

数据库实验报告

生信 2001 张子栋 2020317210101

- GitHub: [MarkdownNotes/数据库实验报告.md at main · Bluuur/MarkdownNotes \(github.com\)](#)
- 同步至 Gitee: [Data Base/数据库实验报告.md · blur/MarkdownNotes \(gitee.com\)](#)

实验1 数据查询和更新

1. 找出所有客户、代理商和商品都在同一城市的三元组 (cid, aid, pid) 。

```
1 select cid, aid, pid
2 from agents,
3     customers,
4     products
5 where agents.city = customers.city
6 and customers.city = products.city;
```

2. 找出所有客户、代理商和商品不都在同一城市 (可能有两个在同一城市) 的三元组 (cid, aid, pid) 。

```
1 select cid, aid, pid
2 from agents,
3     customers,
4     products
5 where agents.city != customers.city
6 or customers.city != products.city;
```

3. 找出所有在同一城市的代理商的aid对。

```
1 select first.aid, second.aid
2 from agents first,
3     agents second
4 where first.city = second.city
5 and first.aid != second.aid;
```

4. 找出同时订购了商品p01和p07的客户的cid值。(若找出客户的cname呢?)

```
1 select orders.cid, cname
2 from orders,
3     customers
4 where orders.pid = 'p01'
5 intersect
6 select orders.cid, cname
7 from orders,
8     customers
9 where orders.pid = 'p07';
```

5. 统计各个产品的销售总量。

```
1 select pid, sum(qty)
2 from orders
3 group by pid;
```

6. 当某个代理商所订购的某样产品的总量超过1000时，打印出所有满足条件的产品和代理商的ID以及这个总量。

```
1 select pid, aid, sum(qty)
2 from orders
3 group by pid, aid
4 having sum(qty) > 1000;
```

7. 找出订购了产品p05的顾客的名字。

```
1 select distinct cname
2 from customers,
3     orders
4 where orders.pid = 'p05'
5        and customers.cid = orders.cid;
```

8. 检索满足以下条件的顾客-代理商的姓名对 (cname,aname) ,其中的顾客cname通过代理商aname订了货。

```
1 select cname, aname
2 from orders,
3     customers,
4     agents
5 where customers.cid = orders.cid
6        and agents.aid = orders.aid;
```

9. 找出至少被两个顾客订购的产品的pid值。

```
1 select distinct pid
2 from orders
3 where pid in (select pid from orders group by pid having count(pid) > 1)
```

10. 在customers表中插入一个新行。

```
1 insert into customers(cid, cname, city)
2 values ('c007', 'winDix', 'Dallas');
```

11. 检索customers表中discnt值为空的行。

```
1 select*
2 from customers
3 where discnt is null
```

12. 检索客户以及他们订购商品的详细信息。（用外联接）

```

1  select customers.cid,
2      cname,
3      city,
4      discnt,
5      ordno,
6      month,
7      aid,
8      pid,
9      qty,
10     dollars
11  from customers
12     left outer join orders on (customers.cid = orders.cid)

```

13. 检索有关住在Duluth或Dallas的代理商的所有信息。（要求使用IN谓词实现）

```

1  select*
2  from agents
3  where city in ('Duluth', 'Dallas')

```

14. 找出通过住在Duluth或Dallas的代理商订货的所有顾客的姓名和折扣率。（要求使用IN谓词实现）

```

1  select cname, discnt
2  from customers,
3       orders
4  where customers.cid = orders.cid
5         and orders.aid in (select aid
6                             from agents
7                             where city in ('Duluth', 'Dallas'))

```

15. 求所有满足以下条件的顾客的cid值：该顾客的discnt的值小于任一住在Duluth的顾客的discnt值。

```

1  select cid
2  from customers
3  where discnt < any (select discnt
4                     from customers
5                     where city = 'Duluth')

```

16. 检索没有通过代理商a05订货的所有顾客的名字。

提示：可以使用not in 或 <>all方式实现。

```

1  select cname
2  from customers,
3       orders
4  where customers.cid = orders.cid
5         and aid <> 'a05'

```

17. 检索一个包含顾客所在的或者代理商所在的城市名称。（使用UNION实现）

```

1  select city
2  from customers
3  union
4  select city
5  from agents

```

18. 在orders表中插入一个新行。

```
1 insert into orders(ordno, month, cid, aid, pid)
2 values (1107, 'aug', 'c006', 'a04', 'p01');
```

19. 创建一个名为swcusts的表，它包含住在西南部的所有顾客，并向该表中插入所有来自Dallas或Austin的顾客。

```
1 insert into swcusts
2 select *
3 from customers
4 where city in ('Dallas', 'Austin');
```

20. 将所有住在New York的代理商的佣金率提高10%。

```
1 update agents
2 set per=per * 1.1
3 where city = 'New York'
```

21. 删除所有住在New York的代理商。

```
1 delete
2 from agents
3 where city = 'New York'
```

22. 创建一个agentorders视图，它扩展了表orders的行，包括订货的代理商的详细信息。

```
1 create view agentorders
2 as
3 select ordno,
4         month,
5         cid,
6         orders.aid,
7         pid,
8         qty,
9         dollars,
10        aname,
11        city,
12        per
13 from orders,
14        agents
15 where orders.aid = agents.aid;
```

23. 利用agentorders视图查询代理商Brown的所有订单信息

```
1 select ordno, motn, cid, pid, qty, dollars
2 from agentorders
3 where aname = 'Brown';
```

24. 创建cacities视图，该视图列出表customers和表agents中所有配对的城市，其中该顾客通过该代理商订购了商品。

```
1 create view cacities(cid.cname, city, discnt, aid, aname, per)
```

```

2  as
3  select customers.cid,
4         cname,
5         customers.city,
6         discnt,
7         agents.aid,
8         aname,
9         per
10 from agents,
11      customers,
12      orders
13 where agent.aid = orders.aid
14        and customers.cid = orders.cid
15        and agents.city = customers.city

```

25. 创建custs视图

```

1  create view custs as
2  select *
3  from customers
4  where discnt <= 15.0
5  with check option;

```

26. 对custs视图进行更新操作。

```

1  update custs set discnt=discnt+4;

```

实验2 学生-课程-选修

实验目的

熟练掌握建立数据库和表，向数据库输入数据、修改数据、删除数据的操作。

实验内容

建立数据库并设计各表，输入多条实际数据，并实现数据的增、删、改操作。

实验步骤

创建用于学生管理数据库，数据库名为 `StudentCourse`，包含学生的基本信息，课程信息和选课信息。数据库 `StudentCourse` 包含下列3个表：

1. student：学生基本信息。
2. course：课程信息表。
3. sc：学生选课表。

各表的结构分别如表1、表2和表3所示。

表1 学生信息表：student

列名	数据类型	长度	是否允许为空值
sno	字符型	10	否
sname	字符型	10	否
ssex	字符型	2	否
sage	整数型		是
sdept	字符型	4	否

表2 课程信息表：course

列名	数据类型	长度	是否允许为空值
cno	字符型	3	否
cname	字符型	30	否
credit	整数型		是
pcno	字符型	3	是

表3 学生选课表：sc

列名	数据类型	长度	是否允许为空值
sno	字符型	10	否
cno	字符型	3	否
grade	整数型		是

SQL 语句

建表语句

```

1  create table student
2  (
3      sno    char(10),
4      sname  char(10),
5      ssex   char(2),
6      sage   int,
7      sdept  char(4)
8  );
9
10 create table course
11 (
12     cno     char(3),
13     cname   char(30),
14     credit  int,
15     pcno    char(3)
16 );
17
18 create table sc
19 (

```

```

20     sno   char(10),
21     cno   char(3),
22     grade int
23 );
24

```

写入数据

这里使用 MySQL 完成, 可以一次插入多条数据.

```

1  insert into student (sno, sname, ssex, sage, sdept)
2  values ('95001', '李勇', '男', 18, 'CS'),
3         ('95002', '刘晨', '女', 19, 'IS'),
4         ('95003', '王敏', '女', 18, 'MA'),
5         ('95004', '张立', '男', 19, 'IS'),
6         ('95005', '刘云', '女', 18, 'CS');
7
8  insert into course (cno, cname, credit, pcno)
9  values ('1', '数据库', 4, '5'),
10         ('2', '数学', 6, null),
11         ('3', '信息系统', 3, '1'),
12         ('4', '操作系统', 4, '6'),
13         ('5', '数据结构', 4, '7'),
14         ('6', '数据处理', 3, null),
15         ('7', 'PASCAL语言', 4, '6');
16
17 insert into sc (sno, cno, grade)
18 values ('95001', '1', 92),
19         ('95001', '2', 85),
20         ('95001', '3', 88),
21         ('95002', '2', 90),
22         ('95002', '3', 80),
23         ('95003', '2', 85),
24         ('95004', '1', 58),
25         ('95004', '2', 85);

```

1. 查询全体学生的学号和姓名

```

1  select sno, sname
2  from student;

```

	sno	sname
1	95001	李勇
2	95002	刘晨
3	95003	王敏
4	95004	张立
5	95005	刘云

2. 查询选修了课程名为"数据库原理"的学生的学号和姓名

```

1 select distinct student.sno, sname
2 from student,
3     sc,
4     course
5 where course.cno = sc.cno
6       and student.sno = sc.sno
7       and sc.cno = 1;

```

	sno	sname
1	95004	张立
2	95001	李勇

3. 查询全体学生的姓名, 出生年份, 和所在系, 并用小写字母表示所有系名, 并给各列指定列名。

```

1 select sname name, 2022 - sage 'birth', lower(sdept) department
2 from student;

```

	name	birth	department
1	李勇	2004	cs
2	刘晨	2003	is
3	王敏	2004	ma
4	张立	2003	is
5	刘云	2004	cs

4. 查询有多少名学生的数据库课程成绩不及格

```

1 select count(cno) 'num'
2 from sc
3 where grade < 60
4       and cno = 1;

```

	num
1	1

5. 查找所有姓"李"的学生的姓名, 学号和性别

```

1 select sname, sno, ssex
2 from student
3 where sname like '李%';

```

	sname	sno	ssex
1	李勇	95001	男

6. 求没有选修数学课程的学生学号

```

1 select sno
2 from student
3 where sno not in (select sno from sc where cno = 2);

```


sno
1 95005

7. 查询选修了课程的学生学号

```
1 select distinct sno
2 from student
3 where sno in (select sno from sc);
```

sno
1 95001
2 95002
3 95003
4 95004

8. 计算1号课程的学生平均成绩, 最高分和最低分

```
1 select avg(grade), max(grade), min(grade)
2 from sc
3 where cno = 1
```

	`avg(grade)`	`max(grade)`	`min(grade)`
1	75.0000	92	58

9. 查询数学系和信息系的学生信息

```
1 select*
2 from student
3 where sdept in ('MA', 'IS');
```

sno	sname	ssex	sage	sdept
1 95002	刘晨	女	19	IS
2 95003	王敏	女	18	MA
3 95004	张立	男	19	IS

10. 将年龄为19岁的学生的成绩置零

```
1 update sc
2 set grade=0
3 where sno in (select sno from student where sage = 19);
```

	sno	sname	ssex	sage	sdept
1	95001	李勇	男	18	CS
2	95002	刘晨	女	19	IS
3	95003	王敏	女	18	MA
4	95004	张立	男	19	IS
5	95005	刘云	女	18	CS

	sno	cno	grade
1	95001	1	92
2	95001	2	85
3	95001	3	88
4	95002	2	0
5	95002	3	0
6	95003	2	85
7	95004	1	0
8	95004	2	0

11. 查询所有选修了1号课程的学生姓名

```

1 select sname
2 from student,
3     sc
4 where student.sno = sc.sno
5        and cno = 1;

```

	sname
1	李勇
2	张立

12. 对每一个性别，求学生的平均年龄，并把结果存入数据库

```

1 create table sex_age
2 (
3     ssex    char(15),
4     avg_age smallint
5 );
6
7 insert
8 into sex_age(ssex, avg_age)
9 select ssex, avg(sage)
10 from student
11 group by ssex;

```

	ssex	avg_age
1	男	19
2	女	18

13. 查询每个学生已获得的学分

```

1 select sno, sum(credit)
2 from course,
3     sc
4 where sc.cno = course.cno
5 group by sno;

```

	sno	sum(credit)
1	95001	13
2	95002	9
3	95003	6
4	95004	10

14. 将所有女生的记录定义为一个视图

```

1 create view female_student
2 as
3 select*
4 from student
5 where ssex = '女';

```

	sno	sname	ssex	sage	sdept
1	95002	刘晨	女	19	IS
2	95003	王敏	女	18	MA
3	95005	刘云	女	18	CS

15. 查询没有选修了1号课程的学生姓名

```

1 select distinct sname
2 from student,
3     sc
4 where student.sno not in (select sno from sc where cno = 1);

```

	sname
1	刘云
2	王敏
3	刘晨

16. 将所有选修了数据库课程的学生们的成绩加5分

```

1 update sc
2 set grade=grade + 5
3 where cno = 1;

```

WHERE	Before			
	sno	cno	grade	
1	95001	1	92	
2	95001	2	85	
3	95001	3	88	
4	95002	2	0	
5	95002	3	0	
6	95003	2	85	
7	95004	1	0	
8	95004	2	0	

WHERE	After			
	sno	cno	grade	
1	95001	1	97	
2	95001	2	85	
3	95001	3	88	
4	95002	2	0	
5	95002	3	0	
6	95003	2	85	
7	95004	1	5	
8	95004	2	0	

17. 查询各系的男女生学生总数, 并按系升序排列, 女生排在前

```

1 select sdept, ssex, count(*) as num
2 from student
3 group by sdept, ssex
4 order by sdept, ssex;

```

	sdept	ssex	num
1	CS	女	1
2	CS	男	1
3	IS	女	1
4	IS	男	1
5	MA	女	1

18. 查询"信息系"(IS)学生"数据结构"课程的平均成绩

```

1 select avg(grade)
2 from sc,
3      student,
4      course
5 where student.sdept = 'IS'
6        and student.sno = sc.sno
7        and course.cname = '数据结构'
8        and course.cno = sc.cno;

```

原数据中没有给出此项成绩

	`avg(grade)`
1	<null>

19. 创建一个反映学生出生年份的视图

```
1 create view birth(sno, sname, sbirth)
2 as
3 select sno, sname, 2022 - sage
4 from student;
```

	sno	sname	sbirth
1	95001	李勇	2004
2	95002	刘晨	2003
3	95003	王敏	2004
4	95004	张立	2003
5	95005	刘云	2004

20. 查询与"王田"在同一个系学习的学生的信息

```
1 select *
2 from student
3 where sdept in (select sdept from student where sname = '王田');
```

原数据中没有给出此项数据

sno	sname	ssex	sage	sdept

21. 查询年龄在20岁以下的学生的姓名及其年龄

```
1 select sname, sage
2 from student
3 where sage < 20
```

	sname	sage
1	李勇	18
2	刘晨	19
3	王敏	18
4	张立	19
5	刘云	18

22. 查询当前至少选修数据库和信息系统其中一门课的学生的学号

```

1 select sno
2 from sc,
3     course
4 where course.cname = '数据库'
5       and course.cno = sc.cno
6 union
7 select sno
8 from sc,
9     course
10 where course.cname = '信息系统'
11       and course.cno = sc.cno;

```

	sno
1	95001
2	95004
3	95002

23. 查询每个学生的学号, 姓名, 选修的课程名和成绩

```

1 select student.sno, sname, cname, grade
2 from student,
3     course,
4     sc
5 where student.sno = sc.sno
6       and sc.cno = course.cno;

```

	sno	sname	cname	grade
1	95004	张立	数据库	5
2	95001	李勇	数据库	97
3	95004	张立	数学	0
4	95003	王敏	数学	85
5	95002	刘晨	数学	0
6	95001	李勇	数学	85
7	95002	刘晨	信息系统	0
8	95001	李勇	信息系统	88

24. 查找名字中包括“俊”的学生的姓名, 学号, 选课课程和成绩

```

1 select sname, student.sno, cname, grade
2 from student,
3     sc,
4     course
5 where sname like '%俊%';

```

没有相关姓名的学生

sname	sno	cname	grade
Select All			

25. 查询学分大于8的学生, 输出学生的学号和学分

```
1 select sno, sum(credit)
2 from sc,
3     course
4 where sc.cno = course.cno
5 group by sno
6 having sum(credit) > 8;
```

	sno	sum(credit)
1	95001	13
2	95002	9
3	95004	10

26. 查询IS,CS,MA系的所有学生的姓名和性别

```
1 select sname, ssex
2 from student
3 where sdept in ('IS', 'MA', 'CS');
```

	sname	ssex
1	李勇	男
2	刘晨	女
3	王敏	女
4	张立	男
5	刘云	女

27. 查询至少选修了2门课程的学生的平均成绩

```
1 select avg(grade)
2 from sc
3 group by sno
4 having count(*) >= 2;
```

	avg(grade)
1	90.0000
2	0.0000
3	2.5000

28. 查询每个学生所选课程的平均成绩, 最高分, 最低分, 和选课门数

```
1 select sno, avg(grade), max(grade), min(grade), count(*)
2 from sc
3 group by sno;
```

	sno	avg(grade)	max(grade)	min(grade)	count(*)
1	95001	90.0000	97	85	3
2	95002	0.0000	0	0	2
3	95003	85.0000	85	85	1
4	95004	2.5000	5	0	2

29. 删除年龄大于21岁所有学生的选课记录

```
1 delete
2 from sc
3 where sno in (select sno from student where sage > 21);
```

不存在大于21岁的学生

30. 查询没有先行课的课程的课程号cno和课程名cname

```
1 select cno, cname
2 from course
3 where pcno is null;
```

	cno	cname
1	2	数学
2	6	数据处理

31. 创建信息系学生信息的视图

```
1 create view is_info
2 as
3 select*
4 from student
5 where sdept = 'IS';
```

	sno	sname	ssex	sage	sdept
1	95002	刘晨	女	19	IS
2	95004	张立	男	19	IS

32. 在信息系的学生视图中查询年龄小于20岁的学生

```
1 select sno, sage
2 from is_info
3 where sage < 20;
```

	sno	sage
1	95002	19
2	95004	19

33. 删除马朝阳同学的所有选课记录

```
1 delete
2 from sc
3 where sno like (select sno from student where sname = '马朝阳');
```

34. 查询选修了3号课程的学生的学号和成绩, 并按分数降序排列


```

1 select sno, grade
2 from sc
3 where cno = 3
4 order by grade desc;

```

	sno	grade
1	95001	88
2	95002	0

35. 查询数据库课程成绩不及格的学生, 输入其学号, 姓名和成绩

```

1 select student.sno, sname, grade
2 from sc,
3      student,
4      course
5 where grade < 60
6      and sc.cno = course.cno
7      and cname = '数据库';

```

	sno	sname	grade
1	95001	李勇	5
2	95002	刘晨	5
3	95003	王敏	5
4	95004	张立	5
5	95005	刘云	5

36. 查询全体学生的情况, 查询结果按所在系号升序排列, 同一系中的学生按年龄降序排列

```

1 select *
2 from student
3 order by sdept, sage desc;

```

	sno	sname	ssex	sage	sdept
1	95001	李勇	男	18	CS
2	95005	刘云	女	18	CS
3	95002	刘晨	女	19	IS
4	95004	张立	男	19	IS
5	95003	王敏	女	18	MA

37. 查询每个学生及其选修课程的情况

```

1 select student.sno, sname, sc.cno, cname
2 from sc,
3      student,
4      course
5 where sc.cno = course.cno
6      and student.sno = sc.sno;

```

	sno	sname	cno	cname
1	95004	张立	1	数据库
2	95001	李勇	1	数据库
3	95004	张立	2	数学
4	95003	王敏	2	数学
5	95002	刘晨	2	数学
6	95001	李勇	2	数学
7	95002	刘晨	3	信息系统
8	95001	李勇	3	信息系统

38. 查询每一门课程的间接先行课

```

1 select c1.cname, c1.cno, c2.cname, c2.cno
2 from course c1,
3      course c2
4 where c1.pcno = c2.cno;

```

	c1.cname	c1.cno	c2.cname	c2.cno
1	信息系统	3	数据库	1
2	数据库	1	数据结构	5
3	PASCAL语言	7	数据处理	6
4	操作系统	4	数据处理	6
5	数据结构	5	PASCAL语言	7

39. 查询选修1号课程且成绩在85分以上的所有学生的学号、姓名

```

1 select sc.sno, sname
2 from sc,
3      student
4 where sc.cno = 1
5       and grade > 85
6       and sc.sno = student.sno;

```

	sno	sname
1	95001	李勇

40. 查询全体学生的所有信息

```

1 select sc.*, student.*, course.*
2 from student,
3      sc,
4      course
5 where student.sno=sc.sno
6       and sc.cno=course.cno;

```

	sc.sno	sc.cno	grade	student.sno	sname	ssex	sage	sdept	course.cno	cname	credit	pcno
1	95004	1		5 95004	张立	男	19	IS	1	数据库	4	5
2	95001	1		97 95001	李勇	男	18	CS	1	数据库	4	5
3	95004	2		0 95004	张立	男	19	IS	2	数学	6	<null>
4	95003	2		85 95003	王敏	女	18	MA	2	数学	6	<null>
5	95002	2		0 95002	刘晨	女	19	IS	2	数学	6	<null>
6	95001	2		85 95001	李勇	男	18	CS	2	数学	6	<null>
7	95002	3		0 95002	刘晨	女	19	IS	3	信息系统	3	1
8	95001	3		88 95001	李勇	男	18	CS	3	信息系统	3	1

41. 查询选修了课程'1'和课程'2'的学生的学号

```
1 select sno
2 from sc
3 where sno in
4     (select distinct sno
5      from sc
6      where cno = 1)
7 and cno = 2;
```

	sno
1	95001
2	95004

42. 创建信息系选修了1号课程的学生的视图

```
1 create view course_one
2 as
3 select student.*
4 from sc,
5      student
6 where cno = 1
7 and sc.sno = student.sno;
```

	sno	sname	ssex	sage	sdept
1	95001	李勇	男	18	CS
2	95004	张立	男	19	IS

43. 建立信息系选修了1号课程且成绩在90分以上的学生的视图

```
1 create view course_one_above_90
2 as
3 select student.*
4 from student,
5      sc
6 where cno = 1
7 and student.sno = sc.sno
8 and grade > 90;
```

	sno	sname	ssex	sage	sdept
1	95001	李勇	男	18	CS

44. 查询修课总学分在10学分以下的学生姓名

```
1 select sname
2 from student
3 where sno in (select sno
4              from sc,
5              course
6              where sc.cno = course.cno
7              group by sno
8              having sum(credit) < 10);
```

	sname
1	刘晨
2	王敏

45. 查询比'刘晨'年龄小的所有学生的信息

```

1 select*
2 from student
3 where sage < (select sage from student where sname = '刘晨');

```

	sno	sname	ssex	sage	sdept
1	95001	李勇	男	18	CS
2	95003	王敏	女	18	MA
3	95005	刘云	女	18	CS

46. 查询所有选修了2号课程的学生的姓名

```

1 select sname
2 from student,
3      sc
4 where cno = 2
5      and sc.sno = student.sno;

```

	sname
1	李勇
2	刘晨
3	王敏
4	张立

47. 查询其他系中比信息系(IS)某一学生年龄小的学生姓名和年龄

```

1 select sname, sage
2 from student
3 where sdept <> 'IS'
4      and sage < any (select sage from student where sdept = 'IS');

```

	sname	sage
1	李勇	18
2	王敏	18
3	刘云	18

48. 查询学生2号课程的成绩，并按照成绩由高到低输出

```

1 select grade
2 from sc
3 where cno = 2
4 order by grade desc;

```

	grade ↕
1	85
2	85
3	0
4	0

49. 查询考试成绩有不及格的学生的学号

```
1 select distinct sno
2 from sc
3 where grade < 60;
```

	sno ↕
1	95002
2	95004

50. 查询其他系中比信息系(IS)学生年龄都小的学生姓名和年龄

```
1 select sname, sage
2 from student
3 where sdept <> 'IS'
4 and sage < all (select sage from student where sdept = 'IS');
```

	sname ↕	sage ↕
1	李勇	18
2	王敏	18
3	刘云	18

51. 将所有学生的学号和他的平均成绩定义为一个视图

```
1 create view avg_grade(sno, avgGrade)
2 as
3 select sno, AVG(grade)
4 from sc
5 group by sno
```

	sno ↕	avgGrade ↕
1	95001	90.0000
2	95002	0.0000
3	95003	85.0000
4	95004	2.5000

52. 在视图 S_G 中查询平均成绩在 90 分以上的学生的学号和平均成绩

```
1 select *
2 from avg_grade
3 where avgGrade >= 90
```

	sno	avgGrade
1	95001	90.0000

53. 查询与计算机系(CS)系所有学生的年龄均不同的学生学号, 姓名和年龄

```

1 select distinct sno, sname, sage
2 from student
3 where sdept != 'CS'
4    and sage NOT IN (select sage
5                      from student
6                      where sdept = 'CS');
```

	sno	sname	sage
1	95002	刘晨	19
2	95004	张立	19

54. 查询信息系选修了1号课程的学生

```

1 select sc.sno, sname
2 from student,
3      sc
4 where sc.cno = 1
5    and sdept = 'IS'
6    and student.sno = sc.sno;
```

	sno	sname
1	95004	张立

55. 查询与其他所有学生年龄均不同的学生学号, 姓名和年龄

```

1 # 没有这样的学生, 修改一个数据以符合要求
2 update student
3 set sage=20
4 where sno = 95001;
5
6 select a.sno, a.sname, a.sage
7 from student a
8 where sage not in (select sage
9                   from student b
10                  where a.sno != b.sno);
```

	sno	sname	sage
1	95001	李勇	20

56. 查询选修了全部课程的学生姓名

```

1 select sname
2 from student,
3      sc
4 where student.sno = sc.sno
5    and sc.cno = all (select cno
6                      from course);
```

57. 求没有人选修的课程号 cno 和 cname

```
1 select cno, cname
2 from course
3 where cno not in (select cno
4                  from sc);
```

	cno	cname
1	4	操作系统
2	5	数据结构
3	6	数据处理
4	7	PASCAL语言

58. 查询满足条件的(sno,cno)对, 其中该学号的学生没有选修该课程号 cno 的课程

```
1 select sno, course.cno
2 from course,
3      sc
4 group by sc.sno, course.cno
5 having course.cno not in (select sc.cno from sc);
```

	sno	cno
1	95001	7
2	95001	6
3	95001	5
4	95001	4
5	95002	7
6	95002	6
7	95002	5
8	95002	4
9	95003	7
10	95003	6
11	95003	5
12	95003	4
13	95004	7
14	95004	6
15	95004	5
16	95004	4

59. 查询每个学生的课程成绩最高的成绩信息(sno,cno,grade)

```

1 select b.sno, b.cno, b.grade
2 from (select sno, max(grade) maxg
3       from sc
4       group by sno) a,
5       sc b
6 where a.sno = b.sno
7       and a.maxg = b.grade;

```

	sno	cno	grade
1	95001	1	97
2	95002	2	0
3	95002	3	0
4	95003	2	85
5	95004	1	5

60. 查询学生总人数

```

1 select COUNT(sno)
2 from student;

```

	COUNT(sno)
1	5

61. 查询年龄在20-30岁直接的学生的姓名, 姓名, 所在系

```

1 select sname, ssex, sdept
2 from student
3 where sage >= 20
4       and sage <= 30;

```

	sname	ssex	sdept
1	李勇	男	CS

62. 查询所有课程的总学分数和平均学分数,以及最高学分和最低学分

```

1 select SUM(credit) sum, avg(credit) avg, MAX(credit) max, min(credit) min
2 from course;

```

	sum	avg	max	min
1	28	4.0000	6	3

63. 求成绩低于该门课程平均成绩的学生的成绩信息(sno,cno,grade)

```

1 select sno, cno, grade
2 from sc
3 where grade < (select avg(a.grade)
4               from sc a,
5               sc b
6               where a.cno = b.cno);

```


	sno	cno	grade
1	95002	2	0
2	95002	3	0
3	95004	1	5
4	95004	2	0

64. 查询各系的学生的人数并按人数从多到少排序

```
1 select sdept, count(sno) count
2 from student
3 group by sdept
4 order by count desc;
```

	sdept	count
1	CS	2
2	IS	2
3	MA	1

65. 创建年龄大于23岁的学生的视图

```
1 create view view66 as
2 select sage
3 from student
4 where sage > 23;
```

	sdept	count
1	CS	2
2	IS	2
3	MA	1

66. 查询选修了课程的学生总数

```
1 select COUNT(sno)
2 from student
3 where sno in (select sno
4               from sc);
```

	`COUNT(sno)`
1	4

67. 查询选修了3门课程以上的学生的学号和姓名

```
1 select sno, sname
2 from student
3 where sno in
4       (select sno
5        from sc
6        group by sno
7        having count(*) >= 3);
```

	sno	sname
1	95001	李勇

68. 查询平均分超过80分的学生的学号和平均分

```
1 select sno, avg(grade) avg
2 from sc
3 group by sno
4 having avg(grade) > 80;
```

	sno	avg
1	95001	90.0000
2	95003	85.0000

69. 比较: 求各学生的60分以上课程的平均分

```
1 select sno, avg(grade) avg
2 from sc
3 where grade >= 60
4 group by sno;
```

	sno	avg
1	95001	90.0000
2	95003	85.0000

70. 查询"信息系"(IS)中选修了2门课程以上的学生的学号

```
1 select sc.sno
2 from student,
3      sc
4 where sdept = 'IS'
5      and student.sno = sc.sno
6 group by sc.sno
7 having COUNT(sc.sno) >= 2;
```

	sno
1	95002
2	95004

71. 查询选修了1号课程或2号课程的学生的学号

```
1 select distinct a.sno
2 from sc a,
3      sc b
4 where a.cno = 1
5      or b.cno = 2 and a.sno = b.sno;
```

	sno
1	95004
2	95001
3	95002
4	95003

72. 查询平均成绩少于70分的学生的学号

```
1 select sno
2 from sc
3 group by sno
4 having avg(grade) <= 70;
```

	sno
1	95002
2	95004

73. 将信息系学生视图IS_Student中学号为“95002”的学生姓名改为“刘辰”

```
1 create view IS_Student
2 as
3 select *
4 from student
5 where sdept='IS';
6
7 update student
8 set sname='刘辰'
9 where sno = '95002' and sdept='IS';
```

74. 向信息系学生视图IS_Student中插入一个新的学生记录, 学号为95029,姓名为“刘一梦”, 年龄为20岁

```
1 insert
2 into IS_Student
3 values ('95029', '刘一梦', '女', '20', 'IS');
```

75. 删除信息系学生视图IS_Student中学号为95004的学生的记录

```
1 delete
2 from IS_Student
3 where sno = 95001;
```

课程感悟与建议

- 关于命名规范
 - 字段命名规范, 建议采用 SnakeCase, 如 `user_name`
 - 表名以 `t_` 开头
 - 见名知意, 不使用缩写
- 考虑使用 MySQL,或讲解中介绍 MySQL

