

# 实验过程

姓名

刘青帅

班级

计算机91

学号

2191211634

注：SQL查询时，有的查询结果为空，因此对于为空的查询增加了一个extra

## H2 一、创建三个表

采用的表命名规则为  $S\times\times\times$ 、 $C\times\times\times$ 、 $SC\times\times\times$ ，其中  $\times\times\times$  表示自己学号的后三位，如学号后三位为“001”的同学建立的各表名称分别为 S001, C001, SC001。

各表包含属性如下，请为各属性选择合理的数据类型：

$S\times\times\times$  (S#, SNAME, SEX, BDATE, HEIGHT, DORM)

$C\times\times\times$  (C#, CNAME, PERIOD, CREDIT, TEACHER)

$SC\times\times\times$  (S#, C#, GRADE) 其中 S#、C#均为外键

学生表

```
openGauss mydb
1 CREATE TABLE S634 (
2   SNO CHAR ( 10 ) NOT NULL,
3   SNAME VARCHAR ( 20 ) NOT NULL,
4   SEX CAHR ( 6 ) NOT NULL,
5   BDATE DATE NOT NULL,
6   HEIGHT DEC ( 3, 2 ) DEFAULT NULL,
7   DORM VARCHAR ( 30 ) NOT NULL,
8   PRIMARY KEY ( SNO )
9 )
```

课程表

```
openGauss mydb
1 CREATE TABLE C634 (
2   CNO CHAR ( 10 ) NOT NULL,
3   CNAME VARCHAR ( 40 ) NOT NULL,
4   PERIOD SMALLINT NOT NULL,
5   CREDIT DEC ( 2, 1 ) NOT NULL,
6   TEACHER VARCHAR ( 40 ) NOT NULL,
7   PRIMARY KEY ( CNO )
8 )
```

## 选课表

```
openGauss mydb public
1 CREATE TABLE SC634 (
2   SNO CHAR ( 10 ) NOT NULL,
3   CNO CHAR ( 10 ) NOT NULL,
4   GRADE DEC ( 4, 1 ) DEFAULT NULL,
5   PRIMARY KEY ( SNO, CNO ),
6   FOREIGN KEY ( SNO )
7     REFERENCES "S634"
8     ON DELETE CASCADE,
9   FOREIGN KEY ( CNO )
10    REFERENCES "C634"
11    ON DELETE RESTRICT,
12   CHECK ( ( GRADE IS NULL ) OR GRADE BETWEEN 0 AND 100 )
13 )
```

说明：删除学生表的时候，选课表的对应元组也要删除



```
1 FOREIGN KEY( SNO )
2   REFERENCES "S634"
3   ON DELETE CASCADE
```

说明：删除课程表的时候，选课表的对应元组禁止删除



```
1 FOREIGN KEY( CNO )
2   REFERENCES "C634"
3   ON DELETE RESTRICT,
```

建好的三个表



## H2

## 二、插入数据到表中

## 插入学生表

```
1 INSERT INTO S634 ( SNO, SNAME, SEX, BDATE, HEIGHT, DORM )
2 VALUES
3 ( '01032010', '王涛', '男', '2002-4-5', 1.72, '东14舍221' ),
4 ( '01032023', '孙文', '男', '2003-6-10', 1.80, '东14舍221' ),
5 ( '01032001', '张晓梅', '女', '2003-11-17', 1.58, '东1舍312' ),
6 ( '01032005', '刘静', '女', '2002-1-10', 1.63, '东1舍312' ),
7 ( '01032112', '董蔚', '男', '2002-2-20', 1.71, '东14舍221' ),
8 ( '03031011', '王倩', '女', '2003-12-20', 1.66, '东2舍104' ),
9 ( '03031014', '赵思扬', '男', '2001-6-6', 1.85, '东18舍421' ),
10 ( '03031051', '周剑', '男', '2001-5-8', 1.68, '东18舍422' ),
11 ( '03031009', '田婷', '女', '2002-8-11', 1.60, '东2舍104' ),
12 ( '03031033', '蔡明明', '男', '2002-3-12', 1.75, '东18舍423' ),
13 ( '03031056', '曹子衿', '女', '2003-12-15', 1.65, '东2舍305' )
```

| sno      | sname | sex | bdate               | height | dorm    |
|----------|-------|-----|---------------------|--------|---------|
| 01032010 | 王涛    | 男   | 2002-04-05 00:00:00 | 1.72   | 东14舍221 |
| 01032023 | 孙文    | 男   | 2003-06-10 00:00:00 | 1.80   | 东14舍221 |
| 01032001 | 张晓梅   | 女   | 2003-11-17 00:00:00 | 1.58   | 东1舍312  |
| 01032005 | 刘静    | 女   | 2002-01-10 00:00:00 | 1.63   | 东1舍312  |
| 01032112 | 董蔚    | 男   | 2002-02-20 00:00:00 | 1.71   | 东14舍221 |
| 03031011 | 王倩    | 女   | 2003-12-20 00:00:00 | 1.66   | 东2舍104  |
| 03031014 | 赵思扬   | 男   | 2001-06-06 00:00:00 | 1.85   | 东18舍421 |
| 03031051 | 周剑    | 男   | 2001-05-08 00:00:00 | 1.68   | 东18舍422 |
| 03031009 | 田婷    | 女   | 2002-08-11 00:00:00 | 1.60   | 东2舍104  |
| 03031033 | 蔡明明   | 男   | 2002-03-12 00:00:00 | 1.75   | 东18舍423 |
| 03031056 | 曹子衿   | 女   | 2003-12-15 00:00:00 | 1.65   | 东2舍305  |

## 插入课程表和结果

```
1 INSERT INTO C634 ( CNO, CNAME, PERIOD, CREDIT, TEACHER )
2 VALUES
3 ( 'CS-01', '数据结构', 60, 3, '张军' ),
4 ( 'CS-02', '计算机组成原理', 80, 4, '王亚伟' ),
5 ( 'CS-04', '人工智能', 40, 2, '李蕾' ),
6 ( 'CS-05', '深度学习', 40, 2, '崔均' ),
7 ( 'EE-01', '信号与系统', 60, 3, '张明' ),
8 ( 'EE-02', '数字逻辑电路', 100, 5, '胡海东' ),
9 ( 'EE-03', '光电子学与光子学', 40, 2, '石韬' )
```

| cno   | cname    | period | credit | teacher |
|-------|----------|--------|--------|---------|
| CS-01 | 数据结构     | 60     | 3.0    | 张军      |
| CS-02 | 计算机组成原理  | 80     | 4.0    | 王亚伟     |
| CS-04 | 人工智能     | 40     | 2.0    | 李蕾      |
| CS-05 | 深度学习     | 40     | 2.0    | 崔均      |
| EE-01 | 信号与系统    | 60     | 3.0    | 张明      |
| EE-02 | 数字逻辑电路   | 100    | 5.0    | 胡海东     |
| EE-03 | 光电子学与光子学 | 40     | 2.0    | 石韬      |

## 插入成绩表和结果

```

1  INSERT INTO SC634 ( SNO, CNO, GRADE )
2  VALUES
3  ( '01032010', 'CS-01', 82.0 ),
4  ( '01032010', 'CS-02', 91.0 ),
5  ( '01032010', 'CS-04', 83.5 ),
6  ( '01032001', 'CS-01', 77.5 ),
7  ( '01032001', 'CS-02', 85.0 ),
8  ( '01032001', 'CS-04', 83.0 ),
9  ( '01032005', 'CS-01', 62.0 ),
10 ( '01032005', 'CS-02', 77.0 ),
11 ( '01032005', 'CS-04', 82.0 ),
12 ( '01032023', 'CS-01', 55.0 ),
13 ( '01032023', 'CS-02', 81.0 ),
14 ( '01032023', 'CS-04', 76.0 ),
15 ( '01032112', 'CS-01', 88.0 ),
16 ( '01032112', 'CS-02', 91.5 ),
17 ( '01032112', 'CS-04', 86.0 ),
18 ( '01032112', 'CS-05', NULL ),
19 ( '03031033', 'EE-01', 93.0 )

```

| sno     | cno   | grade  |
|---------|-------|--------|
| 0103201 | CS-01 | 82.0   |
| 0103201 | CS-02 | 91.0   |
| 0103201 | CS-04 | 83.5   |
| 0103200 | CS-01 | 77.5   |
| 0103200 | CS-02 | 85.0   |
| 0103200 | CS-04 | 83.0   |
| 0103200 | CS-01 | 62.0   |
| 0103200 | CS-02 | 77.0   |
| 0103200 | CS-04 | 82.0   |
| 0103202 | CS-01 | 55.0   |
| 0103202 | CS-02 | 81.0   |
| 0103202 | CS-04 | 76.0   |
| 0103211 | CS-01 | 88.0   |
| 0103211 | CS-02 | 91.5   |
| 0103211 | CS-04 | 86.0   |
| 0103211 | CS-05 | (Null) |
| 0303103 | EE-01 | 93.0   |
| 0303103 | EE-02 | 89.0   |

## H2 三、完成SQL语句

### H3 1. 查询

(1) 查询电子工程系（EE）所开课程的课程编号、课程名称及学分数。

```

1  SELECT CNO,CNAME,CREDIT
2  FROM c634
3  WHERE CNO LIKE 'EE-%'

```

```

1  SELECT CNO,CNAME,CREDIT
2  FROM c634
3  WHERE CNO LIKE 'EE-%'

```

| 信息    | 结果 1     |        |
|-------|----------|--------|
| cno   | cname    | credit |
| EE-01 | 信号与系统    | 3.0    |
| EE-02 | 数字逻辑电路   | 5.0    |
| EE-03 | 光电子学与光子学 | 2.0    |

(2) 查询未选课程“CS-01”的女生学号及其已选各课程编号、成绩。

```
1 SELECT SNO,CNO, GRADE
2 FROM sc634
3 JOIN s634 USING(SNO)
4 WHERE SNO NOT IN(
5     SELECT SNO
6     FROM sc634
7     WHERE CNO = 'CS-01'
8 )
9 AND SEX='女'
```

```
1 SELECT SNO,CNO, GRADE
2 FROM sc634
3 JOIN s634 USING(SNO)
4 WHERE SNO NOT IN(
5     SELECT SNO
6     FROM sc634
7     WHERE CNO = 'CS-01'
8 )
9 AND SEX='女'
```

| 信息       | 结果 1  |       |
|----------|-------|-------|
| sno      | cno   | grade |
| 03031009 | EE-01 | 88.0  |
| 03031009 | EE-02 | 78.5  |
| 03031011 | EE-01 | 91.0  |
| 03031011 | EE-02 | 86.0  |

(3) 查询2000年~2001年出生的学生的基本信息。

```
1 SELECT *
2 FROM s634
3 WHERE BDATE BETWEEN '2000-01-01' AND '2000-12-31'
```

```
1 SELECT *
2 FROM s634
3 WHERE BDATE BETWEEN '2000-01-01' AND '2000-12-31'
```

| 信息 |       | 结果 1  |       |       |        |       |
|----|-------|-------|-------|-------|--------|-------|
|    | sno   | sname | sex   | bdate | height | dorm  |
| ▶  | (N/A) | (N/A) | (N/A) | (N/A) | (N/A)  | (N/A) |

extra: 查询2000年~2003年出生的学生的基本信息

```
1 SELECT *
2 FROM s634
3 WHERE BDATE BETWEEN '2000-01-01' AND '2002-12-31'
```

| 信息       | 结果 1  |     |                     |        |         |
|----------|-------|-----|---------------------|--------|---------|
| sno      | sname | sex | bdate               | height | dorm    |
| 01032010 | 王涛    | 男   | 2002-04-05 00:00:00 | 1.72   | 东14舍221 |
| 01032005 | 刘静    | 女   | 2002-01-10 00:00:00 | 1.63   | 东1舍312  |
| 01032112 | 董蔚    | 男   | 2002-02-20 00:00:00 | 1.71   | 东14舍221 |
| 03031014 | 赵思扬   | 男   | 2001-06-06 00:00:00 | 1.85   | 东18舍421 |
| 03031051 | 周剑    | 男   | 2001-05-08 00:00:00 | 1.68   | 东18舍422 |
| 03031009 | 田婷    | 女   | 2002-08-11 00:00:00 | 1.60   | 东2舍104  |
| 03031033 | 蔡明明   | 男   | 2002-03-12 00:00:00 | 1.75   | 东18舍423 |

(4) 查询每位学生的学号、学生姓名及其已选修课程的学分总数。



```
1 SELECT s634.SNO,s634.SNAME,SUM(CREDIT) "总学分"
2 FROM sc634,s634,c634
3 WHERE sc634.SNO = s634.SNO AND sc634.CNO = c634.CNO
4 GROUP BY s634.SNO
```

```
1 SELECT s634.SNO,s634.SNAME,SUM(CREDIT) "总学分"
2 FROM sc634,s634,c634
3 WHERE sc634.SNO = s634.SNO AND sc634.CNO = c634.CNO
4 GROUP BY s634.SNO
```

| 信息       | 结果 1  |      |
|----------|-------|------|
| sno      | sname | 总学分  |
| 03031014 | 赵思扬   | 8.0  |
| 03031033 | 蔡明明   | 8.0  |
| 01032023 | 孙文    | 9.0  |
| 03031009 | 田婷    | 8.0  |
| 03031011 | 王倩    | 8.0  |
| 01032112 | 董蔚    | 11.0 |
| 03031051 | 周剑    | 8.0  |
| 01032001 | 张晓梅   | 9.0  |
| 01032010 | 王涛    | 9.0  |
| 01032005 | 刘静    | 9.0  |

(5) 查询选修课程“CS-02”的学生中成绩第二高的学生学号。

(虽然这个查询是错的, 如果成绩像99 99 98这样, 这个查询就是错误的。在 扩充数据 的 3.1-5 的 写法二 是正确的查询)

```
1 SELECT SNO
2 FROM sc634 sc1
3 WHERE sc1.cno = 'CS-02'
4     and sc1.grade = (
5         SELECT grade
6         FROM sc634 sc2
7         WHERE sc2.cno = 'CS-02'
8         ORDER BY grade DESC
9         LIMIT 1,1
10     )
```

```
1 SELECT SNO
2 FROM sc634 sc1
3 WHERE sc1.cno = 'CS-02'
4     and sc1.grade = (
5         SELECT grade
6         FROM sc634 sc2
7         WHERE sc2.cno = 'CS-02'
8         ORDER BY grade DESC
9         LIMIT 1,1
10     )
```

| 信息  | 结果 1     |
|-----|----------|
| sno | 01032010 |

### 补充limit函数用法

```
1 limit n //获得前n行
2 limit m,n //跳过第m行, 从第m+1行开始获取n行
```

(6) 查询平均成绩超过“王涛”同学的学生学号、姓名和平均成绩, 并按学号进行降序排列。

```

1 SELECT SNO,s634.SNAME,CAST(AVG(GRADE) AS NUMERIC(4,1) ) AS "平均成绩"
2 FROM sc634 JOIN s634 USING(SNO)
3 WHERE GRADE IS NOT NULL
4 GROUP BY SNO,s634.SNAME
5 HAVING AVG(GRADE) > ALL(
6     SELECT AVG(GRADE)
7     FROM sc634 JOIN s634 USING(SNO)
8     WHERE SNAME='王涛' AND GRADE IS NOT NULL
9 )
10 ORDER BY SNO DESC

```

说明：最开始我 GROUP BY 只写的 SNO ,报错，

报错如下：column "s634.sname" must appear in the GROUP BY clause or be used in an aggregate function

根据错误提示 GROUP BY 加了 SNAME （突然发现老师在28号说了这个）

```

1 SELECT SNO,s634.SNAME,CAST(AVG(GRADE) AS NUMERIC(4,1) ) AS "平均成绩"
2 FROM sc634 JOIN s634 USING(SNO)
3 WHERE GRADE IS NOT NULL
4 GROUP BY SNO,s634.SNAME
5 HAVING AVG(GRADE) > ALL(
6     SELECT AVG(GRADE)
7     FROM sc634 JOIN s634 USING(SNO)
8     WHERE SNAME='王涛' AND GRADE IS NOT NULL
9 )
10 ORDER BY SNO DESC

```

| sno      | sname | 平均成绩 |
|----------|-------|------|
| 03031033 | 蔡明明   | 91.0 |
| 03031011 | 王倩    | 88.5 |
| 01032112 | 董蔚    | 88.5 |

补充CAST函数用法

```

1 SQL数据库中Numeric(4,1)是指：字段是数字型，长度为4，小数为1位
2
3 CAST(表达式 as 数据类型) as 别名
4
5 demo: CAST( AVG(GRADE) AS NUMERIC(4,1) ) AS "平均成绩"

```

(7)查询选修了计算机专业全部课程（课程编号为“CS-xx”）的学生姓名及已获得的学分总数。



```

1 SELECT SNAME, SUM(CASE
2     WHEN GRADE BETWEEN 60 AND 100 THEN
3         CREDIT
4     ELSE
5         0
6 END ) "总学分"
7 FROM s634, sc634, c634
8 WHERE c634.CNO LIKE 'CS-%' AND
9        c634.cno = sc634.cno AND s634.sno = sc634.sno
10 GROUP BY s634.SNO
11 HAVING COUNT(*) = (
12     SELECT COUNT(CNO)
13     FROM c634
14     WHERE CNO LIKE 'CS-%'
15 )

```

```

1 SELECT SNAME, SUM(CASE
2     WHEN GRADE BETWEEN 60 AND 100 THEN
3         CREDIT
4     ELSE
5         0
6 END ) "总学分"
7 FROM s634, sc634, c634
8 WHERE c634.CNO LIKE 'CS-%' AND
9        c634.cno = sc634.cno AND s634.sno = sc634.sno
10 GROUP BY s634.SNO
11 HAVING COUNT(*) = (
12     SELECT COUNT(CNO)
13     FROM c634
14     WHERE CNO LIKE 'CS-%'
15 )
16

```

| 信息    | 结果 1  |
|-------|-------|
| sname | 总学分   |
| (N/A) | (N/A) |

| 信息    | 结果 1 |
|-------|------|
| sname | 总学分  |
| 王涛    | 9.0  |
| 张晓梅   | 9.0  |
| 刘静    | 9.0  |
| 董蔚    | 9.0  |
| 孙文    | 6.0  |

extra:将 COUNT(\*)= 改为 COUNT(\*) ≤

(8) 查询选修了3门以上课程（包括3门）的学生中平均成绩最高的同学学号及姓名。（由于AVG忽略NULL值，而不是将其作为"0"参与计算，因此AVG不用写条件）

```

1 SELECT s634.sno,s634.sname
2 FROM s634,sc634
3 WHERE s634.sno=sc634.sno AND GRADE IS NOT NULL
4 GROUP BY s634.sno
5 HAVING AVG(GRADE) = (
6     SELECT AVG(GRADE)
7     FROM s634,sc634
8     WHERE s634.sno = sc634.sno AND GRADE IS NOT NULL
9     GROUP BY s634.SNO
10    HAVING COUNT(*) ≥ 3
11    ORDER BY AVG(GRADE) DESC
12    LIMIT 1
13 )

```

```

1 SELECT s634.SNO,s634.SNAME
2 FROM s634,sc634
3 WHERE s634.sno = sc634.sno AND GRADE IS NOT NULL
4 GROUP BY s634.SNO
5 HAVING COUNT(*) >= 3
6 ORDER BY AVG(GRADE) DESC
7 LIMIT 1

```

信息

结果 1

| sno        | sname |
|------------|-------|
| ▶ 01032112 | 董蔚    |

## H3 2.插入

分别在S和C表中加入记录（'01032005','刘竞','男','1993-12-10',1.75,'东14舍312'）及（'CS-03','离散数学',64,4,'陈建明'）

插入学生表失败，因为SNO为KEY，KEY已经在学生表中存在

```

1 INSERT INTO s634
2 VALUES('01032005','刘竞','男','1993-12-10',1.75,'东14舍312')

```

信息

```

INSERT INTO s634
VALUES('01032005','刘竞','男','1993-12-10',1.75,'东14舍312')
> ERROR: duplicate key value violates unique constraint "js634_pkey"
DETAIL: Key (sno)=(01032005 ) already exists.

> 时间: 0.027s

```

插入课程表然后查看课程表，插入成功

```
1 INSERT INTO c634
2 VALUES('CS-03','离散数学',64,4,'陈建明')
```

信息

```
INSERT INTO c634
VALUES('CS-03','离散数学',64,4,'陈建明')
> Affected rows: 1
> 时间: 0.024s
```

| cno   | cname    | period | credit | teacher |
|-------|----------|--------|--------|---------|
| CS-01 | 数据结构     | 60     | 3.0    | 张军      |
| CS-02 | 计算机组成原理  | 80     | 4.0    | 王亚伟     |
| CS-04 | 人工智能     | 40     | 2.0    | 李蕾      |
| CS-05 | 深度学习     | 40     | 2.0    | 崔均      |
| EE-02 | 数字逻辑电路   | 100    | 5.0    | 胡海东     |
| EE-03 | 光电子学与光子学 | 40     | 2.0    | 石韬      |
| EE-01 | 信号与系统    | 60     | 3.0    | 张明      |

### H3 3.删除

将S表中已修学分数大于60的学生记录删除

```
1 DELETE FROM s634
2 WHERE SNO IN(
3     SELECT SNO
4     FROM sc634,c634
5     WHERE sc634.cno = c634.cno
6     GROUP BY SNO
7     HAVING SUM(CREDIT) > 60
8 )
```

```

1 DELETE FROM s634
2 WHERE SNO IN(
3     SELECT SNO
4     FROM sc634,c634
5     WHERE sc634.cno = c634.cno
6     GROUP BY SNO
7     HAVING SUM(CREDIT) > 60
8 )

```

信息

```

DELETE FROM s634
WHERE SNO IN(

                                SELECT SNO
                                FROM sc634,c634
                                WHERE sc634.cno = c634.cno
                                GROUP BY SNO
                                HAVING SUM(CREDIT) > 60

                                )

> Affected rows: 0
> 时间: 0.026s

```

### H3 4.修改

将“张明”老师负责的“信号与系统”课程的学时数调整为64，同时增加一个学分。



```

1 UPDATE c634
2 SET CREDIT = CREDIT+1 , PERIOD = 64
3 WHERE CNAME='信号与系统' AND teacher='张明'

```

```

1 UPDATE c634
2 SET CREDIT = CREDIT+1 , PERIOD = 64
3 WHERE CNAME='信号与系统' AND teacher='张明'

```

信息

```

UPDATE c634
SET CREDIT = CREDIT+1 , PERIOD = 64
WHERE CNAME='信号与系统' AND teacher='张明'
> Affected rows: 1
> 时间: 0.027s

```

| cno   | cname    | period | credit | teacher |
|-------|----------|--------|--------|---------|
| CS-01 | 数据结构     | 60     | 3.0    | 张军      |
| CS-02 | 计算机组成原理  | 80     | 4.0    | 王亚伟     |
| CS-04 | 人工智能     | 40     | 2.0    | 李蕾      |
| CS-05 | 深度学习     | 40     | 2.0    | 崔均      |
| EE-02 | 数字逻辑电路   | 100    | 5.0    | 胡海东     |
| EE-03 | 光电子学与光子学 | 40     | 2.0    | 石韬      |
| CS-03 | 离散数学     | 64     | 4.0    | 陈建明     |
| EE-01 | 信号与系统    | 64     | 4.0    | 张明      |

## H3

## 5.建立视图

(1)居住在“东18舍”的男生视图，包括学号、姓名、出生日期、身高等属性。

```
1 CREATE VIEW DORM_EAST_MALE_18
2 AS SELECT SNO,SNAME,BDATE,HEIGHT
3 FROM s634
4 WHERE SEX='男' AND DORM LIKE '东18舍%'
```

```
1 CREATE VIEW DORM_EAST_MALE_18
2 AS SELECT SNO,SNAME,BDATE,HEIGHT
3 FROM s634
4 WHERE SEX='男' AND DORM LIKE '东18舍%'
```

信息

```
CREATE VIEW DORM_EAST_MALE_18
AS SELECT SNO,SNAME,BDATE,HEIGHT
FROM s634
WHERE SEX='男' AND DORM LIKE '东18舍%'
```

> OK

> 时间: 0.03s

| sno      | sname | bdate               | height |
|----------|-------|---------------------|--------|
| 03031014 | 赵思扬   | 2001-06-06 00:00:00 | 1.85   |
| 03031051 | 周剑    | 2001-05-08 00:00:00 | 1.68   |
| 03031033 | 蔡明明   | 2002-03-12 00:00:00 | 1.75   |

(2)“张明”老师所开设课程情况的视图，包括课程编号、课程名称、平均成绩等属性。

```
1 CREATE VIEW COURSE_ZM
2 AS SELECT c634.CNO,CNAME,CAST( AVG(GRADE) AS DEC(5,2) ) AS "平均成绩"
3 FROM c634,sc634
4 WHERE TEACHER='张明' AND c634.cno = sc634.cno
5 GROUP BY c634.CNO
```

```

1 CREATE VIEW COURSE_ZM
2 AS SELECT c634.CNO,CNAME,CAST( AVG(GRADE) AS DEC(5,2) ) AS "平均成绩"
3 FROM c634,sc634
4 WHERE TEACHER='张明' AND c634.cno = sc634.cno
5 GROUP BY c634.CNO

```

信息

```

CREATE VIEW COURSE_ZM
AS SELECT c634.CNO,CNAME,CAST( AVG(GRADE) AS DEC(5,2) ) AS "平均成绩"
FROM c634,sc634
WHERE TEACHER='张明' AND c634.cno = sc634.cno
GROUP BY c634.CNO

```

> OK

> 时间: 0.026s

| cno   | cname | 平均成绩  |
|-------|-------|-------|
| EE-01 | 信号与系统 | 85.80 |

(3)所有选修了“人工智能”课程的学生视图，包括学号、姓名、成绩等属性。

```

1 CREATE VIEW AI
2 AS SELECT s634.SNO,SNAME,GRADE
3 FROM sc634,s634,c634
4 WHERE CNAME='人工智能' AND s634.sno = sc634.sno
5 AND c634.cno = sc634.cno

```

```

1 CREATE VIEW AI
2 AS SELECT s634.SNO,SNAME,GRADE
3 FROM sc634,s634,c634
4 WHERE CNAME='人工智能' AND s634.sno = sc634.sno
5 AND c634.cno = sc634.cno

```

信息

```

CREATE VIEW AI
AS SELECT s634.SNO,SNAME,GRADE
FROM sc634,s634,c634
WHERE CNAME='人工智能' AND s634.sno = sc634.sno
AND c634.cno = sc634.cno

```

> OK

> 时间: 0.044s

| sno      | sname | grade |
|----------|-------|-------|
| 01032010 | 王涛    | 83.5  |
| 01032001 | 张晓梅   | 83.0  |
| 01032005 | 刘静    | 82.0  |
| 01032023 | 孙文    | 76.0  |
| 01032112 | 董蔚    | 86.0  |

## H2 四、扩充数据

## H3 1. 插入少量数据

在S表中补充数据至约1000行，在C表中补充数据至约100行，在SC表中补充数据至约5000行，尝试为三、1.中的（5）-（8）查询编写不同的SQL语句实现，并分析其运行效率。

S634表有1011行，C634表有108行，SC634表有5026行，

| sno      | sname | sex | bdate               | height | dorm    |
|----------|-------|-----|---------------------|--------|---------|
| 01032010 | 王涛    | 男   | 2002-04-05 00:00:00 | 1.72   | 东14舍221 |
| 01032023 | 孙文    | 男   | 2003-06-10 00:00:00 | 1.80   | 东14舍221 |
| 01032001 | 张晓梅   | 女   | 2003-11-17 00:00:00 | 1.58   | 东1舍312  |
| 01032005 | 刘静    | 女   | 2002-01-10 00:00:00 | 1.63   | 东1舍312  |
| 01032112 | 董蔚    | 男   | 2002-02-20 00:00:00 | 1.71   | 东14舍221 |
| 03031011 | 王伟    | 女   | 2003-12-20 00:00:00 | 1.66   | 东2舍104  |
| 03031014 | 赵思扬   | 男   | 2001-06-06 00:00:00 | 1.85   | 东18舍421 |
| 03031051 | 周剑    | 男   | 2001-05-08 00:00:00 | 1.68   | 东18舍422 |
| 03031009 | 田婷    | 女   | 2002-08-11 00:00:00 | 1.60   | 东2舍104  |
| 03031033 | 蔡明明   | 男   | 2002-03-12 00:00:00 | 1.75   | 东18舍423 |

**s634**  
表  
OID  
16403  
所有者  
lqs  
行  
1,011  
表类型  
常规

| cno   | cname    | period | credit | teacher |
|-------|----------|--------|--------|---------|
| CS-01 | 数据结构     | 60     | 3.0    | 张军      |
| CS-02 | 计算机组成原理  | 80     | 4.0    | 王亚伟     |
| CS-04 | 人工智能     | 40     | 2.0    | 李蕾      |
| CS-05 | 深度学习     | 40     | 2.0    | 崔均      |
| EE-02 | 数字逻辑电路   | 100    | 5.0    | 胡海东     |
| EE-03 | 光电子学与光子学 | 40     | 2.0    | 石韬      |
| CS-03 | 离散数学     | 64     | 4.0    | 陈建明     |
| EE-01 | 信号与系统    | 64     | 4.0    | 张明      |

**c634**  
表  
OID  
16409  
所有者  
lqs  
行  
108

| sno     | cno   | grade |
|---------|-------|-------|
| 0103201 | CS-01 | 82.0  |
| 0103201 | CS-02 | 91.0  |
| 0103201 | CS-04 | 83.5  |
| 0103200 | CS-01 | 77.5  |
| 0103200 | CS-02 | 85.0  |
| 0103200 | CS-04 | 83.0  |
| 0103200 | CS-01 | 62.0  |
| 0103200 | CS-02 | 77.0  |
| 0103200 | CS-04 | 82.0  |
| 0103202 | CS-01 | 55.0  |

**sc634**  
表  
OID  
16418  
所有者  
lqs  
行  
5,026  
表类型  
常规

### 3.1-5

查询选修课程“CS-02”的学生中成绩第二高的学生学号。（我自己加的数据中SC表没有 'CS-02'，所以查询结果不变，就不贴图了）

## 写法1

- 可以看出，主要花费在两次 seq Scan (顺序扫描)上，也就是扫描SC表，排序花费和取Limit时间几乎可以忽略不计
- 总花费为288

```
1 SELECT SNO
2 FROM sc634 sc1
3 WHERE sc1.cno = 'CS-02'
4     and sc1.grade = (
5         SELECT grade
6         FROM sc634 sc2
7         WHERE sc2.cno = 'CS-02'
8         ORDER BY grade DESC
9         LIMIT 1,1
10    )
```

### QUERY PLAN

```
► Seq Scan on sc634 sc1 (cost=138.23..288.62 rows=1 width=11)
  Filter: ((cno = 'CS-02'::bpchar) AND (grade = $0))
  InitPlan 1 (returns $0)
    -> Limit (cost=138.23..138.23 rows=1 width=5)
        -> Sort (cost=138.22..138.32 rows=40 width=5)
            Sort Key: sc2.grade DESC
            -> Seq Scan on sc634 sc2 (cost=0.00..137.82 rows=40 width=5)
                Filter: (cno = 'CS-02'::bpchar)
```

### #1 Limit

### #2 Sort

by grade DESC

### #3 Seq Scan

on sc634

### #3 Seq Scan

on sc634

**Seq Scan Node** finds relevant records by sequentially scanning the input record set. When reading from a table, Seq Scans (unlike Index Scans) perform a single read operation (only the table is read).

General IO & Buffers Output Workers

Misc

⌚ Timing: 1.32ms | 96%

☰ Rows: 5 (Planned: 40) | ⬆ over estimated by 8x

🗑 \*Rows Removed by Filter: 5,021 | 99%

\$ Cost: 138 (Total: 138)



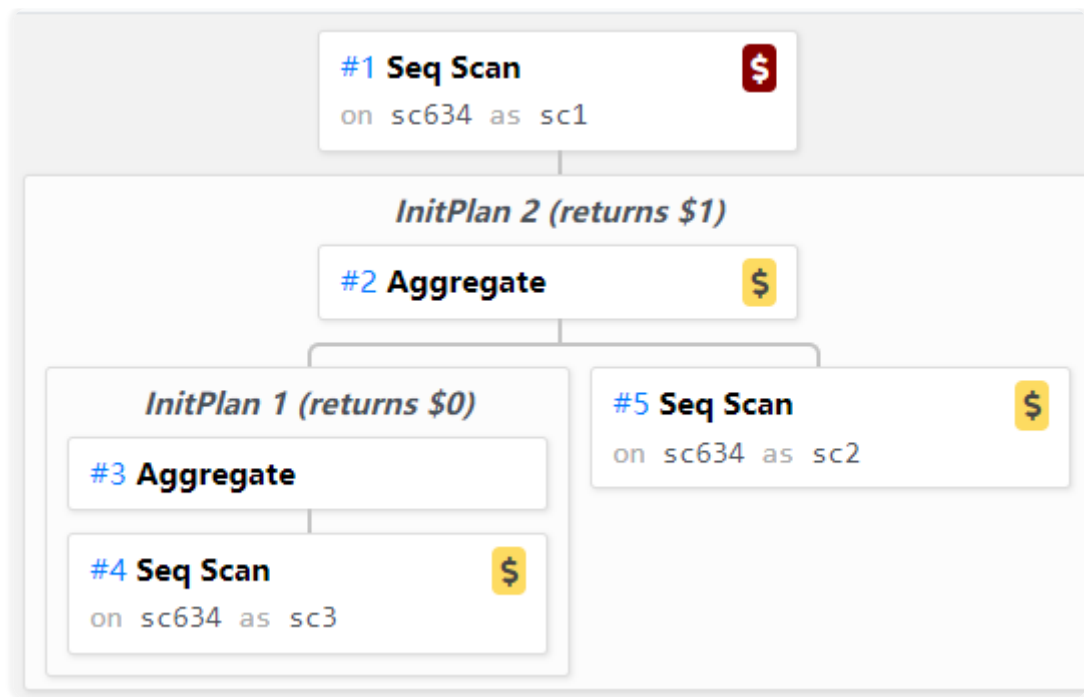
## 写法2

- 可以看出，主要花费在三次 seq Scan (顺序扫描)上
- 总花费为439

```
1 SELECT sc1.sno
2 FROM sc634 sc1
3 WHERE sc1.cno = 'CS-02'
4     AND sc1.grade = (
5         SELECT MAX( GRADE )
6         FROM sc634 sc2
7         WHERE sc2.cno = 'CS-02'
8             AND sc2.grade < (
9                 SELECT MAX ( GRADE )
10                FROM SC634 sc3
11                WHERE sc3.cno = 'CS-02'
12            )
13     )
```

### QUERY PLAN

|  |
|--|
| Seq Scan on sc634 sc1 (cost=288.37..438.76 rows=1 width=11)  |
| Filter: ((cno = 'CS-02'::bpchar) AND (grade = \$1))          |
| InitPlan 2 (returns \$1)                                     |
| ▶ -> Aggregate (cost=288.36..288.37 rows=1 width=37)         |
| InitPlan 1 (returns \$0)                                     |
| -> Aggregate (cost=137.92..137.93 rows=1 width=37)           |
| -> Seq Scan on sc634 sc3 (cost=0.00..137.82 rows=40 width=5) |
| Filter: (cno = 'CS-02'::bpchar)                              |
| -> Seq Scan on sc634 sc2 (cost=0.00..150.39 rows=13 width=5) |
| Filter: ((grade < \$0) AND (cno = 'CS-02'::bpchar))          |



对比

这两个主要差别在 `Seq Scan` 的次数上，嵌套的sql查询少的花费少(顺序扫描的次数少)

### 3.1-6

查询平均成绩超过"王涛"同学的学生学号、姓名和平均成绩，并按学号进行降序排列。（结果是一样的，这里就不在两个写法分别贴了）

| sno      | sname | 平均成绩 |
|----------|-------|------|
| 99337653 | 郑铁林   | 82.2 |
| 99031030 | 赵斌    | 80.0 |
| 98961920 | 阮小娥   | 89.3 |
| 98440495 | 陈秀秀   | 88.0 |
| 98421476 | 赵亚宁   | 85.0 |
| 98317707 | 高雅    | 91.0 |
| 97335425 | 王春霞   | 81.6 |
| 96339682 | 霍庆华   | 90.6 |
| 96305987 | 王博    | 90.0 |
| 96231120 | 赵鑫    | 83.2 |
| 95921424 | 何晓琳   | 81.2 |
| 95414336 | 刘艳婕   | 79.3 |
| 95406789 | 谢海鹏   | 78.8 |
| 95402246 | 周社教   | 88.8 |
| 95195818 | 范昊天   | 81.1 |
| 94676703 | 何健美   | 78.4 |
| 94667977 | 丁向东   | 78.5 |
| 94220358 | 陈东朗   | 83.3 |
| 94215229 | 戴慧琤   | 82.8 |
| 94071220 | 秦军    | 82.5 |
| 93660891 | 李月英   | 80.4 |
| 93538122 | 王朝霖   | 78.4 |
| 93497465 | 李倩倩   | 78.1 |

## 写法1

```

1  SELECT SNO,s634.SNAME,CAST(AVG(GRADE) AS NUMERIC(4,1) ) AS "平均成绩"
2  FROM sc634 JOIN s634 USING(SNO)
3  WHERE GRADE IS NOT NULL
4  GROUP BY SNO,s634.SNAME
5  HAVING AVG(GRADE) > ALL(
6      SELECT AVG(GRADE)
7      FROM sc634 JOIN s634 USING(SNO)
8      WHERE SNAME='王涛' AND GRADE IS NOT NULL
9  )
10 ORDER BY SNO DESC

```

信息 解释 1 结果 1

#### QUERY PLAN

GroupAggregate (cost=627.74..835.02 rows=5025 width=89)

Group By Key: public.sc634.sno, public.s634.sname

Filter: (SubPlan 1)

-> Sort (cost=550.01..562.57 rows=5025 width=25)

Sort Key: public.sc634.sno DESC, public.s634.sname

-> Hash Join (cost=46.75..241.10 rows=5025 width=25)

Hash Cond: (public.sc634.sno = public.s634.sno)

-> Seq Scan on sc634 (cost=0.00..125.26 rows=5025 width=16)

Filter: (grade IS NOT NULL)

-> Hash (cost=34.11..34.11 rows=1011 width=20)

▶ -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=20)

SubPlan 1

-> Materialize (cost=77.73..77.74 rows=1 width=37)

-> Aggregate (cost=77.73..77.74 rows=1 width=37)

-> Nested Loop (cost=4.29..77.70 rows=10 width=5)

-> Seq Scan on s634 (cost=0.00..36.64 rows=2 width=11)

Filter: ((sname)::text = '王涛'::text)

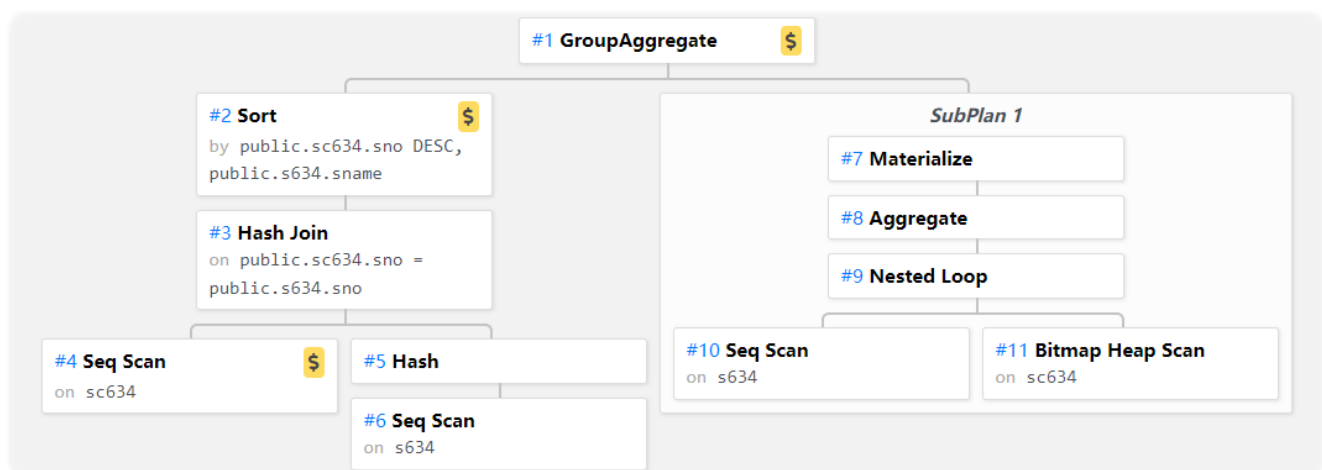
-> Bitmap Heap Scan on sc634 (cost=4.29..20.48 rows=5 width=16)

Recheck Cond: (sno = public.s634.sno)

Filter: (grade IS NOT NULL)

-> Bitmap Index Scan on jsc634\_pkey (cost=0.00..4.29 rows=5 width=0)

Index Cond: (sno = public.s634.sno)



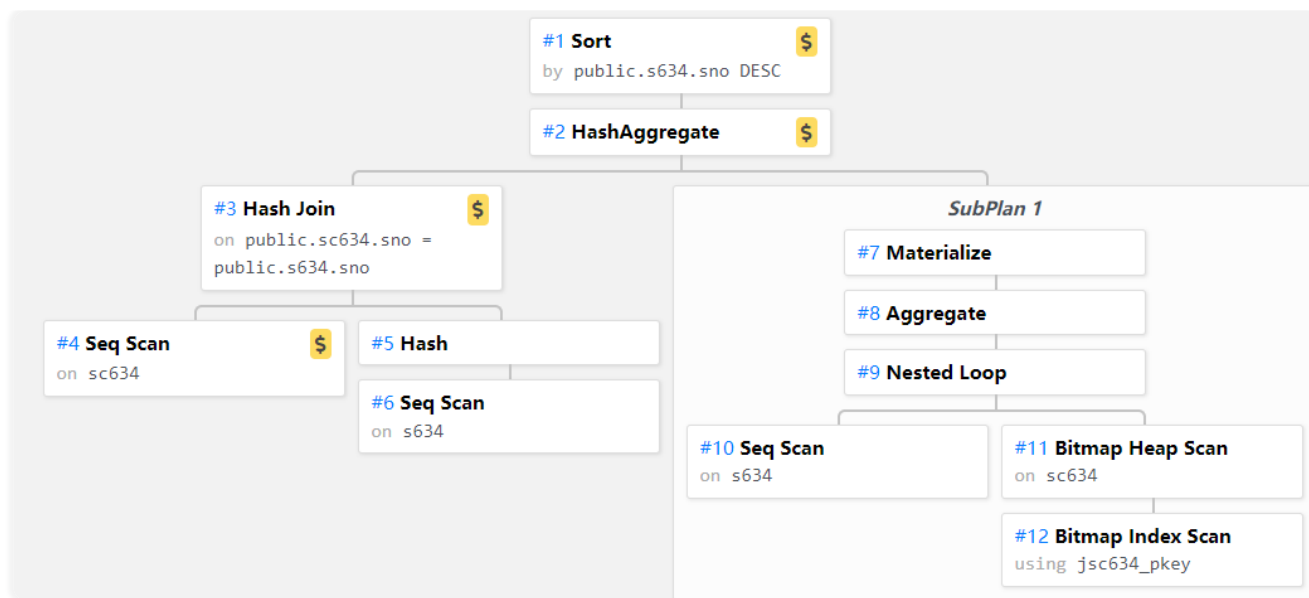
写法2

```

1 SELECT s634.sno,s634.SNAME,CAST(AVG(GRADE) AS NUMERIC(4,1) ) AS "平均成绩"
2 FROM sc634,s634
3 WHERE GRADE IS NOT NULL AND sc634.sno=s634.sno
4 GROUP BY s634.sno,s634.SNAME
5 HAVING AVG(GRADE) > ALL(
6     SELECT AVG(GRADE)
7     FROM sc634,s634
8     WHERE SNAME='王涛' AND GRADE IS NOT NULL AND sc634.sno=s634.sno
9 )
10 ORDER BY SNO DESC

```

| QUERY PLAN   |
|--|
| ► Sort (cost=448.60..451.13 rows=1011 width=89)                      |
| Sort Key: public.s634.sno DESC                                       |
| -> HashAggregate (cost=369.08..398.14 rows=1011 width=89)            |
| Group By Key: public.s634.sno, public.s634.sname                     |
| Filter: (SubPlan 1)  |
| -> Hash Join (cost=46.75..241.10 rows=5025 width=25)                 |
| Hash Cond: (public.sc634.sno = public.s634.sno)                      |
| -> Seq Scan on sc634 (cost=0.00..125.26 rows=5025 width=16)          |
| Filter: (grade IS NOT NULL)  |
| -> Hash (cost=34.11..34.11 rows=1011 width=20)                       |
| -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=20)            |
| SubPlan 1  |
| -> Materialize (cost=77.73..77.74 rows=1 width=37)                   |
| -> Aggregate (cost=77.73..77.74 rows=1 width=37)                     |
| -> Nested Loop (cost=4.29..77.70 rows=10 width=5)                    |
| -> Seq Scan on s634 (cost=0.00..36.64 rows=2 width=11)               |
| Filter: ((sname)::text = '王涛'::text)                                 |
| -> Bitmap Heap Scan on sc634 (cost=4.29..20.48 rows=5 width=16)      |
| Recheck Cond: (sno = public.s634.sno)                                |
| Filter: (grade IS NOT NULL)  |
| -> Bitmap Index Scan on jsc634_pkey (cost=0.00..4.29 rows=5 width=0) |
| Index Cond: (sno = public.s634.sno)                                  |



对比

这两个查询的差别就是 写法1 是 `sc634 JOIN s634 USING(SNO)` ,而 写法2 是 `sc634.sno=s634.sno` ,导致后面的优化不同.

写法1采用 `GroupAggregate`(花费将近200) ( `GroupAggregate` 需要提前sort)

写法2采用 `HashAggregate`(花费将近20)

网上说法: `HashAggregate` 特点是不需要进行排序,在组数值比较小的情况下是比 `GroupAggregate` 要快很多,但是需求的内存会比较多。

## 3.1-7

查询选修了计算机专业全部课程(课程编号为“CS-xx”)的学生姓名及已获得的学分总数。

结果还是空的,因为初始数据所有人没有选择 `CS-05` 这门课,我自己生成的数据也没有

写法1

```

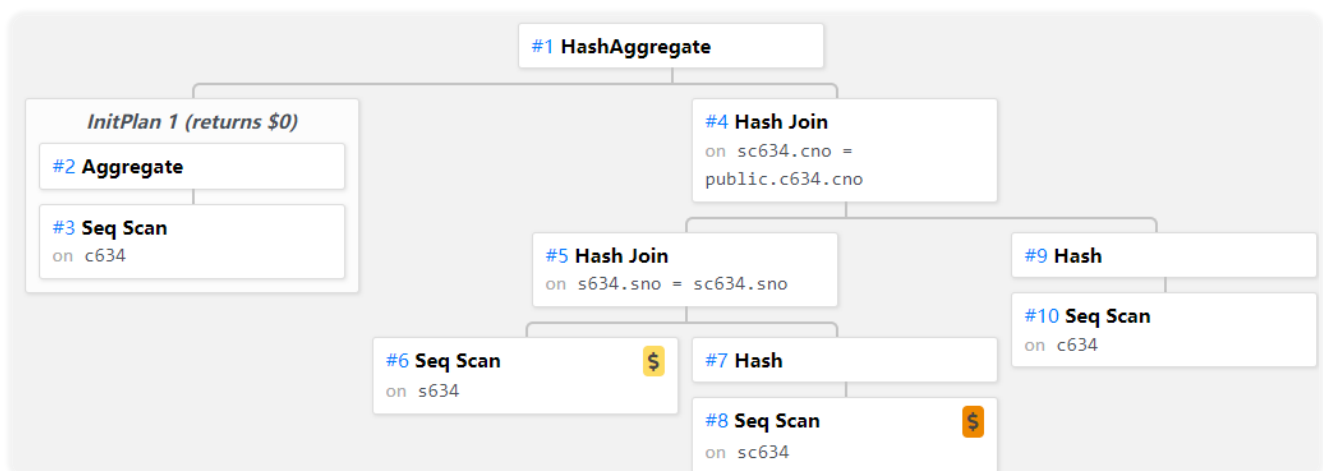
1 SELECT SNAME,SUM(CASE
2   WHEN GRADE BETWEEN 60 AND 100 THEN
3     CREDIT
4   ELSE
5     0
6 END ) "总学分"
7 FROM s634,sc634,c634
8 WHERE c634.CNO LIKE 'CS-%' AND
9       c634.cno = sc634.cno AND s634.sno = sc634.sno
  
```

```

10 GROUP BY s634.sno
11 HAVING COUNT(*) = (
12     SELECT COUNT(CNO)
13     FROM c634
14     WHERE CNO LIKE 'CS-%'
15 )

```

| QUERY PLAN  |
|---|
| ► HashAggregate (cost=185.96..186.48 rows=41 width=70)    |
| Group By Key: s634.sno                                    |
| Filter: (count(*) = \$0)                                  |
| InitPlan 1 (returns \$0)                                  |
| -> Aggregate (cost=4.36..4.37 rows=1 width=19)            |
| -> Seq Scan on c634 (cost=0.00..4.35 rows=5 width=11)     |
| Filter: (cno ~~ 'CS-%'::text)                             |
| -> Hash Join (cost=142.75..181.08 rows=41 width=30)       |
| Hash Cond: (sc634.cno = public.c634.cno)                  |
| -> Hash Join (cost=138.34..176.10 rows=41 width=36)       |
| Hash Cond: (s634.sno = sc634.sno)                         |
| -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=20) |
| -> Hash (cost=137.82..137.82 rows=41 width=27)            |
| -> Seq Scan on sc634 (cost=0.00..137.82 rows=41 width=27) |
| Filter: (cno ~~ 'CS-%'::text)                             |
| -> Hash (cost=4.35..4.35 rows=5 width=16)                 |
| -> Seq Scan on c634 (cost=0.00..4.35 rows=5 width=16)     |
| Filter: (cno ~~ 'CS-%'::text)                             |



## 写法2

```

1 SELECT SNAME, SUM(CASE
2     WHEN GRADE BETWEEN 60 AND 100 THEN
3         CREDIT
4     ELSE

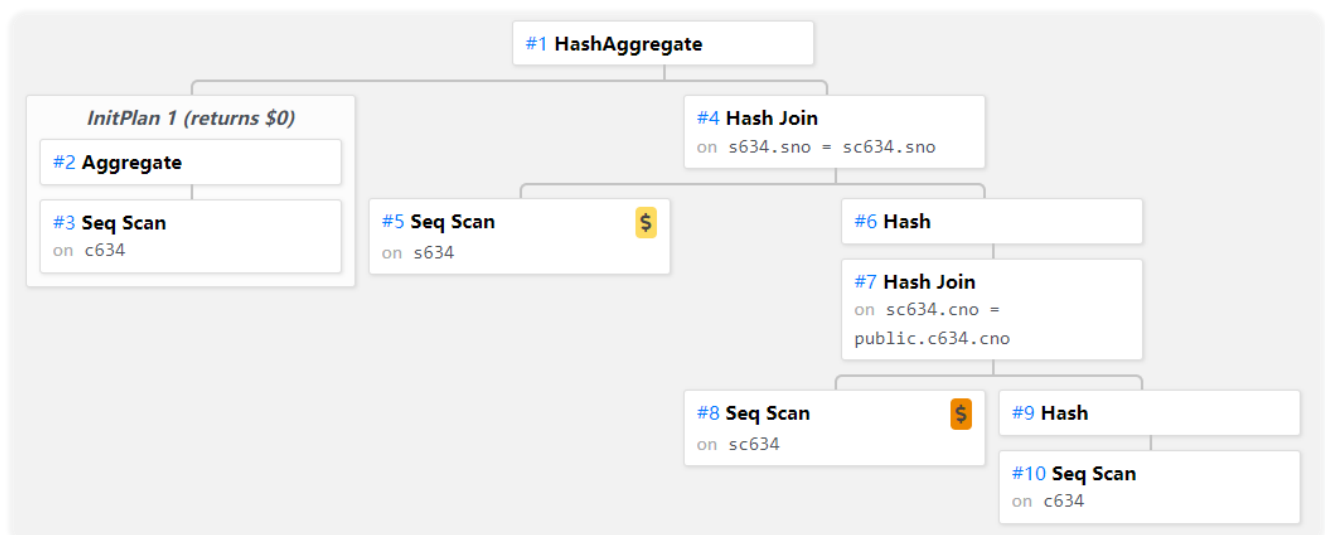
```

```

5      0
6  END ) "总学分"
7  FROM sc634 INNER JOIN c634 ON c634.cno=sc634.cno
8      INNER JOIN s634 ON s634.sno=sc634.sno
9  WHERE c634.CNO LIKE 'CS-%'
10     AND s634.sno = sc634.sno
11     AND c634.cno = sc634.cno
12  GROUP BY s634.SNO
13  HAVING COUNT(*) = (
14      SELECT COUNT(CNO)
15      FROM c634
16      WHERE CNO LIKE 'CS-%'
17  )

```

| QUERY PLAN  |
|---|
| HashAggregate (cost=185.91..186.43 rows=41 width=70)      |
| Group By Key: s634.sno                                    |
| Filter: (count(*) = \$0)                                  |
| InitPlan 1 (returns \$0)                                  |
| -> Aggregate (cost=4.36..4.37 rows=1 width=19)            |
| -> Seq Scan on c634 (cost=0.00..4.35 rows=5 width=11)     |
| Filter: (cno ~~ 'CS-%':text)                              |
| -> Hash Join (cost=143.26..181.03 rows=41 width=30)       |
| Hash Cond: (s634.sno = sc634.sno)                         |
| -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=20) |
| -> Hash (cost=142.75..142.75 rows=41 width=21)            |
| -> Hash Join (cost=4.41..142.75 rows=41 width=21)         |
| Hash Cond: (sc634.cno = public.c634.cno)                  |
| -> Seq Scan on sc634 (cost=0.00..137.82 rows=41 width=27) |
| Filter: (cno ~~ 'CS-%':text)                              |
| -> Hash (cost=4.35..4.35 rows=5 width=16)                 |
| -> Seq Scan on c634 (cost=0.00..4.35 rows=5 width=16)     |
| Filter: (cno ~~ 'CS-%':text)                              |





对比

写法1 `SC.sno=S.sno AND SC.cno=C.cno`

写法2 `sc634 INNER JOIN c634 ON c634.cno=sc634.cno INNER JOIN s634 ON s634.sno=sc634.sno`

导致写法1 `SC和C先连接` 而写法二 `SC和S先连接`，在这个查询中差别微乎其微(只有0.05花费的区别)，我猜想因为这步的结果都是SC的个数，只是扩充了一下信息

## 3.1-8

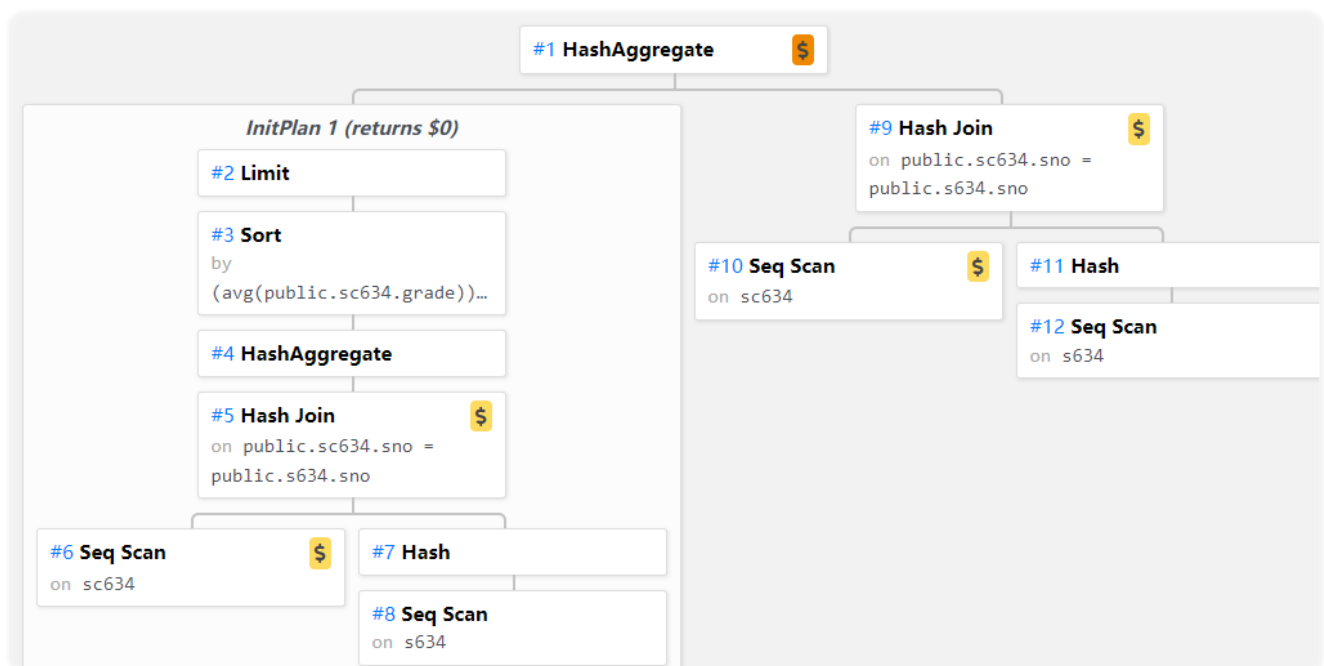
查询选修了3门以上课程（包括3门）的学生中平均成绩最高的同学学号及姓名。

|   | sno      | sname |
|---|----------|-------|
| 主 | 48128560 | 田蓬勃   |

写法1

```
1 SELECT s634.sno,s634.sname
2 FROM s634,sc634
3 WHERE s634.sno=sc634.sno AND GRADE IS NOT NULL
4 GROUP BY s634.sno
5 HAVING AVG(GRADE) = (
6     SELECT AVG(GRADE)
7     FROM s634,sc634
8     WHERE s634.sno = sc634.sno AND GRADE IS NOT NULL
9     GROUP BY s634.SNO
10    HAVING COUNT(*) ≥ 3
11    ORDER BY AVG(GRADE) DESC
12    LIMIT 1
13 )
```

| QUERY PLAN  |
|---|
| ► HashAggregate (cost=565.24..580.40 rows=1011 width=57)    |
| Group By Key: public.s634.sno                               |
| Filter: (avg(public.sc634.grade) = \$0)                     |
| InitPlan 1 (returns \$0)                                    |
| -> Limit (cost=299.01..299.01 rows=1 width=56)              |
| -> Sort (cost=299.01..301.54 rows=1011 width=56)            |
| Sort Key: (avg(public.sc634.grade)) DESC                    |
| -> HashAggregate (cost=278.79..293.95 rows=1011 width=56)   |
| Group By Key: public.s634.sno                               |
| Filter: (count(*) >= 3)                                     |
| -> Hash Join (cost=46.75..241.10 rows=5025 width=16)        |
| Hash Cond: (public.sc634.sno = public.s634.sno)             |
| -> Seq Scan on sc634 (cost=0.00..125.26 rows=5025 width=16) |
| Filter: (grade IS NOT NULL)                                 |
| -> Hash (cost=34.11..34.11 rows=1011 width=11)              |
| -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=11)   |
| -> Hash Join (cost=46.75..241.10 rows=5025 width=25)        |
| Hash Cond: (public.sc634.sno = public.s634.sno)             |
| -> Seq Scan on sc634 (cost=0.00..125.26 rows=5025 width=16) |
| Filter: (grade IS NOT NULL)                                 |
| -> Hash (cost=34.11..34.11 rows=1011 width=20)              |
| -> Seq Scan on s634 (cost=0.00..34.11 rows=1011 width=20)   |



## 写法2

将 `s634.sno=sc634.sno` 改为 `sc634 INNER JOIN s634 ON sc634.sno=s634.sno` 查询是不变的，就不贴图了

### H3

## 2. 插入大量数据

说明：如果这次的查询计划和 插入少量数据 一样，只是数据规模导致cost变大，就不贴图了

在S表中补充数据至约5000行，在C表中补充数据至约1000行，在SC表中补充数据至约30000行，重复四、1.中的SQL语句运行，分析其运行效率，并尝试给出可提高查询效率的改进方法。

插入之后，S634表有5011行，C634表有1008行，SC634表有30026行

|  |  |
|--|--|
|  <b>s634</b><br>表 |  <b>c634</b><br>表 |
| OID<br>16403   | OID<br>16409   |
| 所有者<br>lqs   | 所有者<br>lqs   |
| 行<br>5,011   | 行<br>1,008   |

|   |
|---|
|  <b>sc634</b><br>表 |
| OID<br>16418  |
| 所有者<br>lqs  |
| 行<br>30,026   |

### 3.1-5

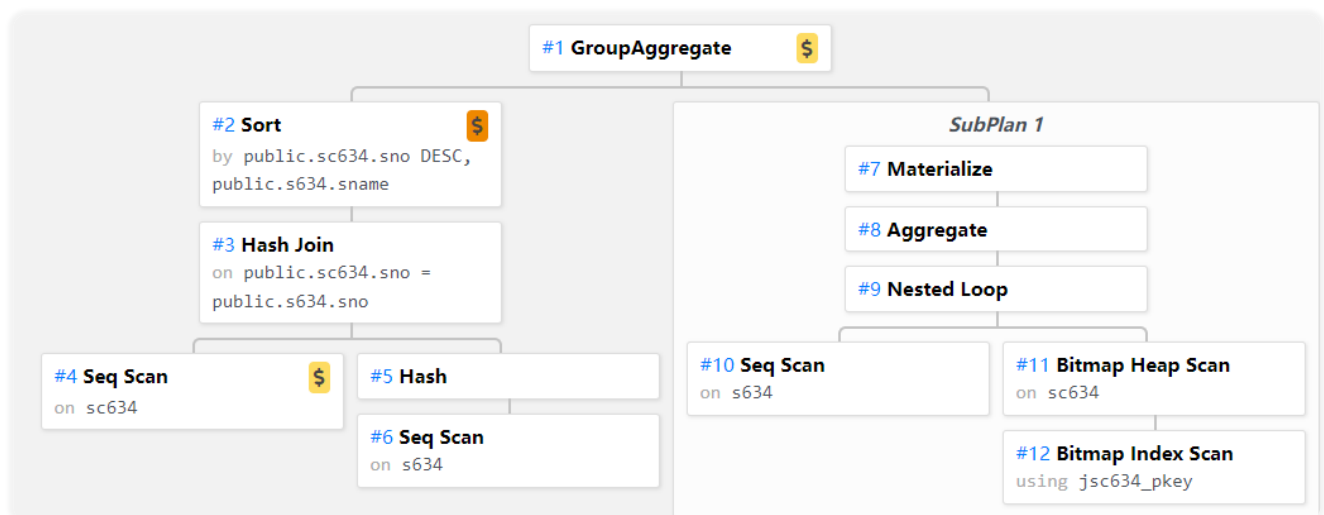
两个写法和少量数据的查询计划是相同的，只不过由于数据规模的增大导致两次花费进一步增大

### 3.1-6

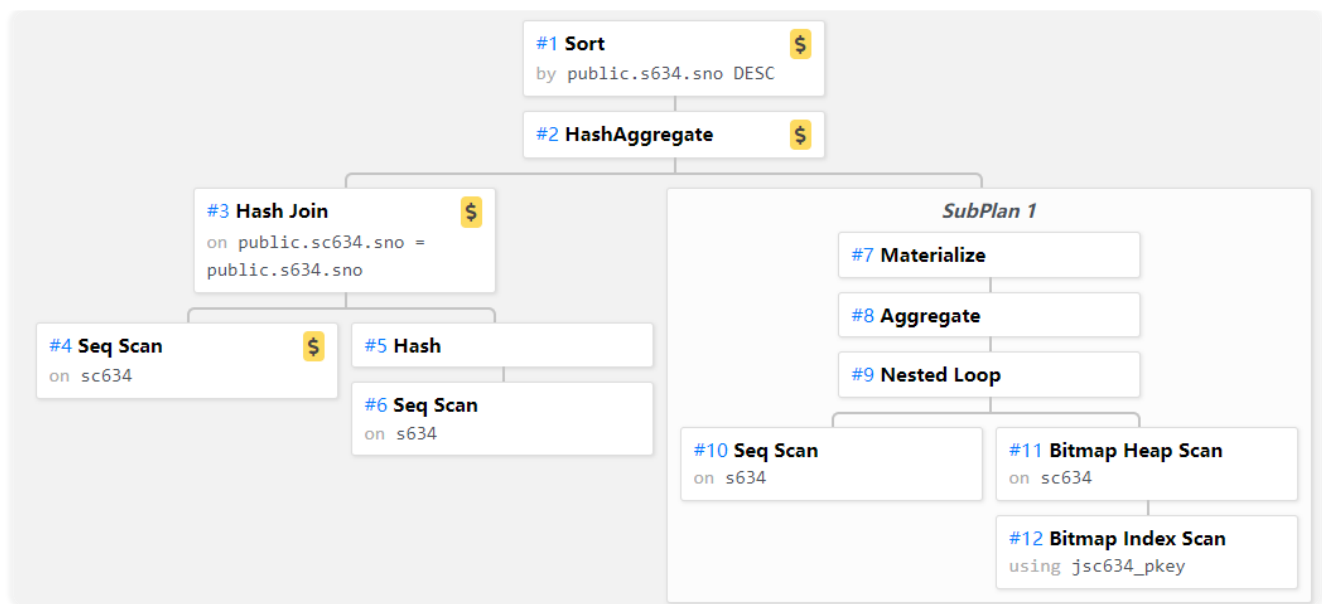
| sno        | sname | 平均成绩 |
|------------|-------|------|
| ▶ 99703124 | 张现阳   | 78.7 |
| 99404824   | 司开卫   | 78.1 |
| 99381556   | 张杰    | 78.8 |
| 99370325   | 刘浩    | 80.5 |
| 99337653   | 郑铁林   | 82.2 |
| 99267802   | 朱瑞    | 78.9 |
| 99210363   | 杨扬    | 97.5 |
| 99159206   | 何元茂   | 81.3 |
| 99031030   | 赵斌    | 80.0 |
| 98973119   | 赵政霄   | 81.1 |
| 98961920   | 阮小娥   | 89.3 |
| 98618539   | 赵讲社   | 77.9 |
| 98571385   | 赵肖    | 81.3 |
| 98525176   | 刘江华   | 78.0 |
| 98440495   | 陈秀秀   | 88.0 |
| 98421476   | 赵亚宁   | 85.0 |
| 98317707   | 高雅    | 91.0 |
| 98300203   | 苏豆    | 79.3 |
| 98269352   | 王良楠   | 78.4 |
| 98194894   | 胡二江   | 78.5 |
| 98123862   | 宋亚勤   | 78.3 |
| 98023386   | 秦得福   | 79.3 |
| 98012601   | 范力宏   | 81.8 |

写法1

| QUERY PLAN   |
|--|
| ► GroupAggregate (cost=3747.99..4986.52 rows=30025 width=90)         |
| Group By Key: public.sc634.sno, public.s634.sname                    |
| Filter: (SubPlan 1)  |
| -> Sort (cost=3549.79..3624.85 rows=30025 width=26)                  |
| Sort Key: public.sc634.sno DESC, public.s634.sname                   |
| -> Hash Join (cost=195.75..1316.85 rows=30025 width=26)              |
| Hash Cond: (public.sc634.sno = public.s634.sno)                      |
| -> Seq Scan on sc634 (cost=0.00..708.26 rows=30025 width=17)         |
| Filter: (grade IS NOT NULL)  |
| -> Hash (cost=133.11..133.11 rows=5011 width=20)                     |
| -> Seq Scan on s634 (cost=0.00..133.11 rows=5011 width=20)           |
| SubPlan 1  |
| -> Materialize (cost=198.20..198.22 rows=1 width=38)                 |
| -> Aggregate (cost=198.20..198.21 rows=1 width=38)                   |
| -> Nested Loop (cost=4.31..198.17 rows=12 width=6)                   |
| -> Seq Scan on s634 (cost=0.00..145.64 rows=2 width=11)              |
| Filter: ((sname)::text = '王涛'::text)                                 |
| -> Bitmap Heap Scan on sc634 (cost=4.31..26.21 rows=6 width=17)      |
| Recheck Cond: (sno = public.s634.sno)                                |
| Filter: (grade IS NOT NULL)  |
| -> Bitmap Index Scan on jsc634_pkey (cost=0.00..4.31 rows=6 width=0) |
| Index Cond: (sno = public.s634.sno)                                  |



写法2



| QUERY PLAN   |
|--|
| Sort (cost=2267.32..2279.84 rows=5011 width=90)                      |
| Sort Key: public.s634.sno DESC                                       |
| -> HashAggregate (cost=1815.30..1959.37 rows=5011 width=90)          |
| Group By Key: public.s634.sno, public.s634.sname                     |
| Filter: (SubPlan 1)  |
| -> Hash Join (cost=195.75..1316.85 rows=30025 width=26)              |
| Hash Cond: (public.sc634.sno = public.s634.sno)                      |
| -> Seq Scan on sc634 (cost=0.00..708.26 rows=30025 width=17)         |
| Filter: (grade IS NOT NULL)  |
| -> Hash (cost=133.11..133.11 rows=5011 width=20)                     |
| -> Seq Scan on s634 (cost=0.00..133.11 rows=5011 width=20)           |
| SubPlan 1  |
| -> Materialize (cost=198.20..198.22 rows=1 width=38)                 |
| -> Aggregate (cost=198.20..198.21 rows=1 width=38)                   |
| -> Nested Loop (cost=4.31..198.17 rows=12 width=6)                   |
| -> Seq Scan on s634 (cost=0.00..145.64 rows=2 width=11)              |
| Filter: ((sname)::text = '王涛'::text)                                 |
| -> Bitmap Heap Scan on sc634 (cost=4.31..26.21 rows=6 width=17)      |
| Recheck Cond: (sno = public.s634.sno)                                |
| Filter: (grade IS NOT NULL)  |
| -> Bitmap Index Scan on jsc634_pkey (cost=0.00..4.31 rows=6 width=0) |
| Index Cond: (sno = public.s634.sno)                                  |

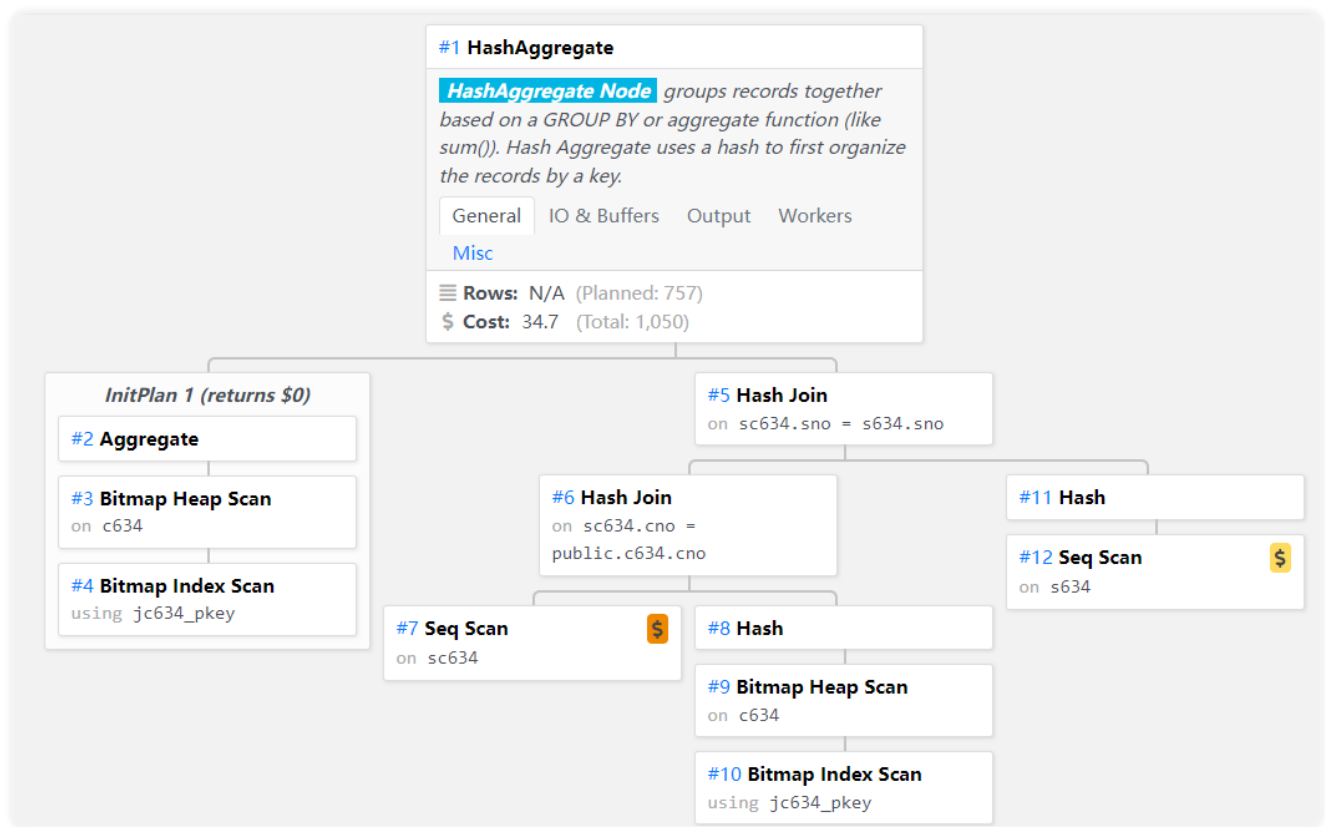
对比

这两个图和插入少量数据其实没什么区别，但值得注意，数据量变大之后，写法2的hashAggregate的cost仅为150左右，hashAggregate之后数据量相比很少，连锁地导致sort的cost仅为10左右。

对比看来，写法1由于是groupAggregate，所以要先排序，sort的cost为75，而GroupAggregate的cost高达1000多

## 3.1-7

写法1和2的底层执行是相同的



少量数据的时候这两个查询一个SC先连接了S，一个先连接C

而数据量变大后sc均先和c连接，并且 **seq Scan** 变成了 **Bitmap Index Scan**

网上查到如下说法：

- **顺序扫描SeqScan**

直接对数据表堆数据(Heap Data)进行顺序扫描,适用于选择率较高的场景.

- **索引扫描IndexScan**

通过访问索引获得元组位置指针后再访问堆数据,适用于选择率较低的场景.

- **位图堆扫描BitmapHeapScan**

位图堆扫描需要首先通过BitmapIndexScan(位图索引扫描)把符合条件的元组所在的Page(Block) ID 存储在Bitmap中,然后再通过Bitmap访问堆数据,适用于选择率不高不低的场景,介于上面两种扫描方式之间.

但是吧，我两次生成的数据是一样的，1个student随机分配5个课程的成绩，如果随机效果好的话，我觉得选择率应该差不多，所以我认为第一次数据量比较小，所以直接用了Seq Scan

### 3.1-8

| sno        | sname |
|------------|-------|
| ▶ 86859567 | 白秀秀   |

两个查询是一样的，且执行过程和 少量数据 是一样的

### H3 总结

HashAggregate和GroupAggregate：但从这两次规模的数据来看，显然HashAggregate的效率高，GroupAggregate前一步需要Sort，而一般来说规模不小，两次综合下来更耗时，相关链接：<https://developer.aliyun.com/article/656603>

Seq Scan和Bitmap Index Scan：前者适用于小规模数据，后者在大规模数据速度变快，相关链接：<http://t.zoukankan.com/gaojian-p-2761952.html>

尽量少嵌套查询

## H2 五、数据备份

数据备份来源：计算机93李云广

他的三个表实现：

| 名       | 类型      | 长度  | 小数点 | 不是 null                             | 键   | 注释 |
|---------|---------|-----|-----|-------------------------------------|-----|----|
| ▶ cno   | char    | 5   | 0   | <input checked="" type="checkbox"/> | 🔑 1 |    |
| cname   | varchar | 90  | 0   | <input checked="" type="checkbox"/> |     |    |
| period  | int2    | 16  | 0   | <input checked="" type="checkbox"/> |     |    |
| credit  | numeric | 3   | 1   | <input checked="" type="checkbox"/> |     |    |
| teacher | varchar | 200 | 0   | <input checked="" type="checkbox"/> |     |    |

| 名      | 类型        | 长度  | 小数点 | 不是 null                             | 键   | 注释 |
|--------|-----------|-----|-----|-------------------------------------|-----|----|
| ▶ sno  | char      | 8   | 0   | <input checked="" type="checkbox"/> | 🔑 1 |    |
| sname  | varchar   | 150 | 0   | <input checked="" type="checkbox"/> |     |    |
| sex    | char      | 3   | 0   | <input checked="" type="checkbox"/> |     |    |
| bdate  | timestamp | 0   | 0   | <input checked="" type="checkbox"/> |     |    |
| height | float4    | 24  | 0   | <input type="checkbox"/>            |     |    |
| dorm   | char      | 15  | 0   | <input type="checkbox"/>            |     |    |

| 名     | 类型      | 长度 | 小数点 | 不是 null                             | 键   | 注释 |
|-------|---------|----|-----|-------------------------------------|-----|----|
| ▶ sno | char    | 8  | 0   | <input checked="" type="checkbox"/> | 🔑 2 |    |
| cno   | char    | 5  | 0   | <input checked="" type="checkbox"/> | 🔑 1 |    |
| grade | numeric | 6  | 3   | <input type="checkbox"/>            |     |    |

三个表截图如下：



| sno        | sname | sex | bdate               | height | dorm    |
|------------|-------|-----|---------------------|--------|---------|
| ▶ 10080520 | 邱赚    | 男   | 2003-11-27 00:00:00 | 1.54   | 西10舍237 |
| 10510744   | 褚玥博   | 女   | 2005-04-05 00:00:00 | 1.58   | 东10舍138 |
| 10160698   | 颜航    | 女   | 2001-09-12 00:00:00 | 1.85   | 东6舍640  |
| 10101821   | 鲁梦决   | 男   | 2005-12-17 00:00:00 | 1.51   | 西12舍631 |
| 10441972   | 周泽泽   | 女   | 2005-01-24 00:00:00 | 1.52   | 东15舍334 |
| 10872896   | 奚旺文   | 女   | 2000-08-20 00:00:00 | 1.68   | 东4舍536  |
| 10482019   | 钱玥    | 女   | 2000-02-03 00:00:00 | 1.77   | 东12舍112 |
| 10436764   | 蔡琳一   | 男   | 2002-02-02 00:00:00 | 1.59   | 西18舍127 |
| 10830957   | 元昂苑   | 男   | 2000-05-27 00:00:00 | 1.75   | 西18舍305 |
| 10348058   | 赵峻文   | 女   | 2002-01-20 00:00:00 | 1.86   | 东12舍639 |
| 10054450   | 姜煜    | 男   | 2005-12-03 00:00:00 | 1.78   | 西6舍518  |
| 10289884   | 骆润    | 男   | 2003-01-14 00:00:00 | 1.75   | 西20舍534 |
| 10696861   | 单炜润   | 女   | 2001-11-08 00:00:00 | 1.74   | 东12舍615 |
| 10718870   | 柳颖    | 女   | 2003-08-10 00:00:00 | 1.83   | 东10舍425 |

| sno        | cno   | grade  |
|------------|-------|--------|
| ▶ 10138493 | EE-05 | 70.460 |
| 10138493   | WB-05 | 74.090 |
| 10138493   | MQ-03 | 95.790 |
| 10138493   | XI-02 | 75.070 |
| 10138493   | IK-01 | 82.300 |
| 10138493   | VU-04 | 52.620 |

| cno     | cname        | period | credit | teacher   |
|---------|--------------|--------|--------|-----------|
| ▶ SM-02 | 陕西话          | 16     | 2.0    | 冯潇        |
| LT-03   | 大学生心理健康与自我调适 | 32     | 4.0    | 吴梦瑶,康楠,姚斌 |
| AB-02   | 课外实践8学分      | 0      | 0.0    | 王倩,岳娅萍    |
| TK-01   | 大学生创业训练与实践   | 32     | 4.0    | 陈永华       |
| LV-04   | 大学生职业发展与规划   | 32     | 4.0    | 郑旭红       |
| EP-05   | 在线开放课程助教     | 0      | 0.0    | 测试        |
| AC-01   | 毕业设计(论文)     | 128    | 8.0    | 冯潇        |
| GG-01   | 中国传统文化实践-2   | 16     | 2.0    | 孙立滨       |
| AJ-01   | 中国古代文学1      | 32     | 2.0    | 马春燕       |
| EE-05   | 中国概况         | 32     | 2.0    | 赵炜        |
| LL-02   | 当代影视鉴赏       | 32     | 2.0    | 张知惊       |
| PF-03   | 简明中国历史       | 32     | 2.0    | 路荣        |
| ID-04   | 汉语综合课2       | 96     | 6.0    | 杨杨        |

评价：表设计的挺合理的，数据质量也挺好的

## H2 六、如何获得数据、插入数据

### H3 获取数据

使用语言: `nodejs` 的 `fetch` 模块和 `fs` 模块

实现说明: 发三次 `post` 请求(每次 `POST` 请求 `pageSize` 字段设置为1000, 每发送一个请求 `pageNumber` 加1)来获得课程信息, 把这3000个信息的 `KKDWD_DISPLAY` 字段放到set里, 得到如下结果

拿到大约3000个课程信息, 发现开课单位共有31个, 手动确定课程编号前两位。

重复上述流程, 获得课程信息并且增加课程编号, 同样获得姓名信息, 最终获得不重复的2720个老师和1567个课程。显然老师的信息不足够5000行, 因此, 将老师的姓氏进行随机替换, 然后再去重, 最终得到了7309个老师信息, 当然, 这些老师信息也可以看作是学生姓名。然后将课程信息和姓名写入两个文件。

由于初始课程信息存在 `EE` 和 `CS` 这两个课程号, 而我校开课单位均为电子与信息学部, 因此我将电子与信息学部开课的编号进一步划分为EE和CS(方法: 如果课程名包括计算机, 人工智能或者开课班级有计算机, 则此课程为CS, 否则为EE)

```
1 Set(31) {
2   '人文社会科学学院', →RW
3   '医学部', →YX
4   '电气工程学院', →DQ
5   '马克思主义学院', →MK
6   '机械工程学院', →JX
7   '化学学院', →HX
8   '法学院', →FX
9   '生命科学与技术学院', →SM
10  '经济与金融学院', →JJ
11  '物理学院', →WL
12  '管理学院', →GL
13  '电子与信息学部', →EE/CS
14  '人居环境与建筑工程学院', →RJ
15  '外国语学院', →WG
16  '公共政策与管理学院', →GG
17  '体育中心', →TY
18  '能源与动力工程学院', →ND
19  '金禾经济研究中心', →JH
20  '化学工程与技术学院', →HX
21  '数学与统计学院', →SX
22  '实践教学中心/工程坊', →SB
23  '航天航空学院', →HK
24  '军事教研室', →JS
25  '材料科学与工程学院', →CL
26  '国际教育学院', →GJ
```

```

27 '新闻与新媒体学院', →XW
28 '钱学森学院/钱学森书院', →QX
29 '学工部/学生处/武装部', →XG
30 '前沿科学技术研究院', →QY
31 '学生就业创业指导服务中心', →JY
32 '教务处' →JW
33 }

```

## 核心代码

```

1  async function sendPostToGetInfo() {
2
3      //发异步的fetch请求
4      const res = await
fetch("http://ehall.xjtu.edu.cn/jwapp/sys/kcbcx/modules/qxkcb/qxfbkcxc.do", {
5          method: 'POST',
6          mode: 'cors', // no-cors, *cors, same-origin
7          cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-
cached
8          credentials: "include", // include, *same-origin, omit
9          headers: {
10              Accept: 'application/json, text/javascript, */*; q=0.01',
11              "Accept-Encoding": 'gzip, deflate',
12              Host: 'ehall.xjtu.edu.cn',
13              Referer: 'http://ehall.xjtu.edu.cn/jwapp/sys/kcbcx/*default/index.do?
amp_sec_version=1&gid=TmRDa3NtVm5tNGdQd1V6YThNQUFndFplUzdaREJHblRnR3VuNEFlaVJ4
bFp1am1ScUxybisxTTVtR2p3NGNEQkFLS1A4bUJkQ09pL0ZrODZzd3JXeGc9PQ&EMAP_LANG=zh&THEM
E=cherry',
14              'Content-type': 'application/x-www-form-urlencoded; charset=UTF-8',
15              Cookie: Cookie,
16              'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.67 Safari/537.36',
17          },
18          body: `querySetting=[{"name":"XNXQDM","value":"2021-2022-
2","linkOpt":"and","builder":"equal"},
[{"name":"RWZTDM","value":"1","linkOpt":"and","builder":"equal"},
{"name":"RWZTDM","linkOpt":"or","builder":"isNull"}]]&*order=+KKDWDWM,+KCH,+KXH&S
KXQ=&KSJC=&JSJC=&pageSize=10000&pageNumber=${pageNumber++}`
19      })
20
21      //解析成json数据格式
22      const body = await res.json();
23      console.log(body);
24      let courseArray = body.datas.qxfbkcxc.rows;
25
26

```

```

27 //参数的意思
28 /**
29  * @SKBJ上课班级
30  * @SKJS上课教师
31  * @KCM课程名称
32  * @KCH课程号
33  * @YPSJDD上课时间地点
34  * @XS学时
35  * @XF学分
36  * @KKDWD_DISPLAY开课单位
37 */
38
39 courseArray.forEach(item => {
40
41     //判断为null undefined 包含空格(说明这是外教)
42     if (!(item.SKJS = null || item.SKJS = undefined || item.SKJS.includes('
43         STUDENT_NAME = STUDENT_NAME.concat(item.SKJS + ',')
44
45     let cno = COURSE_DEPARTEMT.get(item.KKDWD_DISPLAY);
46     if (cno = undefined) { //确定是EE还是CS
47         //CS的课
48         if ((item.KCM !== null && item.KCM.indexOf('计算机') !== -1) || (item.SKBJ !==
49 null && item.SKBJ.indexOf('计算机') !== -1)) cno = 'CS';
50         else cno = 'EE';
51     }
52
53     //之前没有找到并且课程名字不包括 (
54     if (COURSE_NAME.includes(item.KCM) = false &&
55 item.KCM.includes('(')=false) {
56         COURSE_NAME.push(item.KCM.slice(0, 18));
57         COURSE_NUM.push(cno + '-' + randomNumber);
58         randomNumber++;
59     }
60
61     //上课班级
62     let SKBJ = '',
63         indexOfSKBJ;
64
65     if (item.SKBJ = null) {
66         SKBJ = '所有';
67     } else {
68         indexOfSKBJ = item.SKBJ.indexOf(',');
69         if (indexOfSKBJ = -1) SKBJ = item.SKBJ;
70         else SKBJ = item.SKBJ.slice(0, indexOfSKBJ);
71     }
72
73     //上课教师

```

```

73     let SKJS = '',
74         indexOfSKJS;
75
76     if (item.SKJS == null) {
77         SKJS = '不知道';
78     } else {
79         indexOfSKJS = item.SKJS.indexOf(',');
80         if (indexOfSKJS == -1) SKJS = item.SKJS;
81         else SKJS = item.SKJS.slice(0, indexOfSKJS);
82     }
83
84 })
85 }

```

### H3 插入数据

读取Student.txt和Course.txt文件，由此得到学生姓名，老师姓名，课程号和课程名字数组，再随机生成其它信息，然后构造sql插入的字符串，需要注意的是，我每次随机生成学号，因此后面生成的学号可能会和前面相同产生冲突，因此使用Set用于去重。

第二次插入大量数据的时候，随机生成的学号和第一次插入的学号也可能产生冲突，因此先执行一个查询获得所有学号，加入Set用于后续去重

每一个学生随机分配5个课程构成SC

这里贴构造S表插入的字符串代码

```

1  try{
2      student5000Arr = Arrays.copyOfRange(studentArr,1000,5000); //1000-5000
3      course1000Arr = Arrays.copyOfRange(courseArr,100,1000); //100-1000
4      teacher1000Arr = Arrays.copyOfRange(studentArr,studentArr.length-
5      1000,studentArr.length-100); //倒数后900个是充当老师
6      sno5000Arr = Arrays.copyOfRange(studentArr,0,4000); //只是分配空间而已
7      System.out.println(student5000Arr.length); //读了4000个学生的姓名
8      System.out.println(course1000Arr.length); //读了900个课程的cno
9      BufferedWriter out;
10     out = new BufferedWriter( new OutputStreamWriter( new
11     FileOutputStream("C:\\Users\\AIERXUAN\\Desktop\\temp.txt"),
12     StandardCharsets.UTF_8) );
13     //4000个学生的随机信息
14     for(int i=0;i<4000;i++){
15         String sno = getRandomSno(),
16         sname = student5000Arr[i],
17         sex = getRandomSex(),

```

```

15     bdate = getRandomBdate(),
16     dorm = getRandomDorm();
17     double height = getRandomHeight();
18     if(sname==null || sname==" " || sname.equals("")) System.out.println(i);
19     //处理sno重复的问题
20     if( snoSet.contains(sno) ){
21         i--;
22         continue;
23     } else {
24         snoSet.add(sno);
25         sno5000Arr[i] = sno;
26         out.write(sno5000Arr[i] + "\n");
27     }
28     String singleInsertIntoS;
29     if(i!=3999){
30         singleInsertIntoS = "(" + sno + "," + sname + "," + sex + "," + bdate + "," + height + "," + dorm + ")," + "\n";
31     }else{
32         singleInsertIntoS = "(" + sno + "," + sname + "," + sex + "," + bdate + "," + height + "," + dorm + ")";
33     }
34
35     sqlOfS += singleInsertIntoS;
36     //System.out.println( sno + "," + sname + "," + sex + "," + bdate + "," + height + "," + dorm );
37 }
38 out.flush();
39 out.close();
40 }catch (Exception e){
41     System.out.println("Exception");
42     e.printStackTrace();
43 }

```

全部代码在这个链接里面：<https://github.com/LQS-LQS/xjtu-database>，这里就不贴代码了。

### H3 数据展示

| sno        | sname | sex | bdate               | height | dorm    |
|------------|-------|-----|---------------------|--------|---------|
| 39043687   | 孙明望   | 男   | 2000-03-19 00:00:00 | 1.50   | 东11舍607 |
| 67169802   | 孙彦龙   | 女   | 1998-09-06 00:00:00 | 1.81   | 东20舍220 |
| 96339682   | 霍庆华   | 女   | 1998-12-21 00:00:00 | 1.77   | 西17舍117 |
| 06379201   | 刘杨    | 男   | 1999-02-10 00:00:00 | 1.58   | 东9舍022  |
| 80717830   | 王晓悦   | 男   | 1999-03-02 00:00:00 | 1.61   | 东4舍129  |
| 46075063   | 郑亚梅   | 女   | 2001-02-27 00:00:00 | 1.85   | 西19舍223 |
| 32826356   | 李云镇   | 女   | 1998-06-09 00:00:00 | 1.75   | 东18舍509 |
| 64238978   | 周静    | 女   | 1999-12-14 00:00:00 | 1.70   | 西2舍620  |
| 11250444   | 刘燕    | 男   | 1999-12-21 00:00:00 | 1.59   | 北4舍624  |
| 18661540   | 李昕    | 女   | 2001-12-11 00:00:00 | 1.70   | 东17舍017 |
| 03345123   | 程健    | 女   | 1999-10-15 00:00:00 | 1.87   | 东9舍104  |
| 13976252   | 周锋    | 女   | 1998-09-10 00:00:00 | 1.79   | 东7舍115  |
| 80193156   | 王瑜萍   | 女   | 2001-03-22 00:00:00 | 1.80   | 东20舍419 |
| 23506990   | 冯剑琴   | 女   | 2002-07-03 00:00:00 | 1.80   | 南6舍007  |
| 03224369   | 方爱平   | 女   | 1999-01-11 00:00:00 | 1.84   | 西6舍027  |
| 95968089   | 李宇    | 男   | 1999-12-22 00:00:00 | 1.71   | 西15舍626 |
| ▶ 69161987 | 刘虹    | 男   | 1997-02-19 00:00:00 | 1.60   | 西6舍115  |
| 38639021   | 吕社民   | 女   | 1999-02-16 00:00:00 | 1.66   | 南8舍925  |
| 27691181   | 孙红利   | 男   | 2000-04-22 00:00:00 | 1.71   | 东14舍805 |
| 80377656   | 薛华    | 男   | 1999-09-27 00:00:00 | 1.59   | 东1舍923  |

| sno       | cno   | grade |
|-----------|-------|-------|
| 5273967   | DQ-93 | 56.0  |
| 9327160   | WG-94 | 55.0  |
| 4607506   | RW-95 | 92.0  |
| 1803641   | YX-97 | 87.0  |
| 4100805   | YX-92 | 63.5  |
| 0241302   | HK-53 | 95.5  |
| 0300182   | RW-76 | 56.5  |
| 4814387   | GL-85 | 94.0  |
| 3570085   | DQ-20 | 92.5  |
| 9722304   | RJ-69 | 91.0  |
| 6084076   | HX-39 | 59.0  |
| 5814019   | SB-44 | 81.5  |
| 1191906   | DQ-93 | 79.5  |
| 7972811   | CS-45 | 83.5  |
| ▶ 9166561 | ND-48 | 96.5  |
| 1192180   | ND-59 | 77.5  |
| 2572106   | DQ-22 | 97.0  |

| cno      | cname      | period | credit | teacher |
|----------|------------|--------|--------|---------|
| ▶ ND-384 | 流体力学 I     | 40     | 2.0    | 田方华     |
| HX-385   | 物理化学 I -1  | 40     | 2.0    | 陈兵      |
| EE-386   | 半导体器件      | 120    | 6.0    | 周璐薇     |
| RW-387   | 汉字与中国传统文化  | 70     | 3.5    | 苏焜      |
| SM-388   | 生命科学基础I    | 110    | 5.5    | 曾祥金     |
| SM-389   | 生物医学工程导论   | 70     | 3.5    | 范圣华     |
| DQ-390   | 电机设计       | 70     | 3.5    | 杨路      |
| YX-391   | 机能实验学 I    | 60     | 3.0    | 王翊岑     |
| SM-392   | 营养与健康      | 20     | 1.0    | 陈艳      |
| YX-393   | 临床技能训练     | 50     | 2.5    | 段培高     |
| RW-394   | 社区概论       | 50     | 2.5    | 张英良     |
| DQ-395   | 大数据科学与应用技术 | 80     | 4.0    | 范博雅     |
| YX-396   | 口腔科学       | 110    | 5.5    | 朱成志     |
| YX-397   | 法医专业实习     | 60     | 3.0    | 王卫锋     |
| HX-398   | 有机化学Ⅲ      | 80     | 4.0    | 范子瑶     |
| RW-399   | 创新实践       | 110    | 5.5    | 李永涛     |
| HK-400   | 力学分析软件概论   | 110    | 5.5    | 刘大旭     |
| EE-401   | 光电子学实验1    | 110    | 5.5    | 马金玲     |