绩。

（二） 统计

1 查询姓名中有“明”字的学生人数。

2 计算‘JSJ’系的平均年龄及最大年龄。

3 查询学生中姓名为张明、赵英的人数

4 计算每一门课的总分、平均分，最高分、最低分，按**平均分**由高到低排列

5 计算 1001,1002 课程的平均分。

6 查询平均分大于 80 分的学生学号及平均分

7 统计选修课程超过 2 门的学生学号

8 统计有 10 位成绩大于 85 分以上的课程号。

9 统计平均分不及格的学生学号

10 统计有大于两门课不及格的学生学号

（三） 连接

1 查询 JSJ 系的学生选修的课程号

2 查询选修 1002 课程的学生的学生姓名 (不用嵌套及嵌套 2 种方法）

3 查询数据库原理不及格的学生学号及成绩

4 查询选修“数据库原理”课且成绩 80 以上的学生姓名(不用嵌套及嵌套 2 种方法）

5 查询平均分不及格的学生的学号，姓名,平均分。

6 查询女学生平均分高于 75 分的学生姓名。

7 查询男学生学号、姓名、课程号、成绩。(一门课程也没有选修的男学生也要列出，不能遗漏)

（四） 嵌套、相关及其他

1 查询平均分不及格的学生人数

2 查询没有选修 1002 课程的学生的学生姓名

3 查询平均分最高的学生学号及平均分 （2 种方法 TOP , any , all）

*4 查询没有选修 1001，1002 课程的学生姓名。

5 查询 1002 课程第一名的学生学号（2 种方法）

6 查询平均分前三名的学生学号

7 查询 JSJ 系的学生与年龄不大于 19 岁的学生的差集

8 查询 1001 号课程大于 90 分的学生学号、姓名及平均分大于 85 分的学生学号、姓名

9 查询每门课程成绩都高于该门课程平均分的学生学号

10 查询大于本系科平均年龄的学生姓名

三、实验结果

单表

1 查询年龄在 19 至 21 岁之间的女生的学号,姓名,年龄,按年龄从大到小排列。

```
SELECT Sno, Sname, Sage
FROM student
WHERE Ssex = '女' AND Sage BETWEEN 19 AND 21
ORDER BY Sage DESC
```

结果 消息

	Sno	Sname	Sage
1	0004	杨秀红	21
2	8002	赵颖	21
3	8005	张英	21
4	0091	贺秋雪	20

2 查询姓名中第 2 个字为“明”字的学生学号、性别。

```
SELECT Sno, Ssex FROM student WHERE Sname LIKE '_明%'
```

结果 消息

	Sno	Ssex
1	0009	男

3 查询 1001 课程没有成绩的学生学号、课程号

```
SELECT Sno, Cno FROM sc WHERE Cno = 1001 AND Grade IS NULL
```

结果 消息

	Sno	Cno
1	8006	1001

4 查询 JSJ 、SX、WL 系的年龄大于 25 岁的学生学号,姓名，结果按系及学号排列

```
SELECT Sno, Sname
FROM student
WHERE Sdept IN ('JSJ', 'SX', 'WL') AND Sage > 25
ORDER BY Sdept, Sno
```

结果 消息

Sno	Sname

5 按 10 分制查询学生的 sno,cno,10 分制成绩

```
SELECT
    Sno,
    Cno,
    CASE WHEN Grade = 100 THEN 10 ELSE Grade / 10 + 1 END AS TEN_GRADE
FROM sc
```

结果 消息

	Sno	Cno	TEN_GRADE
1	0001	1001	10
2	0001	1002	10
3	0001	1003	8
4	0001	1004	10
5	0002	1001	10
6	0002	1002	10
7	0002	1003	7
8	0002	1004	10
9	0003	1001	8
10	0004	1001	10
11	0004	1002	8
12	0004	1003	7
13	0004	1004	9
14	0009	1001	10
15	0009	1002	9
16	0009	1003	7

查询已成功执行。 HONOR-LAPTOP (15.0 RTM) HONOR-LAPTOP\Holger (63) school 00:00:00 57 行

6 查询 student 表中的学生共分布在那几个系中。（distinct）

```
SELECT DISTINCT Sdept
FROM student
```

结果 消息

	Sdept
1	JSJ
2	SX

7 查询 0001 号学生 1001，1002 课程的成绩。

```
SELECT Grade
FROM sc
WHERE Sno = 0001 AND (Cno = 1001 OR Cno = 1002)
```

结果 消息

	Grade
1	92
2	92

统计

1 查询姓名中有“明”字的学生人数。

```
SELECT COUNT(*) FROM Student WHERE Sname LIKE '%明%'
```

结果 消息

	(无列名)
1	2

2 计算‘JSJ’系的平均年龄及最大年龄。

```
SELECT
    AVG(Sage) AS AVG_AGE,
    MAX(Sage) AS MAX_AGE
FROM Student
WHERE Sdept = 'JSJ'
```

结果 消息

	AVG_AGE	MAX_AGE
1	21	23

3 查询学生中姓名为张明、赵英的人数

```
SELECT
    (SELECT COUNT(*) FROM Student WHERE Sname = '张明') AS '张明',
    (SELECT COUNT(*) FROM Student WHERE Sname = '赵英') AS '赵英'
```

结果 消息

	张明	赵英
1	0	0

4 计算每一门课的总分、平均分，最高分、最低分，按平均分由高到低排列

```
SELECT
    Cno,
    SUM(Grade) AS SUM_GRADE,
    AVG(Grade) AS AVG_GRADE,
    MAX(Grade) AS MAX_GRADE,
    MIN(Grade) AS MIN_GRADE
```

```
FROM sc
GROUP BY Cno
ORDER BY AVG_GRADE DESC
```

结果 消息

	Cno	SUM_GRADE	AVG_GRADE	MAX_GRADE	MIN_GRADE
1	1005	476	95	98	92
2	1001	1291	92	99	77
3	1002	984	89	99	78
4	1004	752	83	98	63
5	1003	891	74	99	20

5 计算 1001,1002 课程的平均分。

```
SELECT
    Cno,
    AVG(Grade) AS AVG_GRADE
FROM sc
WHERE Cno = 1001 OR Cno = 1002
GROUP BY Cno
```

结果 消息

	Cno	AVG_GRADE
1	1001	92
2	1002	89

6 查询平均分大于 80 分的学生学号及平均分

```
SELECT
    Sno,
    AVG(Grade) AS AVG_GRADE
FROM sc
GROUP BY Sno
HAVING AVG(Grade) > 80
```

结果 消息

	Sno	AVG_GRADE
1	0001	87
2	0002	86
3	0004	82
4	0009	81
5	0081	90
6	0091	93
7	0092	92
8	8001	86
9	8002	85
10	8003	85
11	8004	83
12	8007	94
13	8008	91

7 统计选修课程超过 2 门的学生学号

```
SELECT Sno
FROM sc
GROUP BY Sno
HAVING COUNT(*) > 2
```

结果 消息

	Sno
1	0001
2	0002
3	0004
4	0009
5	0081
6	0091
7	0092
8	8001
9	8002
10	8003
11	8004
12	8006
13	8007
14	8008

8 统计有 10 位成绩大于 85 分以上的课程号。

```
SELECT Cno FROM sc WHERE Grade > 85 GROUP BY Cno HAVING COUNT(*) = 10
```

结果 消息

	Cno
1	1002

9 统计平均分不及格的学生学号

```
SELECT Sno FROM sc GROUP BY Sno HAVING AVG(Grade) < 60
```

结果 消息

	Sno
1	8006

10 统计有大于两门课不及格的学生学号

```
SELECT Sno FROM sc WHERE Grade < 60 GROUP BY Sno HAVING COUNT(*) > 2
```

结果 消息

Sno

连接

1 查询 JSJ 系的学生选修的课程号

```
SELECT DISTINCT Cno
FROM sc, Student
WHERE Sdept = 'JSJ' AND sc.Sno = Student.Sno
```

结果 消息

	Cno
1	1001
2	1002
3	1003
4	1004
5	1005

2 查询选修 1002 课程的学生的学生姓名 (不用嵌套及嵌套 2 种方法）

不嵌套：

```
SELECT Sname
FROM sc, Student
WHERE Cno = 1002 AND sc.Sno = Student.Sno
```

嵌套：

```
SELECT Sname
FROM Student
WHERE Sno IN (SELECT Sno FROM sc WHERE Cno = 1002)
```

结果 消息

	Sname
1	周志林
2	李文庆
3	杨秀红
4	钱明明
5	刘亭
6	贺秋雪
7	赵三
8	张华
9	赵颖
10	赵章
11	钱利
12	王铁

3 查询数据库原理不及格的学生学号及成绩

```
SELECT Sno, Grade
FROM sc, course
WHERE sc.Cno=course.Cno AND Grade < 60 AND Cname = '数据库原理'
```

结果 消息

Sno	Grade

4 查询选修“数据库原理”课且成绩 80 以上的学生姓名(不用嵌套及嵌套 2 种方法）

不用嵌套：

```
SELECT Sname
FROM Student, course, sc
WHERE Student.Sno = sc.Sno AND sc.Cno=course.Cno AND
    Grade > 80 AND Cname = '数据库原理'
```

嵌套：

```
SELECT Sname FROM Student WHERE Sno IN (
    SELECT Sno FROM sc WHERE Grade > 80 AND Cno IN (
        SELECT Cno FROM course WHERE Cname = '数据库原理'))
```

结果 消息

	Sname
1	张华
2	赵颖
3	钱凯
4	钱利
5	王铁

5 查询平均分不及格的学生的学号，姓名,平均分。

```
SELECT Student.Sno, Sname, AVG(Grade) AS AVG_GRADE
FROM sc, Student
WHERE sc.Sno = Student.Sno
GROUP BY Student.Sno, Sname
```

```
HAVING AVG(Grade) < 60
```

结果 消息

	Sno	Sname	AVG_GRADE
1	8006	赵章	56

6 查询女学生平均分高于 75 分的学生姓名。

```
SELECT Sname
FROM sc, Student
WHERE sc.Sno = Student.Sno AND Ssex = '女'
GROUP BY Sname
HAVING AVG(Grade) > 75
```

结果 消息

	Sname
1	杨秀红
2	刘亭
3	贺秋雪
4	赵颖

7 查询男学生学号、姓名、课程号、成绩。(一门课程也没有选修的男学生也要列出，不能遗漏)

```
SELECT Student.Sno, Sname, Cno, Grade
FROM Student
LEFT JOIN sc
ON Student.Sno = sc.Sno
WHERE Ssex = '男'
```

结果 消息

	Sno	Sname	Cno	Grade
1	0001	周志林	1001	92
2	0001	周志林	1002	92
3	0001	周志林	1003	76
4	0001	周志林	1004	91
5	0002	李文庆	1001	91
6	0002	李文庆	1002	90
7	0002	李文庆	1003	67
8	0002	李文庆	1004	98
9	0003	陈小明	1001	77
10	0009	钱明明	1001	93
11	0009	钱明明	1002	88
12	0009	钱明明	1003	60
13	0009	钱明明	1004	83
14	0078	王振	NULL	NULL
15	0092	赵三	1001	98
16	0092	赵三	1002	86

查询已成功执行。 HONOR-LAPTOP (15.0 RTM) HONOR-LAPTOP\Holger (63) school 00:00:00 39 行

嵌套、相关及其他

1 查询平均分不及格的学生人数

```
SELECT COUNT(*)
FROM Student
WHERE Sno IN ( SELECT Sno FROM sc GROUP BY Sno HAVING AVG(Grade) < 60)
```

结果 消息

	(无列名)
1	1

2 查询没有选修 1002 课程的学生的学生姓名

```sql
SELECT Sname
FROM Student
WHERE Sno NOT IN ( SELECT Sno FROM sc WHERE Cno = 1002)
```

结果 消息

	Sname
1	陈小明
2	钱凯
3	王华
4	王振
5	张英

3 查询平均分最高的学生学号及平均分 （2 种方法 TOP , any , all）

TOP：

```sql
SELECT TOP 1 Sno, AVG(Grade) AS AVG_GRADE
FROM sc
GROUP BY Sno
ORDER BY AVG(Grade) DESC
```

Any：

```sql
SELECT Sno, AVG(Grade) AS AVG_GRADE
FROM sc
GROUP BY Sno
HAVING NOT (AVG(Grade) < ANY(
    SELECT AVG(Grade)
    FROM sc
    GROUP BY Sno))
```

All：

```sql
SELECT Sno, AVG(Grade) AS AVG_GRADE
FROM sc
GROUP BY Sno
HAVING AVG(Grade) >= ALL(
    SELECT AVG(Grade)
    FROM sc
    GROUP BY Sno)
```

结果 消息

	Sno	AVG_GRADE
1	8007	94

*4 查询没有选修 1001，1002 课程的学生姓名。

```sql
SELECT Sname
FROM Student
WHERE Sno NOT IN (
    SELECT DISTINCT Sno
    FROM sc
    WHERE Cno = 1001 OR Cno = 1002)
```

结果 消息

	Sname
1	王振
2	张英

5 查询 1002 课程第一名的学生学号（2 种方法）

All：(需要排除没有成绩的学生才有查询结果)

```
SELECT Sno
FROM sc
WHERE
    Cno = 1002 AND Grade >= ALL(
        SELECT Grade FROM sc WHERE Cno = 1002 AND Grade IS NOT NULL)
```

TOP：

```
SELECT TOP 1 Sno FROM sc WHERE Cno = 1002 ORDER BY Grade DESC
```

结果 消息

	Sno
1	8008

6 查询平均分前三名的学生学号

```
SELECT TOP 3 Sno FROM sc GROUP BY Sno ORDER BY AVG(Grade) DESC
```

结果 消息

	Sno
1	8007
2	0091
3	0092

7 查询 JSJ 系的学生与年龄不大于 19 岁的学生的差集

```
(SELECT * FROM Student WHERE Sdept = 'JSJ')
EXCEPT
(SELECT * FROM Student WHERE Sage <= 19)
```

结果 消息

	Sno	Sname	Ssex	Sage	Sdept
1	0002	李文庆	男	23	JSJ
2	0004	杨秀红	女	21	JSJ
3	0078	王振	男	21	JSJ
4	8003	钱凯	男	22	JSJ
5	8005	张英	女	21	JSJ
6	8006	赵章	女	22	JSJ
7	8007	钱利	男	23	JSJ
8	8008	王铁	男	21	JSJ

8 查询 1001 号课程大于 90 分的学生学号、姓名及平均分大于 85 分的学生学号、姓名

```
(
    SELECT sc.Sno, Sname
    FROM sc, Student
    WHERE sc.Sno = Student.Sno AND Cno = 1001 AND Grade > 90
)
UNION
(
    SELECT sc.Sno, Sname
    FROM sc, Student
    WHERE sc.Sno = Student.Sno
    GROUP BY sc.Sno, Sname
    HAVING AVG(Grade) > 85
)
```

结果 消息

	Sno	Sname
1	0001	周志林
2	0002	李文庆
3	0004	杨秀红
4	0009	钱明明
5	0081	刘亭
6	0091	贺秋雪
7	0092	赵三
8	8001	张华
9	8002	赵颖
10	8003	钱凯
11	8004	王华
12	8007	钱利
13	8008	王铁

9　查询每门课程成绩都高于该门课程平均分的学生学号

```
CREATE VIEW ARG_VIEW (Cno, AVG_GRADE)
AS (SELECT Cno, AVG(Grade) FROM sc GROUP BY Cno)
GO
SELECT DISTINCT Sno
FROM sc
WHERE sno NOT IN (
    SELECT Sno
    FROM sc, ARG_VIEW
    WHERE sc.Cno = ARG_VIEW.Cno AND Grade <= AVG_GRADE
)
```

结果 消息

	Sno
1	8007

10 查询大于本系科平均年龄的学生姓名

```
SELECT SNAME
FROM Student AS M_ST
WHERE M_ST.Sage > (
    SELECT AVG(Sage)
    FROM Student AS S_ST
    WHERE S_ST.Sdept = M_ST.Sdept
)
```

结果 消息

	SNAME
1	李文庆
2	赵章
3	钱利
4	钱凯
5	刘亭
6	赵三
7	张华

四、实验总结

通过实验，我掌握了单表、多表使用掌握 Select 查询语句进行查询、统计、连接的方法。

试验八、视图

一、实验目的

掌握视图的建立、使用。

二、实验内容

1 建立学生学号、姓名、性别、课程号、成绩的视图 v_sc

查看 V_sc 中的数据。

2 建立学生学号、姓名、出生年份的视图 v_age

查看 V_age 中的数据。

3 建立 'JSJ' 系的学生学号、姓名、性别、年龄的视图 V_JSJ

4 建立每门课程的平均分的视图 V_avggrade

5 将 视图 v_jsj 中 李文庆 的年龄改为 21 岁

6 察看 student 中李文庆的年龄

查看 v_age 中李文庆的出生年月

7 查询每门课程的及格率

思考:

1 利用 V_JSJ 视图,可以更新 SX 的学生的年龄吗? 写出理由

如: update v_jsj set sage=25 where sno=' 0004'

0004 号学生为 SX 系.

2 create view v_student (sno,sname,ssex,sage,sdept) as

select sno,sname,ssex,sage,sdept from student

学籍管理系统使用此视图。现把 student 的列名 sname 改为 xm,请问原来的学籍管理系统能正常运行吗? 如不能，则如何处理能使系统正常运行。

三、实验结果

学生学号、姓名、性别、课程号、成绩的视图 v_sc

```
CREATE VIEW v_sc(Sno, Sname, Ssex, Cno, Grade) AS
SELECT Student.Sno, Sname, Ssex, Cno, Grade
FROM Student, SC WHERE Student.Sno = SC.Sno

SELECT * FROM v_sc
```

结果 消息

	Sno	Sname	Ssex	Cno	Grade
1	0001	周志林	男	1001	92
2	0001	周志林	男	1002	92
3	0001	周志林	男	1003	76
4	0001	周志林	男	1004	91
5	0002	李文庆	男	1001	91
6	0002	李文庆	男	1002	90
7	0002	李文庆	男	1003	67
8	0002	李文庆	男	1004	98
9	0003	陈小明	男	1001	77
10	0004	杨秀红	女	1001	97
11	0004	杨秀红	女	1002	78
12	0004	杨秀红	女	1003	65
13	0004	杨秀红	女	1004	89
14	0009	钱明明	男	1001	93
15	0009	钱明明	男	1002	88
16	0009	钱明明	男	1003	60
17	0009	钱明明	男	1004	83
18	0081	刘亭	女	1001	90
19	0081	刘亭	女	1002	NULL
20	0081	刘亭	女	1003	NULL
21	0091	贺秋雪	女	1001	93
22	0091	贺秋雪	女	1002	87
23	0091	贺秋雪	女	1003	99

查询已成功执行。 (local) (9.0 RTM) LAB-243\Administrator (52) school 00:00:00 57 行

学生学号、姓名、出生年份的视图 v_age

```
CREATE VIEW v_age(Sno, Sname, Sbirth) AS
SELECT Sno, Sname, 2021 - Sage FROM Student

SELECT * FROM v_age
```

结果 消息

	Sno	Sname	Sbirth
1	0001	周志林	2001
2	0002	李文庆	1998
3	0003	陈小明	2001
4	0004	杨秀红	2000
5	0009	钱明明	2001
6	0078	王振	2000
7	0081	刘亭	1999
8	0091	贺秋雪	2001
9	0092	赵三	1999
10	8001	张华	1998
11	8002	赵颖	2000
12	8003	钱凯	1999
13	8004	王华	2000
14	8005	张英	2000
15	8006	赵章	1999
16	8007	钱利	1998
17	8008	王铁	2000

'JSJ' 系的学生学号、姓名、性别、年龄的视图 V_JSJ

```
CREATE VIEW V_JSJ(Sno, Sname, Ssex, Sage) AS
SELECT Sno, Sname, Ssex, Sage FROM Student WHERE Sdept = 'JSJ'

SELECT * FROM V_JSJ
```

结果 消息

	Sno	Sname	Ssex	Sage
1	0002	李文庆	男	23
2	0004	杨秀红	女	21
3	0078	王振	男	21
4	8003	钱凯	男	22
5	8005	张英	女	21
6	8006	赵章	女	22
7	8007	钱利	男	23
8	8008	王铁	男	21

每门课程的平均分的视图 V_avggrade

```
CREATE VIEW v_average(Cno, Grage) AS
SELECT Cno,  AVG(Grade) FROM SC GROUP BY Cno

SELECT * FROM v_average
```

结果 消息

	Cno	Grage
1	1001	92
2	1002	89
3	1003	74
4	1004	83
5	1005	95

视图 v_jsj 中 李文庆 的年龄改为 21 岁

```
UPDATE V_JSJ SET Sage = 21 WHERE Sname = '李文庆'
```

student 中李文庆的年龄

```
SELECT Sage FROM Student WHERE Sname = '李文庆'
```

结果 消息

	Sage
1	21

v_age 中李文庆的出生年月

```
SELECT Sbirth FROM v_age WHERE Sname = '李文庆'
```

结果 消息

	Sbirth
1	2000

每门课程的及格率

```
CREATE VIEW V_SC_TOTAL(cno, TotalNumber) AS
SELECT Cno, COUNT(*) FROM SC GROUP BY Cno

CREATE VIEW V_SC_PASS(Cno, PassNumber) AS
SELECT Cno, COUNT(*) FROM SC WHERE Grade >= 60 GROUP BY Cno

SELECT V_SC_TOTAL.Cno, PassNumber * 1.0 / TotalNumber AS PassRate
FROM V_SC_TOTAL, V_SC_PASS
WHERE V_SC_TOTAL.Cno = V_SC_PASS.Cno
```

结果 消息

	Cno	PassRate
1	1001	0.933333333333
2	1002	0.916666666666
3	1003	0.785714285714
4	1004	0.900000000000
5	1005	0.833333333333

思考 1：利用 V_JSJ 视图,可以更新 SX 的学生的年龄吗？

不可以，因为 V_JSJ 视图中不包含 SX 学生的记录

思考 2：把 student 的列名 sname 改为 xm，原来的学籍管理系统能正常运行吗？

```
EXEC SP_RENAME 'Student.Sname', 'xm'

SELECT * FROM v_student
```

不能正常运行：

消息

```
消息 207，级别 16，状态 1，第 1 行
列名 'sname' 无效。
消息 4413，级别 16，状态 1，第 1 行
由于绑定错误，无法使用视图或函数 'v_student'。
```

解决办法：重建视图

```
ALTER VIEW v_student(sno,sname,ssex,sage,sdept) AS
SELECT sno,xm,ssex,sage,sdept FROM student
```

```
SELECT * FROM v_student
```

结果 消息

	sno	sname	ssex	sage	sdept
1	0001	周志林	男	20	SX
2	0002	李文庆	男	21	JSJ
3	0003	陈小明	男	20	SX
4	0004	杨秀红	女	21	JSJ
5	0009	钱明明	男	20	SX
6	0078	王振	男	21	JSJ
7	0081	刘亭	女	22	SX
8	0091	贺秋雪	女	20	SX
9	0092	赵三	男	22	SX
10	8001	张华	男	23	SX
11	8002	赵颖	女	21	SX
12	8003	钱凯	男	22	JSJ
13	8004	王华	男	21	SX
14	8005	张英	女	21	JSJ
15	8006	赵章	女	22	JSJ
16	8007	钱利	男	23	JSJ
17	8008	王铁	男	21	JSJ

四、实验总结

通过实验，我学会了视图的建立和使用，了解了使用视图进行增删改查。