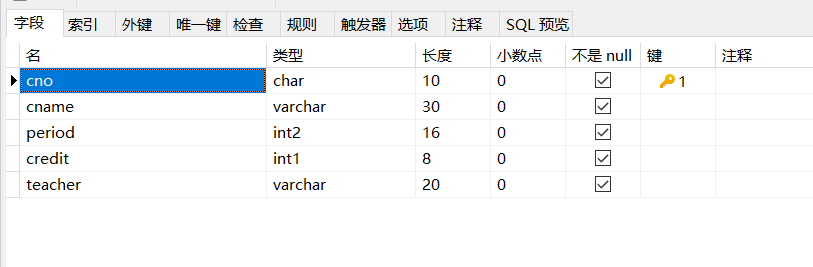
使用openGauss数据库，并使用华为云服务器安装

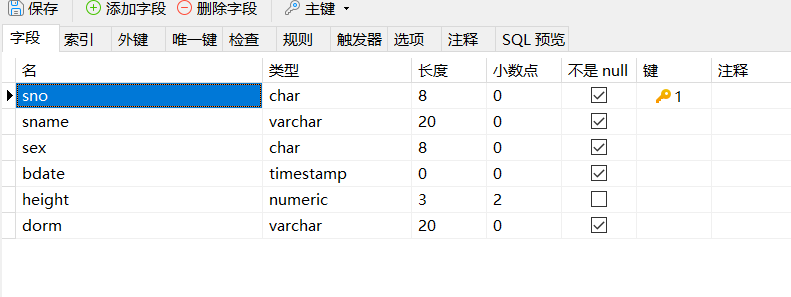
一、使用navicat显示数据库

三个表的属性设计如下：

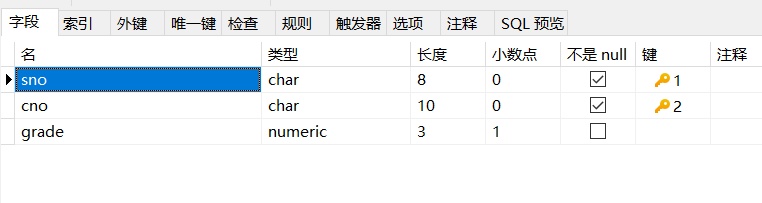
C030表



S030表

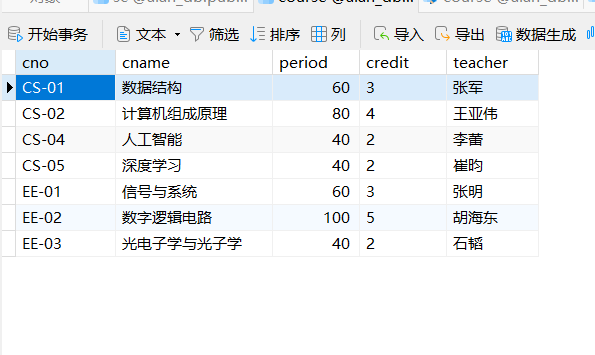


SC030表

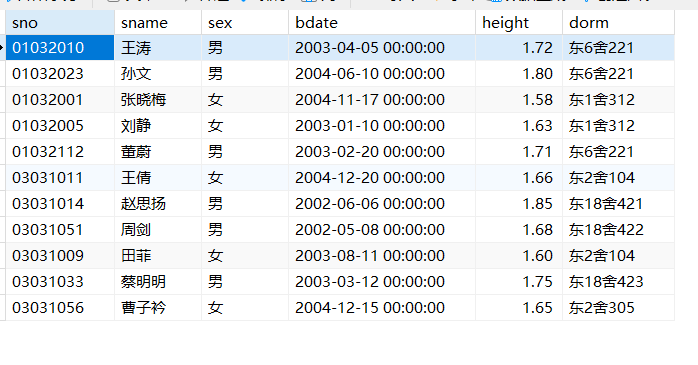


录入数据如下：

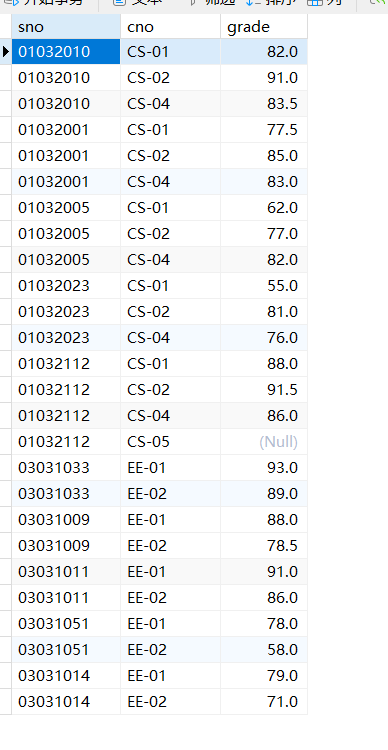
C030表



S030表



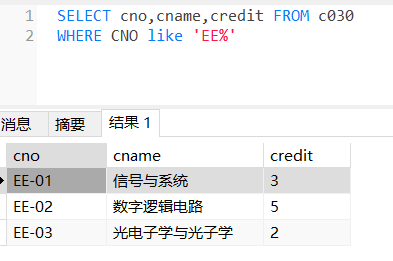
SC030表



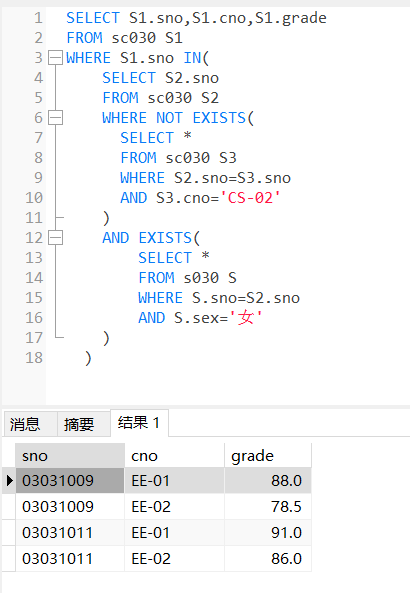
二、使用sql查询

1．在上述基本表上完成以下查询：

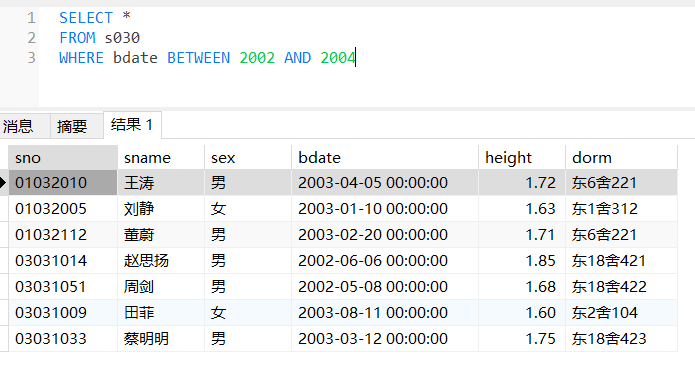
(1) 查询电子工程系（EE）所开课程的课程编号、课程名称及学分数。



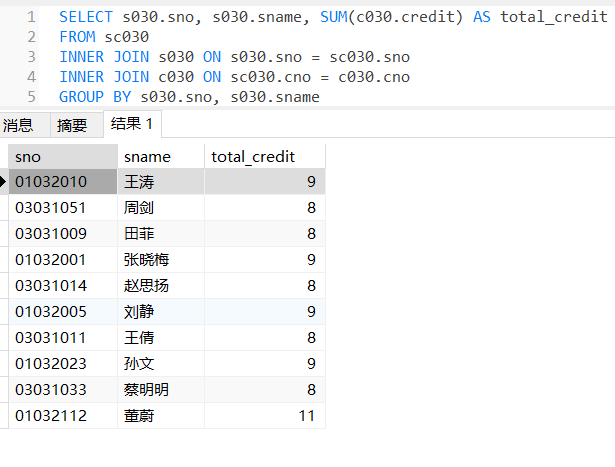
(2) 查询未选修课程“CS-02”的女生学号及其已选各课程编号、成绩。



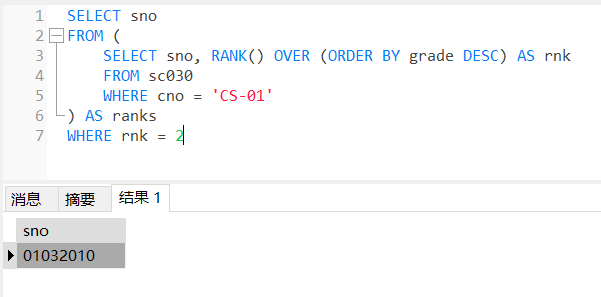
(3) 查询 2002 年～2003 年出生学生的基本信息。



(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。



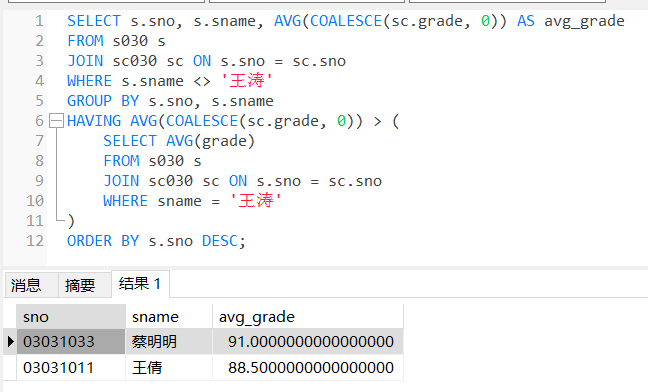
(5) 查询选修课程“CS-01”的学生中成绩第二高的学生学号。



(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。

数据中sc表中存在null值

使用COALESCE(sc.grade, 0)将null转化为0

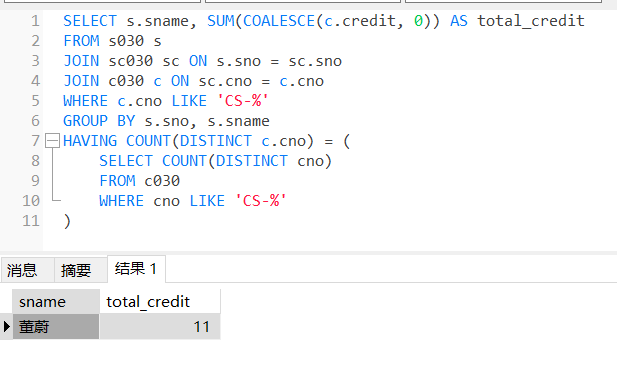


(7)查询选修了计算机专业全部课程（课程编号为“CS-××”）的学生姓名

及已获得的学分总数。

数据中sc表中存在null值，视为已选修课但未获得学分

使用COALESCE(c.credit, 0)将null转化为0

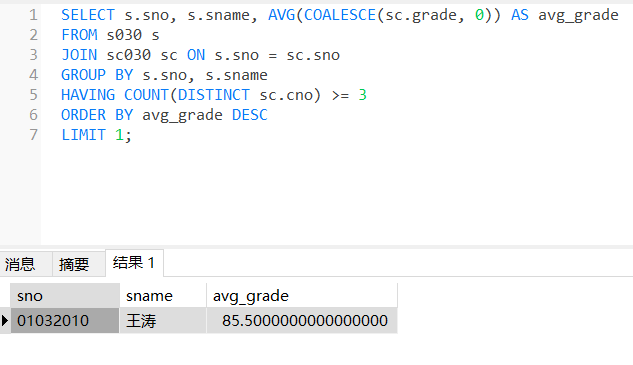


(8) 查询选修了 3 门以上课程（包括 3 门）的学生中平均成绩最高的同学学

号及姓名。

同样使用COALESCE(sc.grade, 0)将null转化为0，在计算平均成绩时会有变化

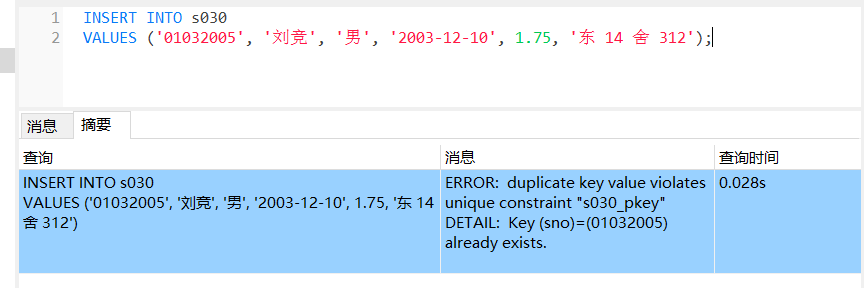
从董蔚变为王涛

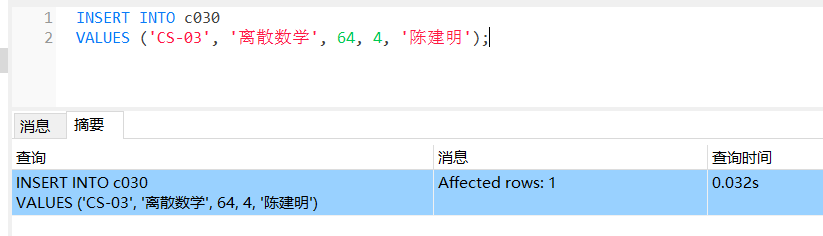


2．分别在 S×××和 C×××表中加入记录(‘01032005’，‘刘竞’，‘男’，

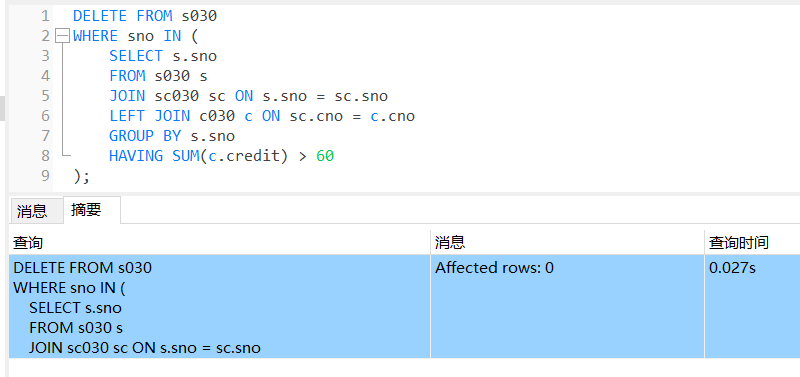
‘2003-12-10’，1.75，‘东 14 舍 312’)及(‘CS-03’，“离散数学”，64，4，‘陈建明’)。

两条数据一条主键冲突未插入，一条成功插入





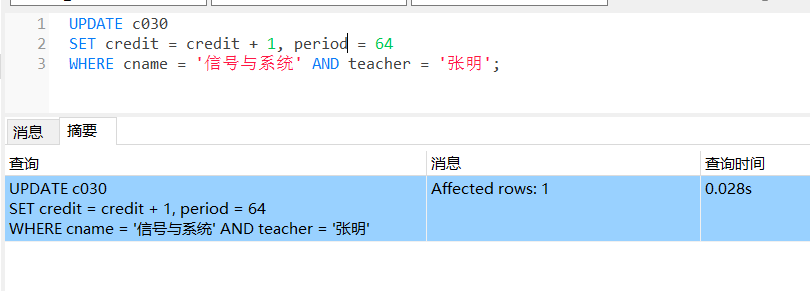
3．将 S×××表中已修学分数大于 60 的学生记录删除。



S表未变更

4．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

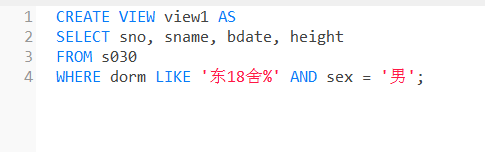
加一个学分。

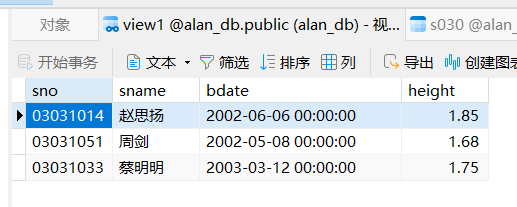


5．建立如下视图：

(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

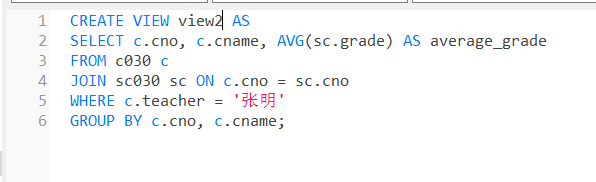
性。

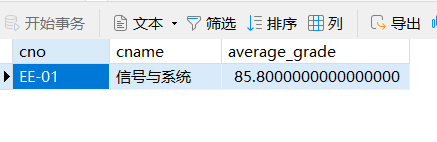




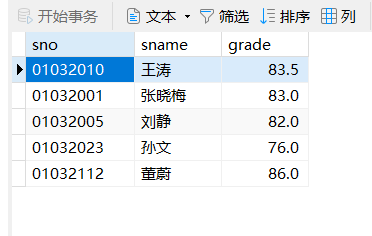
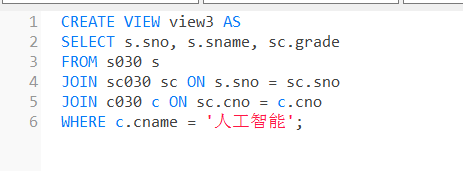
(2)“张明”老师所开设课程情况的视图，包括课程编号、课程名称、平均成

绩等属性。





(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。



四、完成以下操作，将相应结果截屏图保存，并写入实验报告中。

**数据生成方法放在文件末尾，使用JDBC方法导入数据**

1．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 20000 行。在向 SC×××表中补充数据的过程中，随机选择成绩低于60分的 200 行选课记录删除。

插入20000行数据程序如下，另外两个类似：

1. **public** **static** **void** insert\_sc() {
2. String csvFile = "sc.csv";
3. String line;
4. String csvSplitBy = ",";
5. **int** maxInsertCount = 20000;
6. **int** insertCount = 0;
8. **try** (BufferedReader br = **new** BufferedReader(**new** InputStreamReader(**new** FileInputStream(csvFile), "UTF-8"));
9. Connection conn = getConnection();
10. Statement stmt = conn.createStatement()) {
12. conn.setAutoCommit(**false**);
13. // 跳过CSV文件的标题行
14. br.readLine();
15. System.out.println("Start inserting data...");
17. String insertSQL = "INSERT INTO SC030 (sno, cno, grade) VALUES (?, ?, ?)";
18. **try** (PreparedStatement preparedStatement = conn.prepareStatement(insertSQL)) {
19. **while** ((line = br.readLine()) != **null** && insertCount < maxInsertCount) {
20. // 使用逗号分隔符解析每行数据
21. String[] data = line.split(csvSplitBy);
23. preparedStatement.setString(1, data[0]);
24. preparedStatement.setString(2, data[1]);
26. **if** (data[2].isEmpty()) {
27. preparedStatement.setNull(3, java.sql.Types.DECIMAL);
28. } **else** {
29. **try** {
30. preparedStatement.setBigDecimal(3, **new** BigDecimal(data[2]));
31. } **catch** (NumberFormatException e) {
32. System.err.println("Skipping invalid GRADE: " + data[2]);
33. **continue**;
34. }
35. }
36. // 执行插入操作
37. preparedStatement.addBatch();
38. // 打印插入信息
39. System.out.println("Inserted record: " + data[0] + ", " + data[1] + ", " + data[2]);
40. insertCount++;
42. **if** (insertCount % 10 == 0) {
43. preparedStatement.executeBatch();
44. conn.commit();
45. }
46. }
48. // 执行剩余的批处理
49. **if** (insertCount % 10 != 0) {
50. preparedStatement.executeBatch();
51. conn.commit();
52. }
54. System.out.println("Inserted " + insertCount + " records into SC030 table.");
55. } **catch** (SQLException e) {
56. conn.rollback();
57. e.printStackTrace();
58. }
59. } **catch** (IOException | SQLException e) {
60. e.printStackTrace();
61. }
62. }

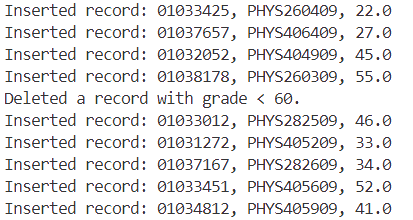
删除操作程序如下：

1. **public** **static** **void** deleteLowGrades() {
2. **int** deleteCount = 200; // 每次删除一条数据，并重复200次
3. **try** (Connection conn = getConnection();
4. Statement statement = conn.createStatement()) {
5. conn.setAutoCommit(**true**);
6. System.out.println("Start deleting records...");
8. **for** (**int** i = 0; i < deleteCount; i++) {
9. String deleteSQL = "DELETE FROM SC030 WHERE ctid IN (SELECT ctid FROM SC030 WHERE grade < 60 LIMIT 1)";
10. **int** rowsDeleted = statement.executeUpdate(deleteSQL);
11. System.out.println("Deleted a record with grade < 60.");
12. **if** (rowsDeleted > 0) {
13. System.out.println("Deleted a record with grade < 60.");
14. conn.commit();
15. } **else** {
16. System.out.println("No records to delete with grade < 60 remaining.");
17. **break**;
18. }
19. **try** {
20. Thread.sleep(100); // sleep for a brief moment (100 milliseconds)
21. } **catch** (InterruptedException e) {
22. Thread.currentThread().interrupt();
23. }
24. }
25. conn.commit();
26. System.out.println("Deletion completed.");
27. } **catch** (SQLException e) {
28. e.printStackTrace();
29. }
30. }

采用多线程的方式执行

1. ExecutorService executor = Executors.newFixedThreadPool(2); // 创建一个包含两个线程的线程池
2. executor.submit(openGaussDemo::insert\_sc); // 提交插入任务
3. executor.submit(openGaussDemo::deleteLowGrades); // 提交删除任务
5. executor.shutdown(); // 关闭线程池

结果如下，可以观察到是并行执行的



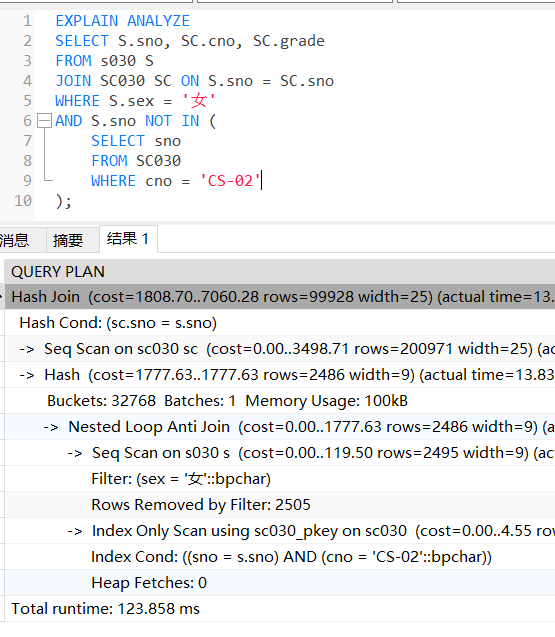
2．在 S×××表中补充数据至约 5000 行，在 C×××表中补充数据至约 1000

行，在 SC×××表中补充数据至约 200000 行。尝试为三、1.中的部分查询（不

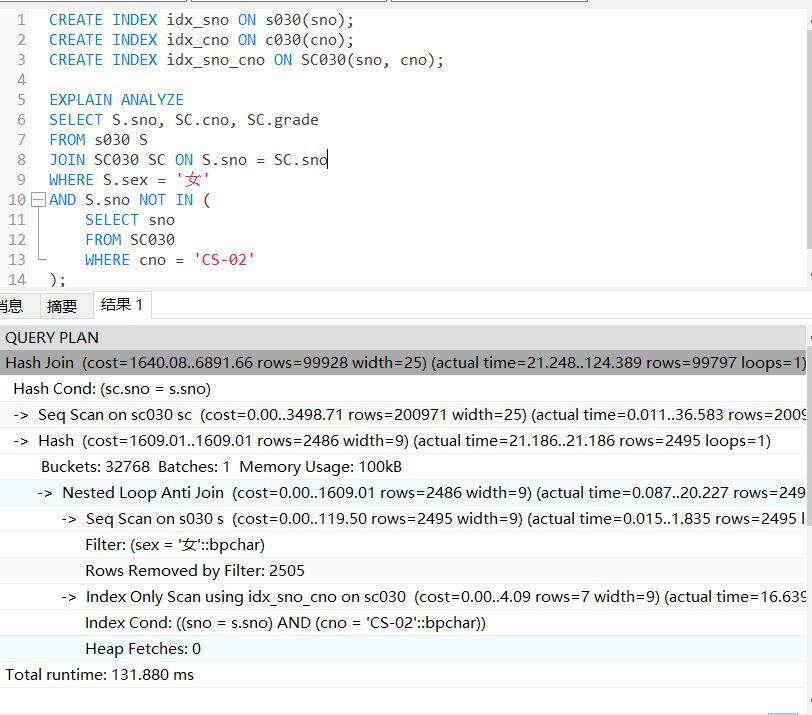
少于 3 个）编写不同的 SQL 语句实现，并分析其运行效率。如果可能，请尝试

给出可提高查询效率的改进方法。

(1) 查询未选修课程“CS-02”的女生学号及其已选各课程编号、成绩。

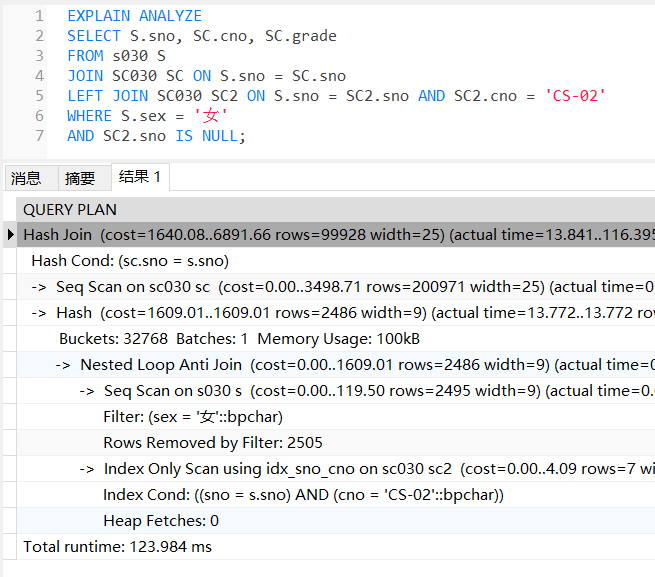


优化一：添加索引，可以看到加入索引后时间大幅缩短

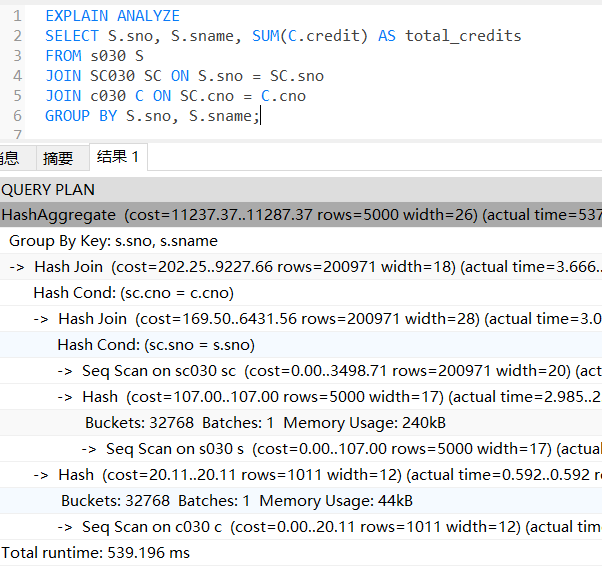


优化二：使用 LEFT JOIN 来查找那些没有匹配 cno = 'CS-02' 记录的 sno。

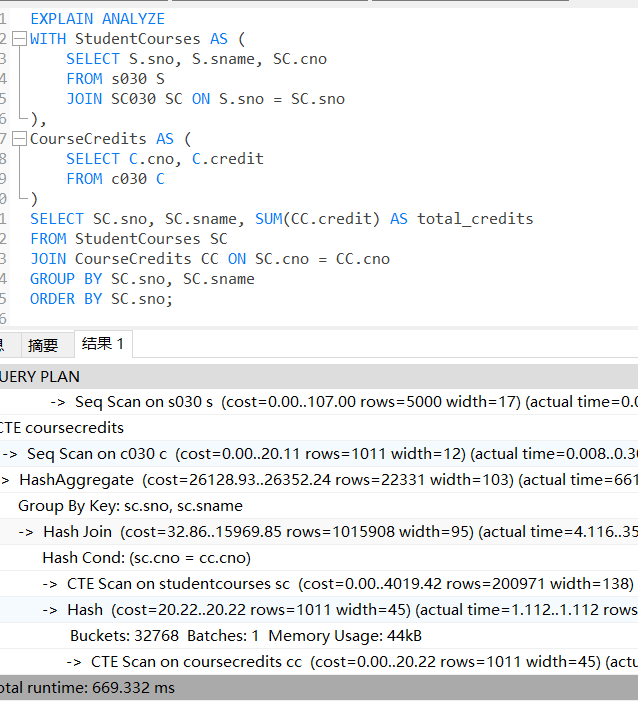
SC2.sno IS NULL 用来筛选没有匹配 CS-02 的学生。



（2）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



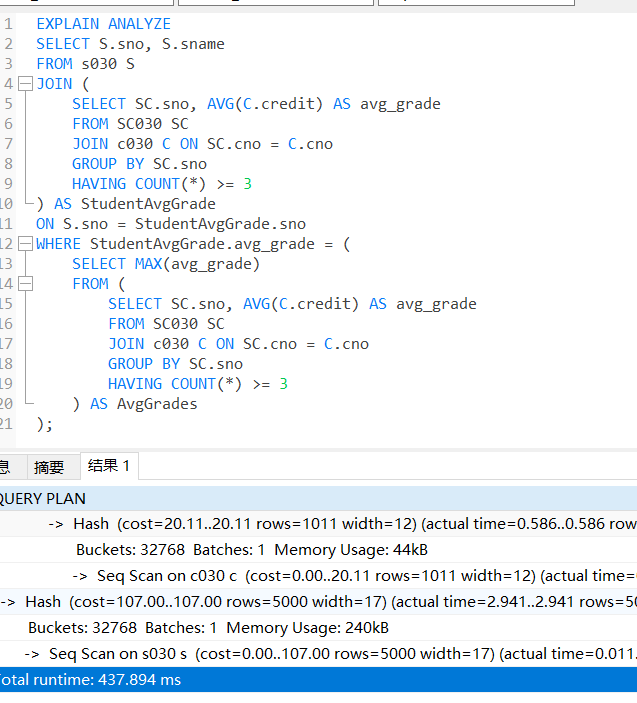
优化后，使用WITH 子句引入临时结果集，但是时间反而变长，猜测是因为数据量过大反而会增加开销



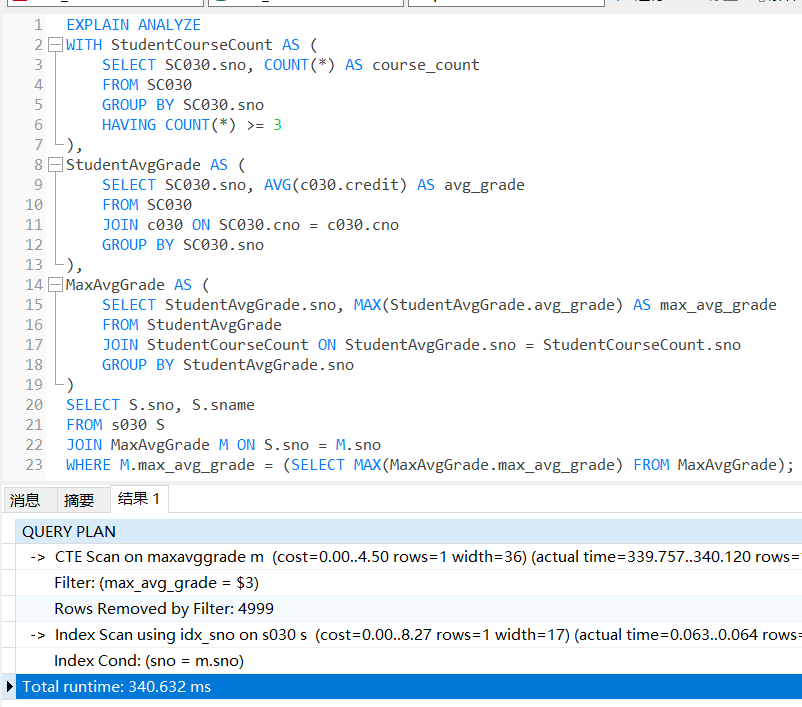
（3）查询选修了 3 门以上课程（包括 3 门）的学生中平均成绩最高的同学学

号及姓名。

不使用with子句



使用with子句，可以看到有明显改善



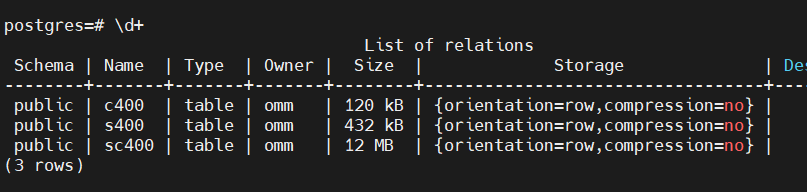
五、完成上述实验内容后，对数据库进行备份，并交给另一位同学进行恢复

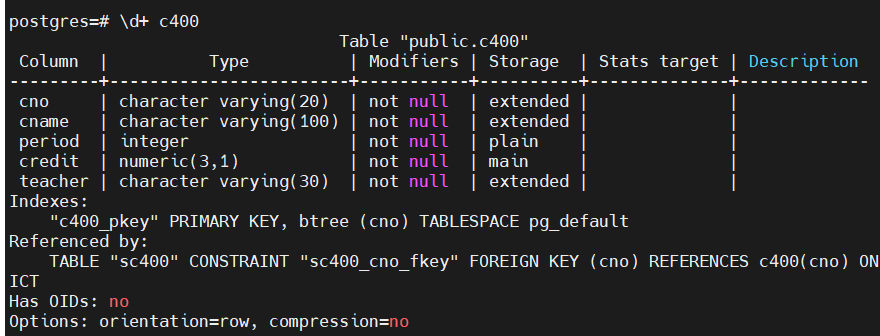
实验。在成功恢复其他同学交付的数据库备份后，分析其表设计合理性及生成

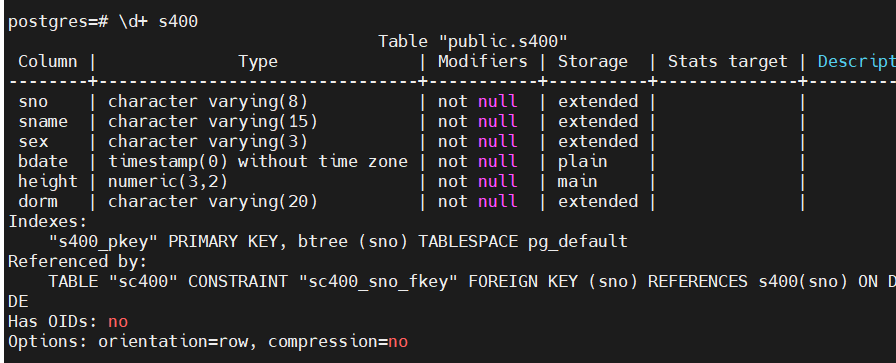
的数据质量，将相应结果截屏图保存，并写入实验报告中。

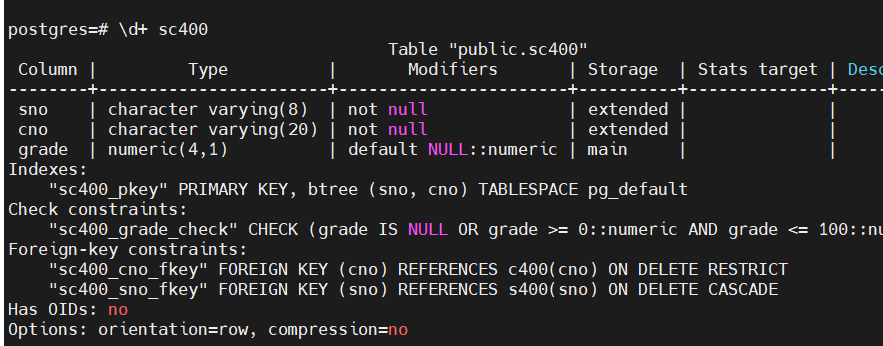
恢复的人的姓名：徐远帆，学号：2216113400

恢复后表的数据与定义如下：









评价：总体数据定义较好，只是在定义course表时使用int类型会使内存偏大，应该使用smallint或tinyint类型

六、我的爬虫程序

以爬选课信息为例

1. **def** \_\_init\_\_(self) -> None:
2. self.session = session
3. self.session.get('http://ehall.xjtu.edu.cn/new/index.html?browser=no')
4. self.session.get(
5. 'http://ehall.xjtu.edu.cn/portal/html/select\_role.html',
6. params={
7. 'appId': 4768574631264620,
8. },
9. )
10. resp = self.session.get(
11. 'http://ehall.xjtu.edu.cn/appMultiGroupEntranceList',
12. params={
13. 'r\_t': get\_timestamp(),
14. 'appId': 4768574631264620,
15. 'param': ''
16. },
17. )
19. data=resp.json()
20. target\_url = ''
21. **for** group **in** data['data']['groupList']:
22. **if** group['groupName'] == '移动应用学生':
23. target\_url = group['targetUrl']
24. self.session.get(target\_url)
25. = self.session.post(
26. 'http://ehall.xjtu.edu.cn/jwapp/sys/cjcx/modules/cjcx/xscjcx.do',
27. data={
28. 'querySetting': [
29. {"name": "SFYX", "caption": "是否有效", "linkOpt": "AND", "builderList": "cbl\_m\_List",
30. "builder": "m\_value\_equal", "value": "1", "value\_display": "是"},
31. ],
32. 'pageSize': self.NUM,
33. 'pageNumber': 1,
34. }
35. )
36. grade = resp.json()['datas']['xscjcx']['rows']

然后进行处理数据，导入csv文件