数据库实验

基础配置

1. 安装

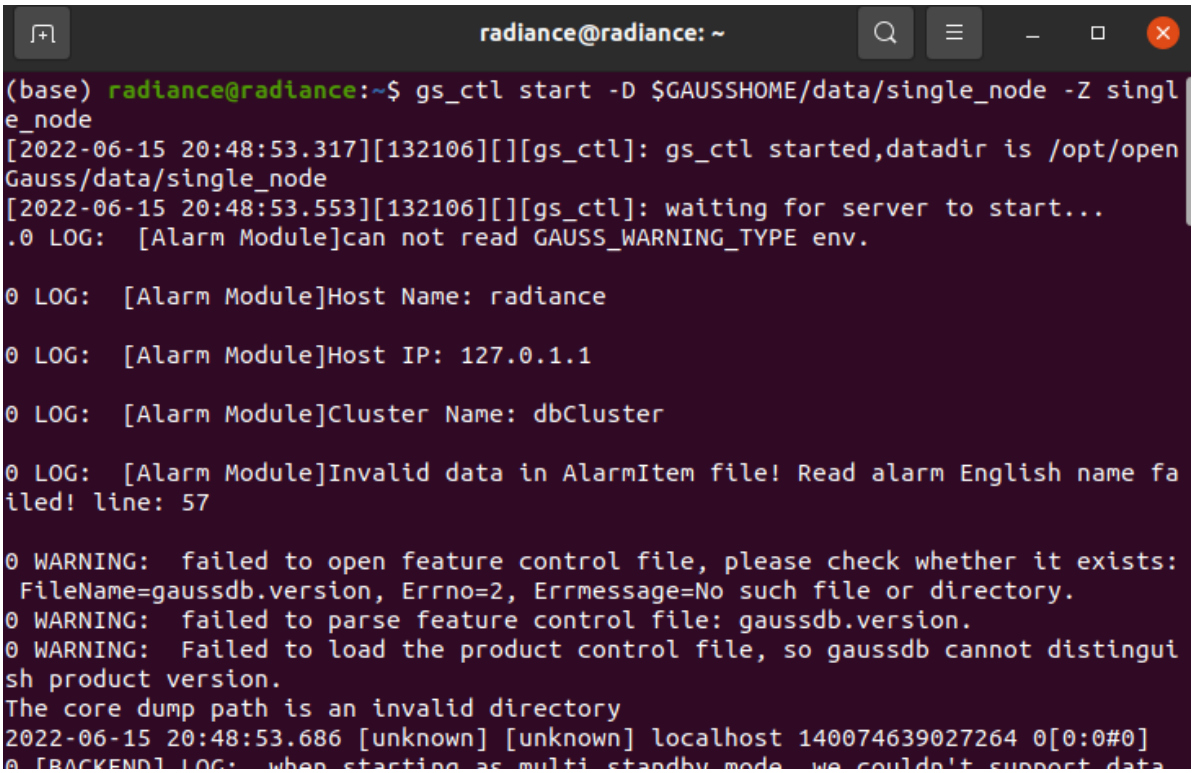
下载openGauss极简版

根据链接进行安装（安装过程略）

2. 启动

在本机上启动openGauss服务

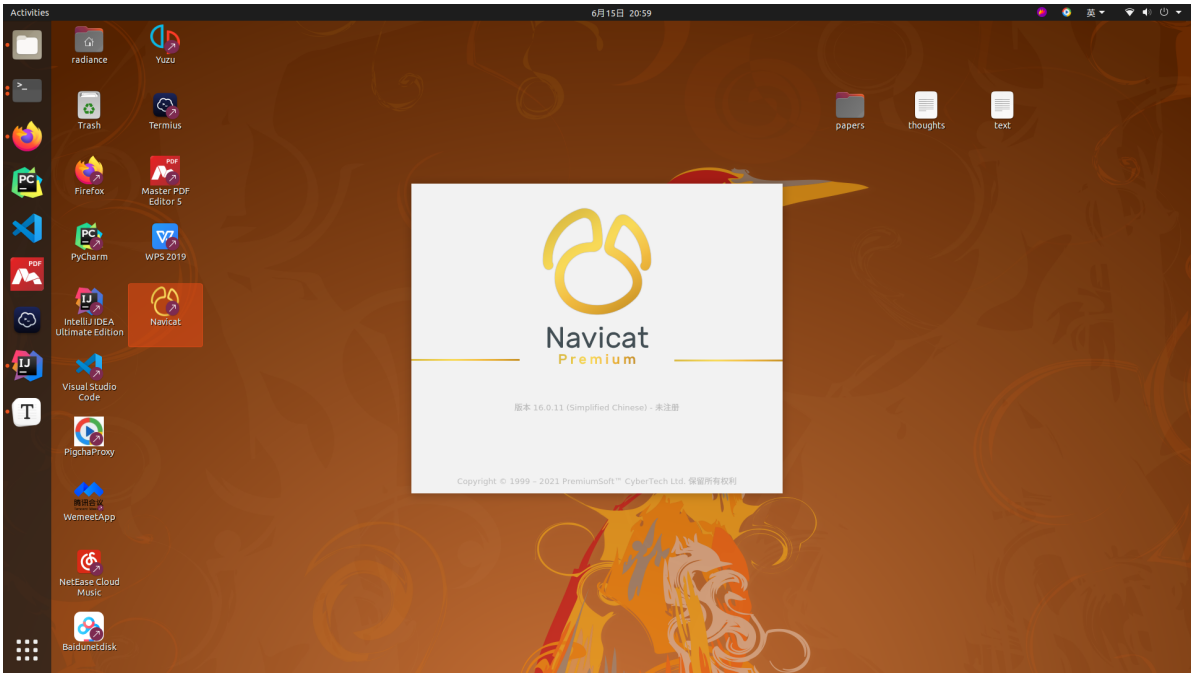
`gs_ctl start -D $GAUSSHOME/data/single_node -Z single_node`



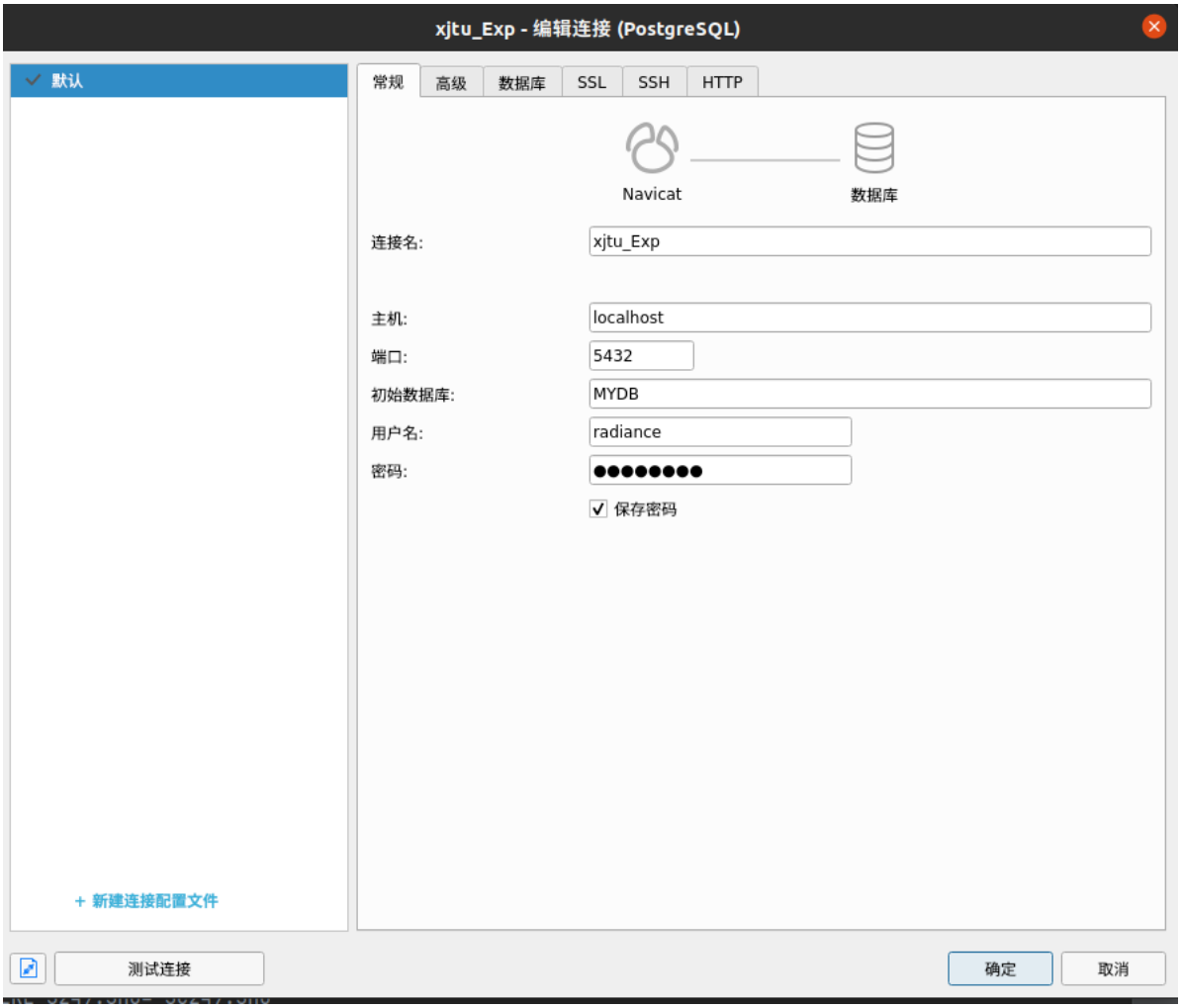
3. 连接

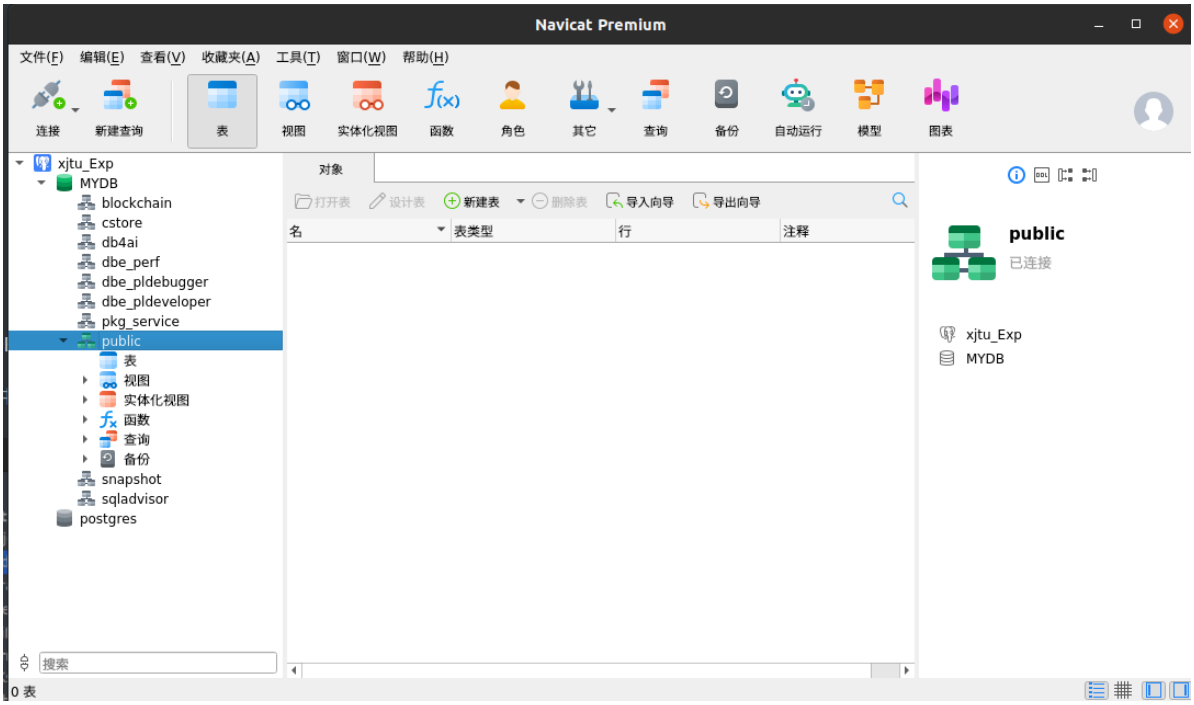
通过navicat连接

启动navicat软件



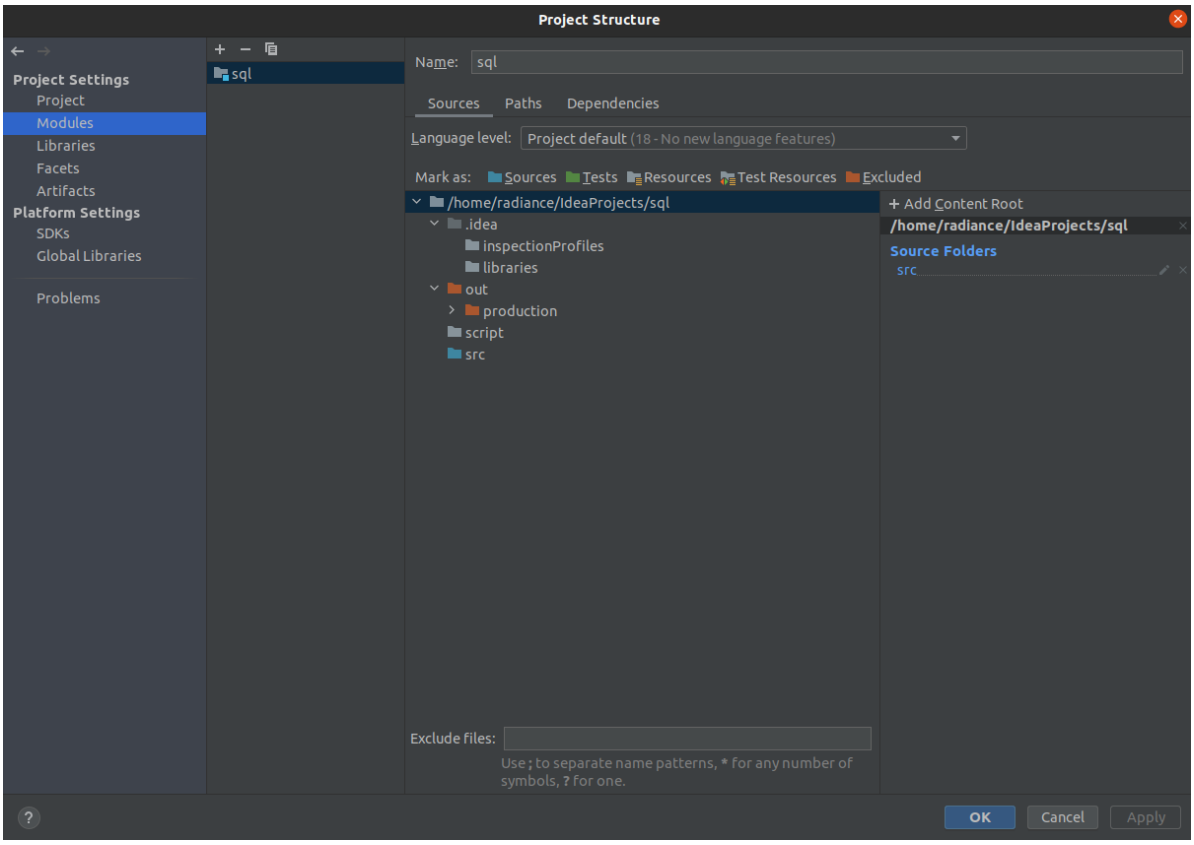
在navicat中连接数据库





通过JDBC连接

在IDEA中配置项目，添加库文件



通过GetConnection函数连接

```
    public static Connection GetConnection(String username, String passwd, String db, Integer port) {
        String driver = "org.postgresql.Driver";
        String sourceURL = String.format("jdbc:postgresql://localhost:%d/%s", port, db);
        Connection conn;
        try {
```

```java
            Class.forName(driver);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

        try {
            //创建数据库连接。
            conn = DriverManager.getConnection(sourceURL, username, passwd);
            System.out.println("Connection succeed!");
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

        return conn;
    }
```

```
Jun 08, 2022 6:15:22 PM org.postgresql.core.v3.ConnectionFactoryImpl
openConnectionImpl
INFO: [a85e2185-e799-490e-9860-75368f40aaca] Try to connect. IP: localhost:5432
Jun 08, 2022 6:15:22 PM org.postgresql.core.v3.ConnectionFactoryImpl
openConnectionImpl
INFO: [127.0.0.1:52572/ocalhost/127.0.0.1:5432] Connection is established. ID:
a85...
Jun 08, 2022 6:15:22 PM org.postgresql.core.v3.ConnectionFactoryImpl
openConnectionImpl
INFO: Connect complete. ID: a85...
Connection succeed!
```

数据库基本操作

创建基本表

```java
CREATE TABLE IF NOT EXISTS S249
    (Sno Integer PRIMARY KEY, Sname VARCHAR(32), Sex Char(4), BDATE Date, Height
Number, Dorm VARCHAR(32));
CREATE TABLE IF NOT EXISTS C249
    (Cno VARChar(16) PRIMARY KEY, Cname VARCHAR(32), Period Integer, Credit
Float, Teacher VARCHAR(32));
CREATE TABLE IF NOT EXISTS SC249
    (Sno Integer, Cno VARChar(16), Grade Number,
    PRIMARY KEY(Sno, Cno), Foreign Key(Sno) references S249(Sno), Foreign
Key(Cno) references C249(Cno));
```

插入基础数据

```java
INSERT INTO S249 VALUES
(1032010,'王涛','男','2002-4-5',1.72,'东14舍221'),
(1032023,'孙文','男','2003-6-10',1.8,'东14舍221'),
(1032001,'张晓梅','女','2003-11-17',1.58,'东1舍312'),
(1032005,'刘静','女','2002-1-10',1.63,'东1舍312'),
(1032112,'董蔚','男','2003-12-20',1.71,'东14舍221'),
```

```
(3031011,'王倩','女','2002-2-20',1.66,'东2舍104'),
(3031014,'赵思扬','男','2001-6-6',1.85,'东18舍421'),
(3031051,'周剑','男','2001-5-8',1.68,'东18舍422'),
(3031009,'田婷','女','2002-8-11',1.6,'东2舍104'),
(3031033,'蔡明明','男','2002-3-12',1.75,'东18舍423'),
(3031056,'曹子衿','女','2003-12-15',1.65,'东2舍305');

INSERT INTO C249 VALUES
('CS-01','数据结构',60,3,'张军'),
('CS-02','计算机组成原理',80,4,'王亚伟'),
('CS-04','人工智数据能',40,2,'李蕾'),
('CS-05','深度学习',40,2,'崔均'),
('EE-01','信号与系统',60,3,'张明'),
('EE-02','数字逻辑电路',100,5,'胡海东'),
('EE-03','光电子学与光子学',40,2,'石韬');

INSERT INTO SC249 VALUES
(1032010,'CS-01',82),
(1032010,'CS-02',91),
(1032010,'CS-04',83.5),
(1032001,'CS-01',77.5),
(1032001,'CS-02',85),
(1032001,'CS-04',83),
(1032005,'CS-01',62),
(1032005,'CS-02',77),
(1032005,'CS-04',82),
(1032023,'CS-01',55),
(1032023,'CS-02',81),
(1032023,'CS-04',76),
(1032112,'CS-01',88),
(1032112,'CS-02',91.5),
(1032112,'CS-04',86),
(1032112,'CS-05',NULL),
(3031033,'EE-01',93),
(3031033,'EE-02',89),
(3031009,'EE-01',88),
(3031009,'EE-02',78.5),
(3031011,'EE-01',91),
(3031011,'EE-02',86),
(3031051,'EE-01',78),
(3031051,'EE-02',58),
(3031014,'EE-01',79),
(3031014,'EE-02',71);
```

查询操作

```
-- 1
SELECT Cno, Cname FROM C249 WHERE Cno LIKE CONCAT('EE', '%');

-- 2
SELECT SC249.Sno, SC249.Cno, SC249.Grade FROM SC249 JOIN (SELECT S249.SNO FROM
S249 WHERE Sex = '女') AS T
    ON SC249.Sno=T.Sno WHERE T.SNO NOT IN (SELECT Sno FROM SC249 WHERE Cno = 'CS-
01');

-- 3
SELECT * FROM S249 WHERE Bdate BETWEEN '2000-01-01' and '2001-12-31';
```

```sql
-- 4
SELECT S249.Sno, S249.Sname, SUM(C249.Credit)
FROM S249 JOIN
    (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS SC
ON S249.Sno=SC.Sno
JOIN C249 ON C249.Cno=SC.Cno
GROUP BY S249.Sno;

-- 5
-- SELECT Sno, Grade FROM SC249 WHERE Cno='CS-02'  ORDER BY Grade DESC LIMIT 1,1;
SELECT Sno, Grade FROM SC249 WHERE Cno='CS-02' AND Grade IN(
SELECT MAX(Grade) FROM SC249 WHERE Cno='CS-02' AND Grade !=
(SELECT MAX(Grade) FROM SC249 WHERE Cno='CS-02'))
```

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 结果 4

cno	cname
EE-01	信号与系统
EE-02	数字逻辑电路
EE-03	光电子学与光子学

sno	cno	grade
3031011	EE-01	91
3031011	EE-02	86
3031009	EE-01	88
3031009	EE-02	78.5

sno	sname	sex	bdate	height	dorm
3031014	赵思扬	男	2001-06-06 00:00:00	1.85	东18舍421
3031051	周剑	男	2001-05-08 00:00:00	1.68	东18舍422

sno	sname	sum
3031033	蔡明明	8
1032005	刘静	9
3031051	周剑	3
1032001	张晓梅	9
1032023	孙文	6
3031009	田婷	8
1032112	董蔚	9
3031011	王倩	8
1032010	王涛	9
3031014	赵思扬	8

信息 | 摘要 | 结果 1 | 解释 1 | 解释 2

sno	grade
1032010	91

```sql
-- 6 TODO: Need to be optimized
SELECT Sno, Sname , AvgGrade FROM
```

```sql
    (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade) AvgGrade
    FROM S249 JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL) AS SC
    ON S249.Sno=SC.Sno GROUP BY SC.Sno, S249.Sname) AS Tmp
WHERE AvgGrade > (SELECT AvgGrade FROM
                        (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade)
AvgGrade
                                FROM S249
                                JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL)
AS SC
                                ON S249.Sno=SC.Sno
                                GROUP BY SC.Sno, S249.Sname)  WHERE Sname='王涛')
ORDER BY Sno DESC;

-- 7
SELECT S249.Sname
FROM S249
WHERE NOT EXISTS
    (SELECT *
    FROM (SELECT Cno FROM C249 WHERE Cno LIKE CONCAT('CS', '%')) AS Cor
    WHERE NOT EXISTS
        (SELECT *
        FROM SC249
        WHERE S249.Sno= SC249.Sno
        AND SC249.Cno= Cor.Cno));


-- 8 TODO: Need to be optimized
SELECT ST.Sno, Sname, AvgGrade
FROM
(    (SELECT S249.Sno Sno
    FROM S249 JOIN SC249 SC
    ON S249.Sno=SC.Sno
    GROUP BY S249.Sno HAVING COUNT(SC.Cno)>=3) AS ST
JOIN
    (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade) AvgGrade
       FROM S249 JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL) AS SC
       ON S249.Sno=SC.Sno
       GROUP BY SC.Sno, S249.Sname) AS AV
ON ST.Sno=AV.Sno) ORDER BY AvgGrade DESC LIMIT 1;
```

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 结果 4 | 结果 5 | 结果 6 | 结果 7 | 结果 8

sno	sname	avggrade
3031033	蔡明明	91.0000000000000000
3031011	王倩	88.5000000000000000
1032112	董蔚	88.5000000000000000

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 结果 4 | 结果 5 | 结果 6 | 结果 7 | 结果 8

sname
董蔚

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 结果 4 | 结果 5 | 结果 6 | 结果 7 | 结果 8

sno	sname	avggrade
1032112	董蔚	88.5000000000000000

插入数据

```
INSERT INTO S249 VALUES
(0032005,'刘竞','男','1993-12-10',1.75,'东14舍312');

INSERT INTO C249 VALUES
('CS-03','离散数学',64,4,'陈建明');
```

sno	sname	sex	bdate	height	dorm
1032010	王涛	男	2002-04-05 00:00:00	1.72	东14舍221
1032023	孙文	男	2003-06-10 00:00:00	1.8	东14舍221
1032001	张晓梅	女	2003-11-17 00:00:00	1.58	东1舍312
1032005	刘静	女	2002-01-10 00:00:00	1.63	东1舍312
1032112	董蔚	男	2003-12-20 00:00:00	1.71	东14舍221
3031011	王倩	女	2002-02-20 00:00:00	1.66	东2舍104
3031014	赵思扬	男	2001-06-06 00:00:00	1.85	东18舍421
3031051	周剑	男	2001-05-08 00:00:00	1.68	东18舍422
3031009	田婷	女	2002-08-11 00:00:00	1.6	东2舍104
3031033	蔡明明	男	2002-03-12 00:00:00	1.75	东18舍423
3031056	曹子衿	女	2003-12-15 00:00:00	1.65	东2舍305
32005	刘竞	男	1993-12-10 00:00:00	1.75	东14舍312

cno	cname	period	credit	teacher
CS-01	数据结构	60	3	张军
CS-02	计算机组成原理	80	4	王亚伟
CS-04	人工智能	40	2	李蕾
CS-05	深度学习	40	2	崔均
EE-01	信号与系统	60	3	张明
EE-02	数字逻辑电路	100	5	胡海东
EE-03	光电子学与光子学	40	2	石韬
CS-03	离散数学	64	4	陈建明

删除数据

```
DELETE FROM SC249  S249
WHERE Sno IN
   (
   SELECT S249.Sno Sno
   FROM S249 JOIN
       (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS SC
   ON S249.Sno=SC.Sno
   JOIN C249 ON C249.Cno=SC.Cno
   GROUP BY S249.Sno HAVING SUM(C249.Credit)>60
) ;
```

信息 摘要

```
DELETE FROM SC249  S249
WHERE Sno IN
   (
   SELECT S249.Sno Sno
   FROM S249 JOIN
       (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS SC
   ON S249.Sno=SC.Sno
   JOIN C249 ON C249.Cno=SC.Cno
   GROUP BY S249.Sno HAVING SUM(C249.Credit)>60
)
> Affected rows: 0
> 时间: 0.001 秒
```

更新数据

```
UPDATE C249 SET Period=64, Credit=Credit+1 WHERE Teacher='张明';
```

cno	cname	period	credit	teacher
CS-01	数据结构	60	3	张军
CS-02	计算机组成原理	80	4	王亚伟
CS-04	人工智能	40	2	李蕾
CS-05	深度学习	40	2	崔均
EE-02	数字逻辑电路	100	5	胡海东
EE-03	光电子学与光子学	40	2	石韬
CS-03	离散数学	64	4	陈建明
EE-01	信号与系统	64	4	张明

建立试图

```
CREATE VIEW V1 AS SELECT * FROM S249 WHERE Sex='男' AND Dorm LIKE CONCAT('东18',
'%');

CREATE VIEW V2 AS
    SELECT C249.Cno, C249.Cname, AVG(SC249.Grade)
    FROM C249 JOIN SC249 ON C249.Cno = SC249.Cno WHERE Teacher='张明' GROUP BY
C249.Cno, C249.Cname;

CREATE VIEW V3 AS
    SELECT S249.*
    FROM (S249 JOIN SC249 ON S249.Sno=SC249.Sno) JOIN C249 ON SC249.Cno=C249.Cno
    WHERE C249.Cname='人工智能';
```

对象 | v1 @ MYDB.public (xjt... | v2 @ MYDB.public (xjt... | v3 @ MYDB.public (xjt...

开始事务 文本 筛选 排序 导出 数据生成 创建图表

sno	sname	sex	bdate	height	dorm
3031014	赵思扬	男	2001-06-06 00:00:00	1.85	东18舍421
3031051	周剑	男	2001-05-08 00:00:00	1.68	东18舍422
3031033	蔡明明	男	2002-03-12 00:00:00	1.75	东18舍423

对象 | v1 @ MYDB.public (xjt... | v2 @ MYDB.public (xjt... | v3 @ MYDB.public (xjt...

开始事务 | 文本 | 筛选 | 排序 | 导出 | 数据生成 | 创建图表

cno	cname	avg
EE-01	信号与系统	85.8000000000000000

对象 | v1 @ MYDB.public (xjt... | v2 @ MYDB.public (xjt... | v3 @ MYDB.public (xjt...

开始事务 | 文本 | 筛选 | 排序 | 导出 | 数据生成 | 创建图表

sno	sname	sex	bdate	height	dorm
1032010	王涛	男	2002-04-05 00:00:00	1.72	东14舍221
1032001	张晓梅	女	2003-11-17 00:00:00	1.58	东1舍312
1032005	刘静	女	2002-01-10 00:00:00	1.63	东1舍312
1032023	孙文	男	2003-06-10 00:00:00	1.8	东14舍221
1032112	董蔚	男	2003-12-20 00:00:00	1.71	东14舍221

补充数据

JDBC

- 编写`ExecCommand`函数，可以执行普通操作，如有错则则会提示

```
public static void ExecCommand(Connection conn, String command) throws
InterruptedException {
    Statement stmt = null;
    try {
        stmt = conn.createStatement();
        stmt.execute(command);
        stmt.close();
        TimeUnit.MICROSECONDS.sleep(1000);

    } catch (SQLException e) {
        System.out.println("Error occurs when executing " + command);
        if (stmt != null) {
            try {
                stmt.close();
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
        e.printStackTrace();
    }
}
```

- 编写`ExecSelect`函数，可以执行`SELECT`操作并输出查询结果

```
public static void ExecSelect(Connection conn, String sql) {

    Statement stmt =null;
    try {
        stmt = conn.createStatement();
        System.out.println("========================================");
        System.out.printf("Executing %s:%n",sql);
```

```
        ResultSet rs = stmt.executeQuery(sql);
        String str=null;
        while(rs.next()){
            str = "";
            for(int i=1;i<=rs.getMetaData().getColumnCount();i++){
                str += rs.getString(i)+",";
            }
            System.out.println(str);
        }
        if (str == null){
            System.out.println("Found empty!");
        }
        System.out.println("========================================");
        rs.close();
        stmt.close();
    } catch (SQLException e) {
        if (stmt != null) {
            try {
                stmt.close();
            }
            catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
        System.out.println("Error!");
        e.printStackTrace();
        System.out.println("========================================");

    }
}
```

输出示例：

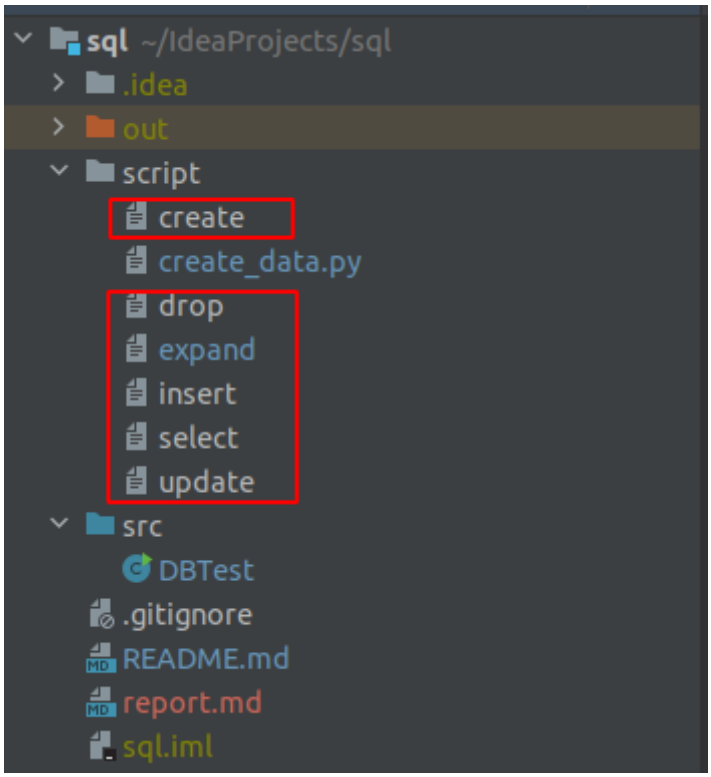
```
========================================
Executing SELECT Cno, Cname FROM C249 WHERE Cno LIKE CONCAT('EE', '%'); :
EE-01,信号与系统,
EE-02,数字逻辑电路,
EE-03,光电子学与光子学,
========================================
========================================
Executing  SELECT SC249.Sno, SC249.Cno, SC249.Grade FROM SC249 JOIN (SELECT S249.SNO FROM S249 WHERE Sex = '女') AS T     ON SC249.Sno=T.Sno WHERE T.SNO NOT IN (SELECT Sno FROM SC249 WHERE Cno
3031011,EE-01,91,
3031011,EE-02,86,
3031009,EE-01,88,
3031009,EE-02,78.5,
========================================
```

- 编写`ExecFile`函数，可以将文件内的非注释行读入并执行sql语句

```
public static void ExecFile(Connection conn, String filename) throws IOException,
InterruptedException {

    FileReader fr=new FileReader(filename);
    BufferedReader br=new BufferedReader(fr);
    String line;
    String buf="";

    while ((line=br.readLine())!=null) {
        if (!line.contains("--")){
            buf = buf + line + " ";
        }
        if (line.contains(";")){
            if (buf.toLowerCase().contains("select") &
!buf.toLowerCase().contains("create")){
                ExecSelect(conn, buf);
```

```
            }
            else {
                System.out.println(buf);
                ExecCommand(conn, buf);
            }
            buf = "";
        }
    }
    br.close();
    fr.close();
}
```

- 可以在`script/`目录下创建若干sql命令文件，通过`ExecFile`函数读入并执行




```
public static void main(String[] args){
        //创建数据库连接。
        String USERNAME = "radiance";
        String PASSWORD = "Sql123456";
        String DB = "MYDB";
        Integer PORT = 5432;
        try {
            Connection conn = GetConnection(USERNAME, PASSWORD, DB, PORT);
            assert conn != null;

            ExecFile(conn, "./script/drop");
            ExecFile(conn, "./script/create");
            ExecFile(conn, "./script/insert");
            ExecFile(conn, "./script/expand");
            ExecFile(conn, "./script/select");
            ExecFile(conn, "./script/update");

            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (IOException | InterruptedException e) {
```

```
            throw new RuntimeException(e);
        }
    }
```

DBTest.java expand

```
INSERT INTO S249 VALUES
(1033002, '汤或书', '女', '1999-07-20', 1.51, '东4舍312'),
(1033005, '秦又定', '女', '1999-11-25', 1.78, '西14舍103'),
(1033008, '费克夫', '女', '2001-05-10', 1.75, '西16舍1511'),
(1033009, '费把母', '女', '1999-06-15', 1.4, '西3舍622'),
(1033011, '康又年', '男', '2000-11-08', 1.43, '东16舍1014'),
(1033014, '云门将', '男', '1999-07-06', 1.72, '东20舍1613'),
(1033017, '柳亲许', '女', '1999-12-30', 1.68, '东13舍715'),
(1033020, '乐新', '男', '2001-03-27', 1.71, '东4舍1518'),
(1033022, '郝常望', '男', '2001-05-09', 1.5, '西1舍65'),
(1033025, '廉可话', '女', '2000-10-14', 1.76, '东6舍311'),
(1033026, '汤记或', '女', '2000-03-10', 1.57, '西6舍419'),
(1033028, '费气', '男', '2001-07-16', 1.77, '东19舍139'),
(1033031, '冯问两', '男', '2001-03-09', 1.59, '西11舍1112'),
(1033032, '费父色', '女', '1999-02-12', 1.98, '东8舍1919'),
(1033034, '冯玉被', '男', '1999-09-10', 1.9, '西14舍1019'),
(1033037, '伍次被', '男', '1999-02-01', 1.88, '东13舍614'),
(1033040, '奚德因', '男', '2000-02-27', 1.72, '东6舍514'),
(1033042, '施龙行', '男', '1999-10-14', 1.86, '西14舍148'),
(1033045, '苏国使', '女', '1999-04-26', 1.88, '东13舍113'),
(1033046, '赵法想', '男', '1999-05-12', 1.72, '东11舍713'),
(1033049, '周高少', '女', '2000-02-06', 1.49, '东18舍147'),
(1033052, '戚年很', '男', '2001-12-10', 1.48, '东18舍48'),
(1033054, '花己给', '男', '2000-10-02', 1.6, '西10舍1610'),
```

Python 随机生成数据

- 在`script/`目录下创建`expand`文件，在其中写入随机生成的命令。
- 为保证`SC`表中**外键依赖**，将生成的`sno`和`cno`储存，在生成`SC`表随机数据时将其随机组合作为主键。
- 为保证`SC`表中**主键唯一**，考虑到`python dict`底层为`HASH`，使用`dict`数据结构储存`主键`
  - 若`dict.get(主键) == True`，说明该主键已生成过，则重新随机生成。
  - 使用迭代器，优化代码结构。

```python
import random
import time
import os

S_LEN = 5000
C_LEN = 1000
SC_LEN = 30000

SNO_START = int(1033e3)

# yyyy, mm, dd, h, m ,s
date1 = (1999, 1, 1, 0, 0, 0, -1, -1, -1)
time1 = time.mktime(date1)
date2 = (2002, 1, 1, 0, 0, 0, -1, -1, -1)
time2 = time.mktime(date2)
```

```
first_name = ["赵", "钱", "孙", "李", "周", "吴", "郑", "王", "冯", "陈", "褚", "卫",
"蒋", "沈", "韩", "杨", "朱", "秦", "尤", "许", "何",]
last_name = ['玉', '明', '龙', '芳', '军', '玲', '', '立', '玲', '', '国', "地",
"为", "子", "中", "", "", "", "国", "年", "着", "就",]
genders = ['女', '男']
dorms = ['东', '西']
sno = SNO_START

first_class = ['深度', '爱情', '经济', '电机', '电路', '睡眠', '操作', '数据库', '',
'','高等', '概率','初级','中等']
last_class = ['学习', '课程', '教学', '项目', '基础', '技术','本领','概论','综述']
deps = ['CS', 'EE', 'SC', 'HH', 'AI', 'HW','FF','BB','CC',
'DD','EE','FF','GG','HH','MM']

snos = []
cnos = []

log = open(os.path.join('.', 'expand'), 'w')


def gen_name():
    while True:
        full_name = random.choice(first_name) + random.choice(last_name) +
random.choice(last_name)
        if len(full_name) > 1:
            return full_name


def record(msg):
    print(msg, end='')
    log.write('%s' % msg)
    log.flush()


record('INSERT INTO S249 VALUES \n')
for i in range(S_LEN):
    count = random.randint(1, 3)
    sno = sno + count
    full_name = gen_name()
    random_time = random.uniform(time1, time2)  # uniform返回随机实数 time1 <= time
< time2
    birthday = time.strftime("%Y-%m-%d", (time.localtime(random_time)))

    gender = random.choice(genders)
    height = random.uniform(1.4, 2.0)
    height = round(height, 2)
    dorm = '%s%d舍%d%d' % (random.choice(dorms), random.randint(1, 20),
                           random.randint(1, 20), random.randint(1, 22),)
    snos.append(sno)
    if i != S_LEN - 1:
        record("({}, '{}', '{}', '{}', {}, '{}'),\n".format(sno, full_name,
gender, birthday, height, dorm))
    else:
        record("({}, '{}', '{}', '{}', {}, '{}');\n".format(sno, full_name,
gender, birthday, height, dorm))
record('\n\n')

# 迭代器
```

```
def cache(func):
    ca = {}
    while True:
        args = func()
        if not ca.get(args):
            ca[args] = True
            yield args


record('INSERT INTO C249 VALUES \n')
cno_gen = cache(lambda: ('%s-%d' % (random.choice(deps), random.randint(1,
100))))
for i in range(C_LEN):
    cno = next(cno_gen)

    class_name = random.choice(first_class) + random.choice(first_class) +
random.choice(last_class)
    full_name = gen_name()

    ctime = random.randrange(20, 60, 4)

    gender = random.choice(genders)
    credit = random.randrange(1, 13) / 2
    credit = round(credit, 1)
    cnos.append(cno)
    if i != C_LEN - 1:
        record("('{}', '{}', {}, {}, '{}'),\n".format(cno, class_name, ctime,
credit, full_name))
    else:
        record("('{}', '{}', {}, {}, '{}');\n".format(cno, class_name, ctime,
credit, full_name))

record('\n\n')

record('INSERT INTO SC249 VALUES \n')
key_gen = cache(lambda: (random.choice(snos), random.choice(cnos)))
for i in range(SC_LEN):
    key = next(key_gen)
    grade = random.randrange(80, 200) / 2
    grade = round(grade, 1)
    if i != SC_LEN - 1:
        record("({}, '{}', {}),\n".format(key[0], key[1], grade))
    else:
        record("({}, '{}', {});\n".format(key[0], key[1], grade))
record('\n\n')

record('-- Finish')

log.flush()
log.close()
```

插入数据

Python脚本生成部分数据写入`sciprt/expand`中

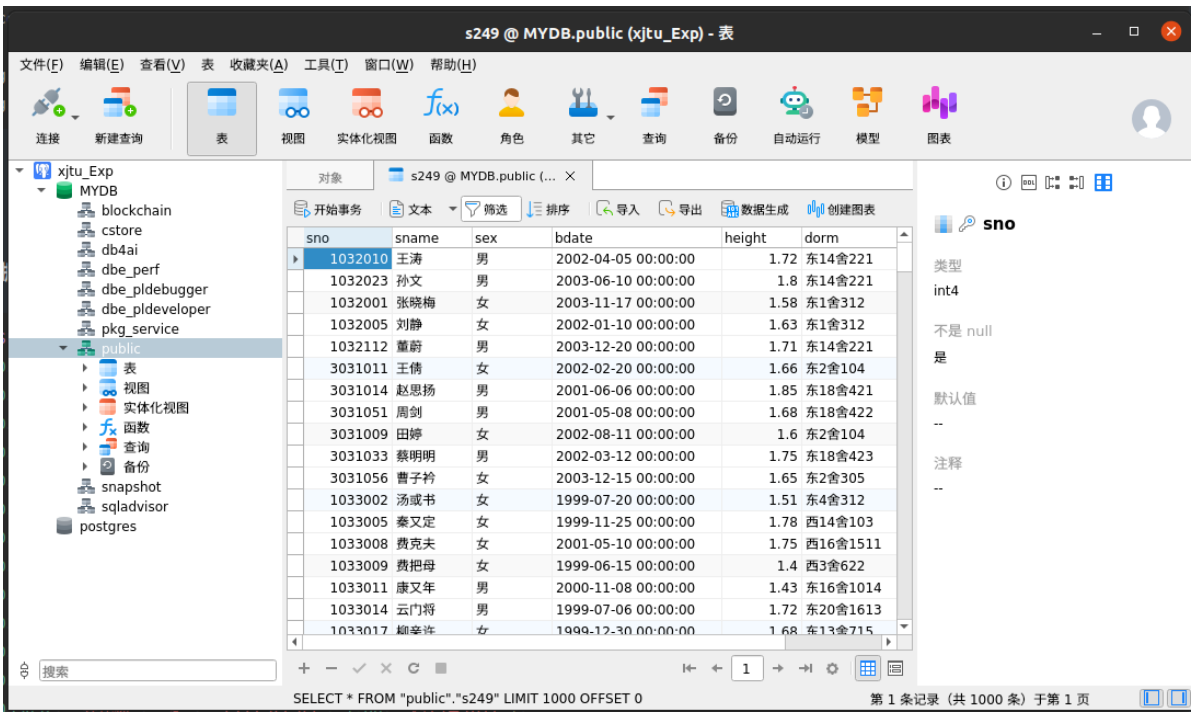
```sql
INSERT INTO S249 VALUES
(1033001, '陈或月', '男', '1999-11-08', 1.93, '西16舍163'),
(1033004, '吕太接', '女', '2001-05-15', 1.46, '东3舍121'),
(1033007, '郝德明', '男', '2000-04-25', 1.45, '东18舍137'),
(1033010, '马种', '男', '1999-02-02', 1.47, '东7舍1718'),
(1033013, '贺国加', '女', '1999-08-30', 1.68, '东8舍620'),
(1033016, '于问文', '女', '1999-12-08', 1.94, '西10舍127'),
(1033017, '孙几色', '男', '2000-08-01', 1.73, '西2舍612'),
(1033020, '穆车写', '女', '1999-02-07', 1.69, '西15舍15'),
(1033022, '奚往', '女', '2001-06-02', 1.66, '东4舍1915'),
(1033025, '方从放', '女', '2000-04-30', 1.5, '西19舍192'),
(1033028, '马下国', '女', '1999-06-30', 1.72, '西13舍35'),
(1033029, '郝走拉', '女', '2000-01-29', 1.84, '西16舍1119'),
(1033032java, '姜比物', '女', '1999-09-03', 1.66, '西13舍1817'),
(1033033, '金告就', '男', '2000-06-01', 1.62, '东15舍1620');

INSERT INTO C249 VALUES
('EE-42', '经济电机课程', 56, 4.0, '俞再光'),
('HW-5', '电路学习', 48, 2.0, '戚少位'),
('CS-14', '爱情项目', 20, 2.0, '康感下'),
('HH-33', '电路经济基础', 28, 5.5, '章军太'),
('EE-20', '电路教学', 48, 2.0, '陈教代'),
('SC-80', '操作电路基础', 48, 6.0, '和母公'),
('AI-31', '经济电机教学', 28, 2.0, '尹所或'),
('CS-45', '经济数据库学习', 36, 5.0, '平特直'),
('HH-38', '操作操作项目', 56, 3.0, '吴至日'),
('CS-29', '数据库学习', 40, 0.5, '堪着住'),
('AI-41', '数据库操作教学', 28, 5.5, '周车望');

INSERT INTO SC249 VALUES
(1033060, 'HH-75', 63.0),
(1033081, 'CS-45', 74.0),
(1033061, 'SC-50', 72.0),
(1033107, 'SC-33', 53.0),
(1033161, 'CS-80', 80.0),
(1033028, 'HH-45', 41.0),
(1033074, 'AI-79', 61.0),
(1033110, 'HH-100', 94.0),
(1033103, 'AI-27', 94.5),
(1033072, 'AI-23', 85.5),
(1033128, 'AI-69', 64.0),
(1033013, 'CS-67', 91.5),
(1033144, 'HH-27', 44.5),
(1033190, 'HH-79', 54.5),
(1033144, 'AI-69', 88.0),
(1033092, 'HH-38', 57.0),
(1033079, 'SC-17', 97.5),
(1033032, 'HH-27', 63.0),
(1033198, 'SC-3', 85.5),
(1033163, 'HW-81', 64.0),
(1033045, 'SC-80', 71.0),
(1033091, 'HH-45', 62.0),
(1033173, 'SC-50', 40.0),
```

```
(1033032, 'EE-58', 88.0),
(1033054, 'AI-31', 73.0),
(1033105, 'HH-73', 53.5),
(1033161, 'HH-27', 68.5),
(1033123, 'HW-5', 79.0);
```

通过`ExecFile`函数执行

```
            ExecFile(conn, "./script/expand");
```

在 Navicat中可以看到，数据通过JDBC正确写入表中



性能分析

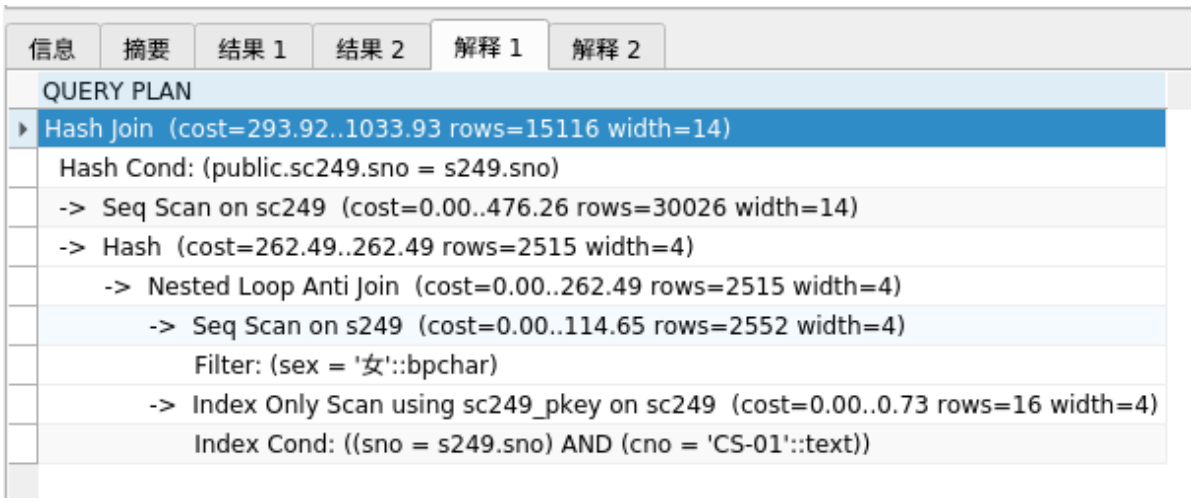
查询操作优化

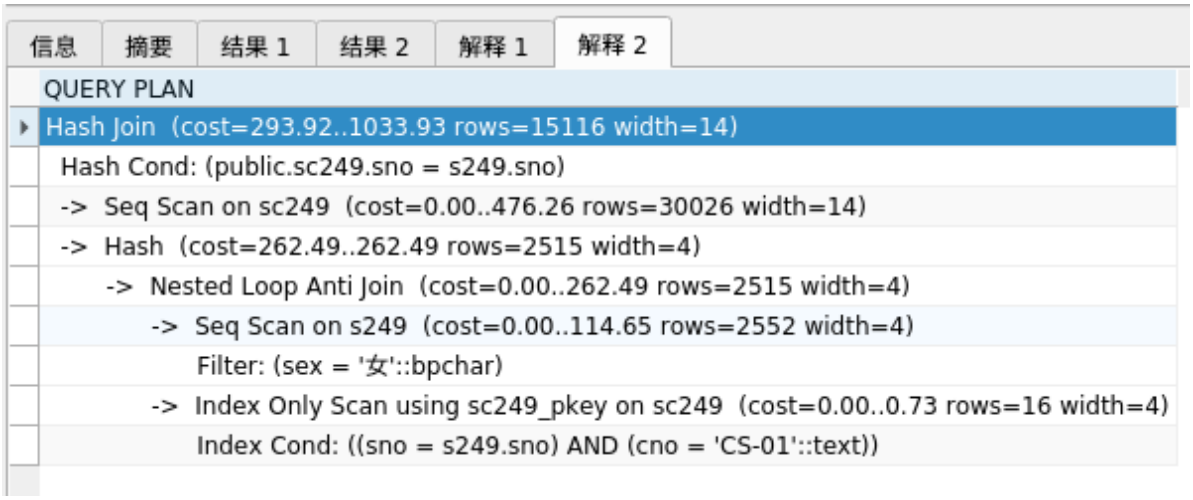
```
-- 2
SELECT SC249.Sno, SC249.Cno, SC249.Grade FROM SC249 JOIN (SELECT S249.SNO FROM
S249 WHERE Sex = '女') AS T
    ON SC249.Sno=T.Sno WHERE T.SNO NOT IN (SELECT Sno FROM SC249 WHERE Cno = 'CS-
01');

SELECT SC249.Sno, SC249.Cno, SC249.Grade FROM SC249 JOIN (SELECT S249.SNO FROM
S249 WHERE Sex = '女') AS T
    ON SC249.Sno=T.Sno WHERE NOT EXISTS (SELECT Sno FROM SC249 WHERE Cno = 'CS-
01' AND T.SNO = sc249.sno);
```

分析：

在本例上性能基本相同，但最好使用NOT EXISTS而不是NOT IN

原因是“如果查询语句使用了not in，那么对内外表都进行全表扫描，没有用到索引；而not exists的子查询依然能用到表上的索引。所以无论哪个表大，用not exists都比not in 要快”，具体可见参考链接。





参考：[sql中的in与not in,exists与not exists的区别]

```
-- 4
SELECT S249.Sno, S249.Sname, SUM(C249.Credit)
FROM S249 JOIN
    (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS SC
ON S249.Sno=SC.Sno
JOIN C249 ON C249.Cno=SC.Cno
GROUP BY S249.Sno;
-- --------------------------------
SELECT S249.Sno, S249.Sname, SUM(C249.Credit)
FROM C249 JOIN
    (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS SC
ON C249.Cno=SC.Cno
JOIN S249 ON S249.Sno=SC.Sno
GROUP BY S249.Sno;
-- --------------------------------
SELECT S249.Sno, S249.Sname, SUM(C249.Credit)
FROM S249, C249, (SELECT * FROM SC249 WHERE Grade>=60 AND Grade IS NOT NULL) AS
SC
WHERE S249.Sno=SC.Sno AND C249.Cno=SC.Cno
GROUP BY S249.Sno;
```

分析：

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 解释 1 | 解释 2 | 解释 3

```
QUERY PLAN
HashAggregate  (cost=1398.62..1448.73 rows=5011 width=29)
  Group By Key: s249.sno
  ->  Hash Join  (cost=198.41..1298.80 rows=19966 width=21)
        Hash Cond: ((sc249.cno)::text = (c249.cno)::text)
        ->  Hash Join  (cost=164.75..990.61 rows=19966 width=18)
              Hash Cond: (sc249.sno = s249.sno)
              ->  Seq Scan on sc249  (cost=0.00..551.33 rows=19966 width=9)
                    Filter: ((grade IS NOT NULL) AND (grade >= 60::numeric))
              ->  Hash  (cost=102.11..102.11 rows=5011 width=13)
                    ->  Seq Scan on s249  (cost=0.00..102.11 rows=5011 width=13)
        ->  Hash  (cost=21.07..21.07 rows=1007 width=13)
              ->  Seq Scan on c249  (cost=0.00..21.07 rows=1007 width=13)
```

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 解释 1 | 解释 2 | 解释 3

```
QUERY PLAN
HashAggregate  (cost=1398.62..1448.73 rows=5011 width=29)
  Group By Key: s249.sno
  ->  Hash Join  (cost=198.40..1298.79 rows=19966 width=21)
        Hash Cond: (sc249.sno = s249.sno)
        ->  Hash Join  (cost=33.66..859.51 rows=19966 width=12)
              Hash Cond: ((sc249.cno)::text = (c249.cno)::text)
              ->  Seq Scan on sc249  (cost=0.00..551.33 rows=19966 width=9)
                    Filter: ((grade IS NOT NULL) AND (grade >= 60::numeric))
              ->  Hash  (cost=21.07..21.07 rows=1007 width=13)
                    ->  Seq Scan on c249  (cost=0.00..21.07 rows=1007 width=13)
        ->  Hash  (cost=102.11..102.11 rows=5011 width=13)
              ->  Seq Scan on s249  (cost=0.00..102.11 rows=5011 width=13)
```

信息 | 摘要 | 结果 1 | 结果 2 | 结果 3 | 解释 1 | 解释 2 | 解释 3

```
QUERY PLAN
HashAggregate  (cost=1398.62..1448.73 rows=5011 width=29)
  Group By Key: s249.sno
  ->  Hash Join  (cost=198.41..1298.80 rows=19966 width=21)
        Hash Cond: ((sc249.cno)::text = (c249.cno)::text)
        ->  Hash Join  (cost=164.75..990.61 rows=19966 width=18)
              Hash Cond: (sc249.sno = s249.sno)
              ->  Seq Scan on sc249  (cost=0.00..551.33 rows=19966 width=9)
                    Filter: ((grade IS NOT NULL) AND (grade >= 60::numeric))
              ->  Hash  (cost=102.11..102.11 rows=5011 width=13)
                    ->  Seq Scan on s249  (cost=0.00..102.11 rows=5011 width=13)
        ->  Hash  (cost=21.07..21.07 rows=1007 width=13)
              ->  Seq Scan on c249  (cost=0.00..21.07 rows=1007 width=13)
```

```sql
-- 6
-- 优化前
SELECT Sno, Sname , AvgGrade FROM
```

```sql
    (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade) AvgGrade
    FROM S249 JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL) AS SC
    ON S249.Sno=SC.Sno GROUP BY SC.Sno, S249.Sname HAVING G) AS Tmp
WHERE AvgGrade > (SELECT AvgGrade FROM
                        (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade)
AvgGrade
                                FROM S249
                                JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL)
AS SC
                                ON S249.Sno=SC.Sno
                                GROUP BY SC.Sno, S249.Sname)  WHERE Sname='王涛')
ORDER BY Sno DESC;

-- 优化后
SELECT Sno, Sname , AvgGrade FROM
    (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade) AvgGrade
    FROM S249 JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL) AS SC ON
S249.Sno=SC.Sno
        GROUP BY SC.Sno, S249.Sname HAVING Avggrade >
            any(
                SELECT AVG(SC.Grade) AvgGrade FROM (SELECT * FROM SC249 WHERE
Grade IS NOT NULL) AS SC WHERE Sno=(SELECT Sno FROM S249 WHERE Sname='王涛')
                GROUP BY SC.Sno
            )
         )
ORDER BY Sno DESC;
```

结果：

信息 | 摘要 | 结果 1 | 结果 2 | 解释 1 | 解释 2

sno	sname	avggrade
3031033	蔡明明	91.0000000000000000
3031011	王倩	88.5000000000000000
1042717	华玉家	86.0000000000000000
1042699	郑无山	88.5000000000000000
1042484	韦女让	98.5000000000000000
1042424	尤物子	91.1428571428571429
1042409	岑接立	87.7500000000000000
1042296	钱报笑	90.0000000000000000
1042270	柳长处	86.8750000000000000
1042256	姚民者	87.5000000000000000
1042232	毕期海	86.1000000000000000
1042148	方美向	94.5000000000000000
1041991	华新美	86.7000000000000000
1041957	孟应	94.0000000000000000
1041888	秦经情	92.5833333333333333
1041851	韦间受	85.6250000000000000
1041750	柏电工	86.1666666666666667
1041662	华道乐	86.8333333333333333
1041319	彭手或	87.9166666666666667
1041201	章见真	89.8333333333333333
1041190	乐实夫	95.1666666666666667

分析：

性能得到大幅度提高

信息 | 摘要 | 结果 1 | 结果 2 | 解释 1 | 解释 2

```
QUERY PLAN
Limit  (cost=15633.80..15633.80 rows=1 width=45)
  ->  Sort  (cost=15633.80..17514.86 rows=752426 width=45)
        Sort Key: (avg(sc249.grade)) DESC
        ->  Hash Join  (cost=2658.50..11871.67 rows=752426 width=45)
              Hash Cond: (sc249.sno = public.s249.sno)
              ->  HashAggregate  (cost=1279.06..1654.37 rows=30025 width=50)
                    Group By Key: sc249.sno, public.s249.sname
                    ->  Hash Join  (cost=164.77..1053.87 rows=30025 width=18)
                          Hash Cond: (sc249.sno = public.s249.sno)
                          ->  Seq Scan on sc249  (cost=0.00..476.26 rows=30025 width=9)
                                Filter: (grade IS NOT NULL)
                          ->  Hash  (cost=102.12..102.12 rows=5012 width=13)
                                ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=13)
              ->  Hash  (cost=1316.79..1316.79 rows=5012 width=4)
                    ->  HashAggregate  (cost=1204.02..1266.67 rows=5012 width=17)
                          Group By Key: public.s249.sno
                          Filter: (count(sc.cno) >= 3)
                          ->  Hash Join  (cost=164.77..1053.89 rows=30026 width=9)
                                Hash Cond: (sc.sno = public.s249.sno)
                                ->  Seq Scan on sc249 sc  (cost=0.00..476.26 rows=30026 width=9)
                                ->  Hash  (cost=102.12..102.12 rows=5012 width=4)
                                      ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=4)
```

信息 | 摘要 | 结果 1 | 结果 2 | 解释 1 | 解释 2

```
QUERY PLAN
Index Scan using s249_pkey on s249  (cost=2394.48..2402.75 rows=1 width=13)
  Index Cond: (sno = $0)
  InitPlan 1 (returns $0)
    ->  Limit  (cost=2394.48..2394.48 rows=1 width=41)
          ->  Sort  (cost=2394.48..2406.97 rows=4996 width=41)
                Sort Key: (avg(sc249.grade)) DESC
                ->  HashAggregate  (cost=2307.05..2369.50 rows=4996 width=41)
                      Group By Key: sc249.sno
                      ->  Hash Join  (cost=1379.44..2231.98 rows=15013 width=9)
                            Hash Cond: (sc249.sno = public.s249.sno)
                            ->  Seq Scan on sc249  (cost=0.00..476.26 rows=30026 width=9)
                            ->  Hash  (cost=1316.79..1316.79 rows=5012 width=4)
                                  ->  HashAggregate  (cost=1204.02..1266.67 rows=5012 width=17)
                                        Group By Key: public.s249.sno
                                        Filter: (count(sc.cno) >= 3)
                                        ->  Hash Join  (cost=164.77..1053.89 rows=30026 width=9)
                                              Hash Cond: (sc.sno = public.s249.sno)
                                              ->  Seq Scan on sc249 sc  (cost=0.00..476.26 rows=30026 width=9)
                                              ->  Hash  (cost=102.12..102.12 rows=5012 width=4)
                                                    ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=4)
```

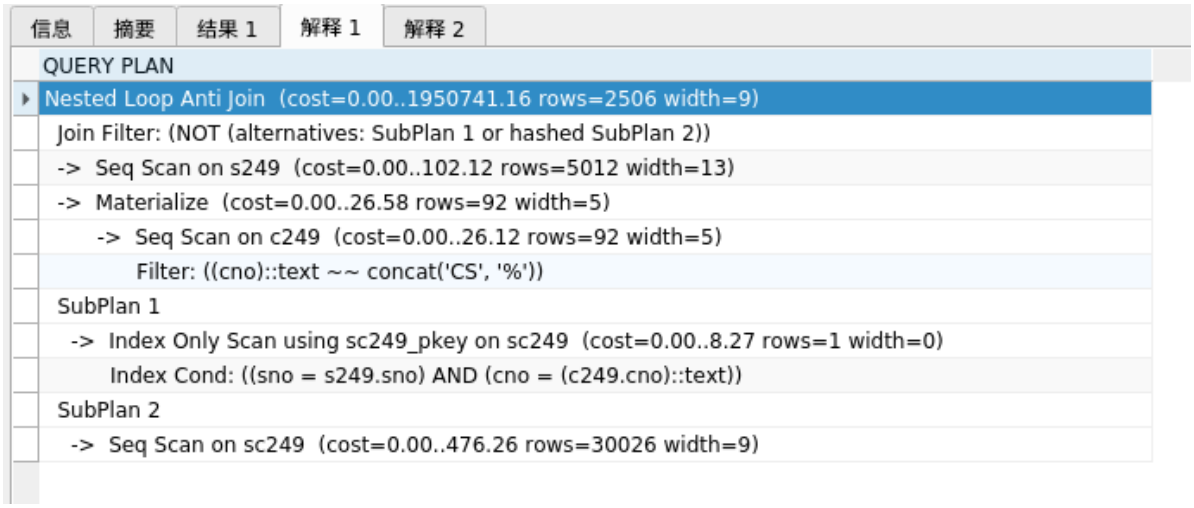
```sql
-- 7
SELECT S249.Sname
FROM S249
WHERE NOT EXISTS
    (SELECT *
    FROM (SELECT Cno FROM C249 WHERE Cno LIKE CONCAT('CS', '%')) AS Cor
    WHERE NOT EXISTS
        (SELECT *
        FROM SC249
        WHERE S249.Sno= SC249.Sno
```

```
        AND SC249.Cno= Cor.Cno));
-- -------------------------------
SELECT S249.Sname
FROM S249
WHERE NOT EXISTS
    (SELECT *
    FROM C249 COR
    WHERE NOT EXISTS
        (SELECT *
        FROM SC249
        WHERE S249.Sno= SC249.Sno
        AND SC249.Cno= Cor.Cno  AND  Cno LIKE CONCAT('CS', '%')));
```

分析：

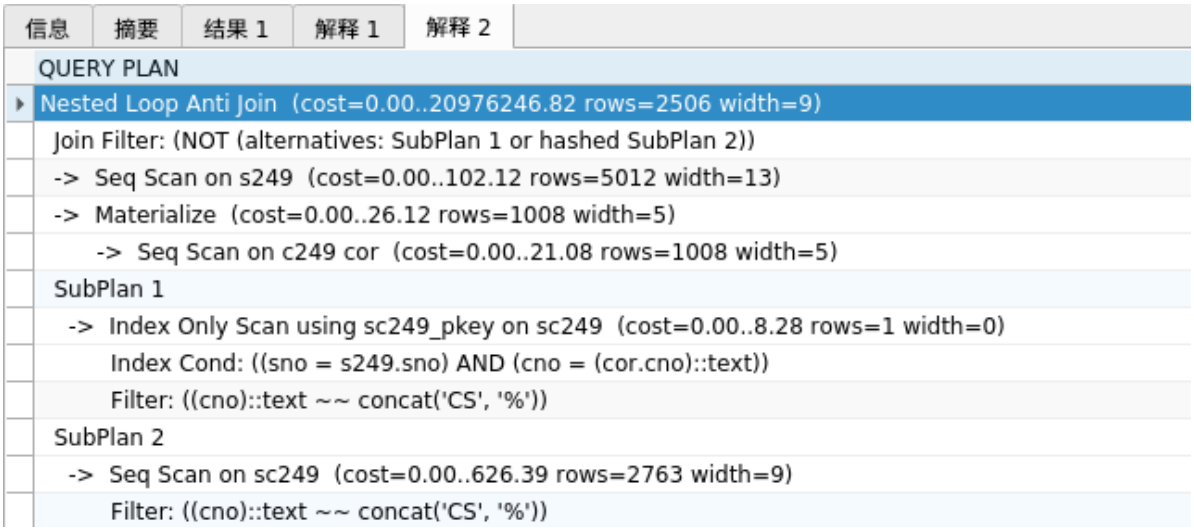
此处将`Cno LIKE CONCAT('CS', '%')`放入内循环，结果是负优化。

从`Expalin`中可以看出，负优化的主要原因主要在于两个SubPlan中均加入了`Cno LIKE CONCAT('CS', '%')`的`filter`导致性能下降。



信息 | 摘要 | 结果 1 | 解释 1 | 解释 2

```
QUERY PLAN
Nested Loop Anti Join  (cost=0.00..1950741.16 rows=2506 width=9)
  Join Filter: (NOT (alternatives: SubPlan 1 or hashed SubPlan 2))
  ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=13)
  ->  Materialize  (cost=0.00..26.58 rows=92 width=5)
        ->  Seq Scan on c249  (cost=0.00..26.12 rows=92 width=5)
              Filter: ((cno)::text ~~ concat('CS', '%'))
  SubPlan 1
    ->  Index Only Scan using sc249_pkey on sc249  (cost=0.00..8.27 rows=1 width=0)
          Index Cond: ((sno = s249.sno) AND (cno = (c249.cno)::text))
  SubPlan 2
    ->  Seq Scan on sc249  (cost=0.00..476.26 rows=30026 width=9)
```



信息 | 摘要 | 结果 1 | 解释 1 | 解释 2

```
QUERY PLAN
Nested Loop Anti Join  (cost=0.00..20976246.82 rows=2506 width=9)
  Join Filter: (NOT (alternatives: SubPlan 1 or hashed SubPlan 2))
  ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=13)
  ->  Materialize  (cost=0.00..26.12 rows=1008 width=5)
        ->  Seq Scan on c249 cor  (cost=0.00..21.08 rows=1008 width=5)
  SubPlan 1
    ->  Index Only Scan using sc249_pkey on sc249  (cost=0.00..8.28 rows=1 width=0)
          Index Cond: ((sno = s249.sno) AND (cno = (cor.cno)::text))
          Filter: ((cno)::text ~~ concat('CS', '%'))
  SubPlan 2
    ->  Seq Scan on sc249  (cost=0.00..626.39 rows=2763 width=9)
          Filter: ((cno)::text ~~ concat('CS', '%'))
```

```
-- 8
-- 优化前
SELECT ST.Sno, Sname
FROM
(     (SELECT S249.Sno Sno
    FROM S249 JOIN SC249 SC
    ON S249.Sno=SC.Sno
    GROUP BY S249.Sno HAVING COUNT(SC.Cno)>=3) AS ST
```

```sql
JOIN
    (SELECT SC.Sno Sno, S249.Sname Sname, AVG(SC.Grade) AvgGrade
       FROM S249 JOIN (SELECT * FROM SC249 WHERE Grade IS NOT NULL) AS SC ON
S249.Sno=SC.Sno
       GROUP BY SC.Sno, S249.Sname) AS AV
ON ST.Sno=AV.Sno) ORDER BY AvgGrade DESC LIMIT 1;

-- 优化后
SELECT Sno, Sname FROM S249 WHERE Sno = (
    SELECT Sno FROM
        (SELECT Sno, Grade FROM SC249 WHERE Sno IN
            (SELECT S249.Sno Sno
                FROM S249 JOIN SC249 SC
                ON S249.Sno=SC.Sno
                GROUP BY S249.Sno HAVING COUNT(SC.Cno)>=3))
GROUP BY Sno ORDER BY  AVG(Grade) DESC LIMIT 1)
```

结果：

信息 | 摘要 | 结果 1 | 解释 1

sno	sname	avggrade
1038099	彭英少	96.1666666666666667

分析：

性能得到大幅度提高

信息 | 摘要 | 结果 1 | 结果 2 | 解释 1 | 解释 2

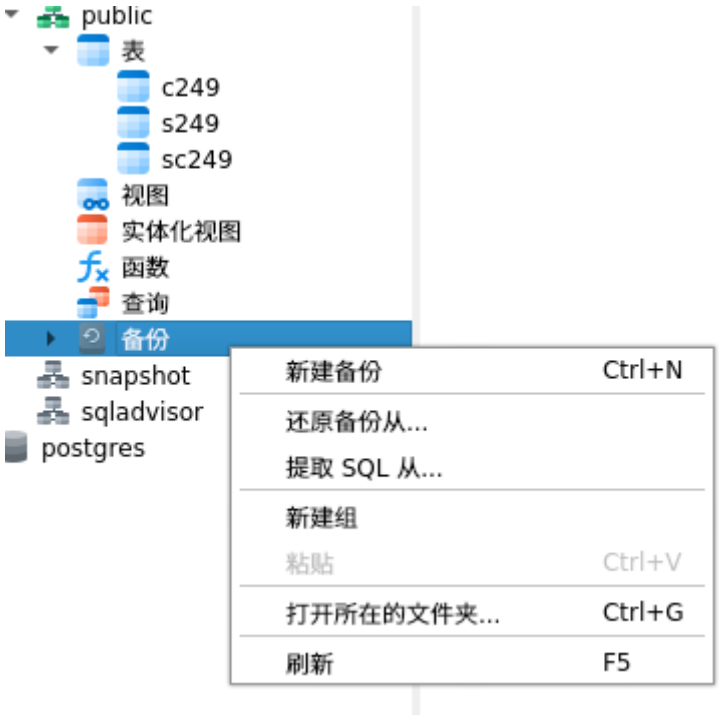
```
QUERY PLAN
Limit  (cost=15633.80..15633.80 rows=1 width=45)
  ->  Sort  (cost=15633.80..17514.86 rows=752426 width=45)
      Sort Key: (avg(sc249.grade)) DESC
      ->  Hash Join  (cost=2658.50..11871.67 rows=752426 width=45)
          Hash Cond: (sc249.sno = public.s249.sno)
          ->  HashAggregate  (cost=1279.06..1654.37 rows=30025 width=50)
              Group By Key: sc249.sno, public.s249.sname
              ->  Hash Join  (cost=164.77..1053.87 rows=30025 width=18)
                  Hash Cond: (sc249.sno = public.s249.sno)
                  ->  Seq Scan on sc249  (cost=0.00..476.26 rows=30025 width=9)
                      Filter: (grade IS NOT NULL)
                  ->  Hash  (cost=102.12..102.12 rows=5012 width=13)
                      ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=13)
          ->  Hash  (cost=1316.79..1316.79 rows=5012 width=4)
              ->  HashAggregate  (cost=1204.02..1266.67 rows=5012 width=17)
                  Group By Key: public.s249.sno
                  Filter: (count(sc.cno) >= 3)
                  ->  Hash Join  (cost=164.77..1053.89 rows=30026 width=9)
                      Hash Cond: (sc.sno = public.s249.sno)
                      ->  Seq Scan on sc249 sc  (cost=0.00..476.26 rows=30026 width=9)
                      ->  Hash  (cost=102.12..102.12 rows=5012 width=4)
                          ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=4)
```

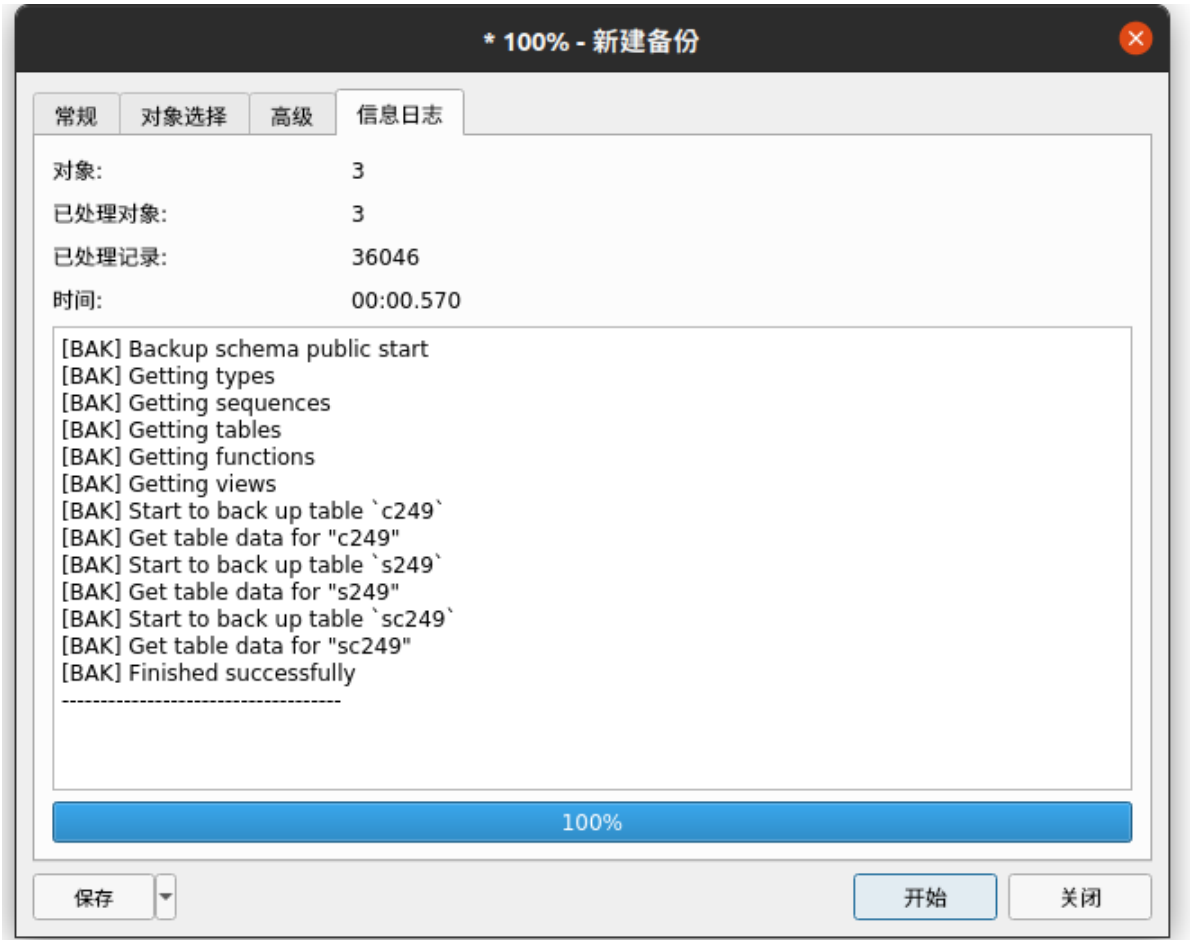
信息 | 摘要 | 结果 1 | 结果 2 | 解释 1 | 解释 2

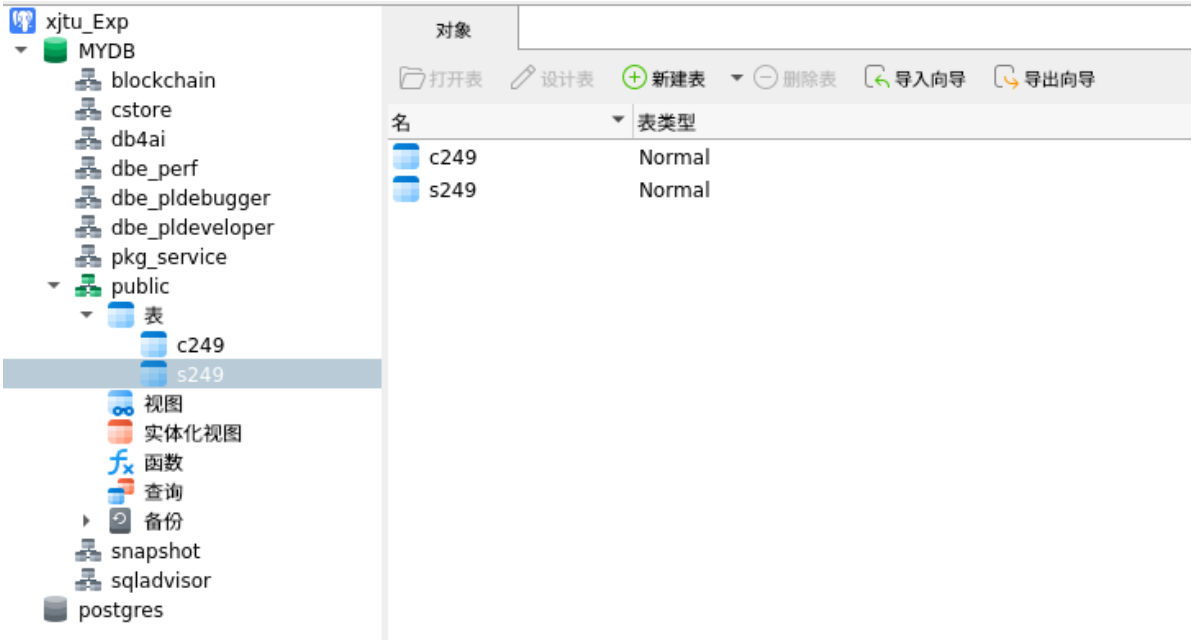
```
QUERY PLAN
Index Scan using s249_pkey on s249  (cost=2394.48..2402.75 rows=1 width=13)
  Index Cond: (sno = $0)
  InitPlan 1 (returns $0)
    ->  Limit  (cost=2394.48..2394.48 rows=1 width=41)
          ->  Sort  (cost=2394.48..2406.97 rows=4996 width=41)
                Sort Key: (avg(sc249.grade)) DESC
                ->  HashAggregate  (cost=2307.05..2369.50 rows=4996 width=41)
                      Group By Key: sc249.sno
                      ->  Hash Join  (cost=1379.44..2231.98 rows=15013 width=9)
                            Hash Cond: (sc249.sno = public.s249.sno)
                            ->  Seq Scan on sc249  (cost=0.00..476.26 rows=30026 width=9)
                            ->  Hash  (cost=1316.79..1316.79 rows=5012 width=4)
                                  ->  HashAggregate  (cost=1204.02..1266.67 rows=5012 width=17)
                                        Group By Key: public.s249.sno
                                        Filter: (count(sc.cno) >= 3)
                                        ->  Hash Join  (cost=164.77..1053.89 rows=30026 width=9)
                                              Hash Cond: (sc.sno = public.s249.sno)
                                              ->  Seq Scan on sc249 sc  (cost=0.00..476.26 rows=30026 width=9)
                                              ->  Hash  (cost=102.12..102.12 rows=5012 width=4)
                                                    ->  Seq Scan on s249  (cost=0.00..102.12 rows=5012 width=4)
```

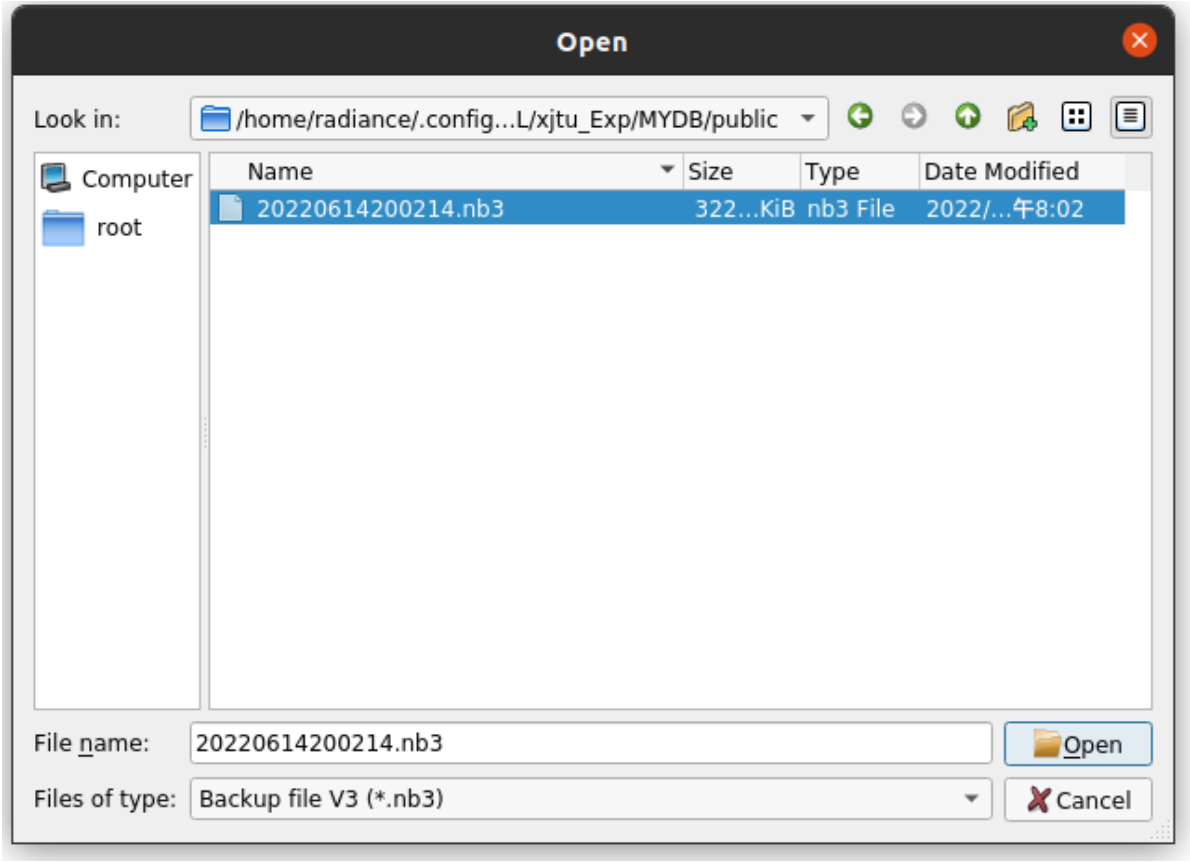
数据库备份

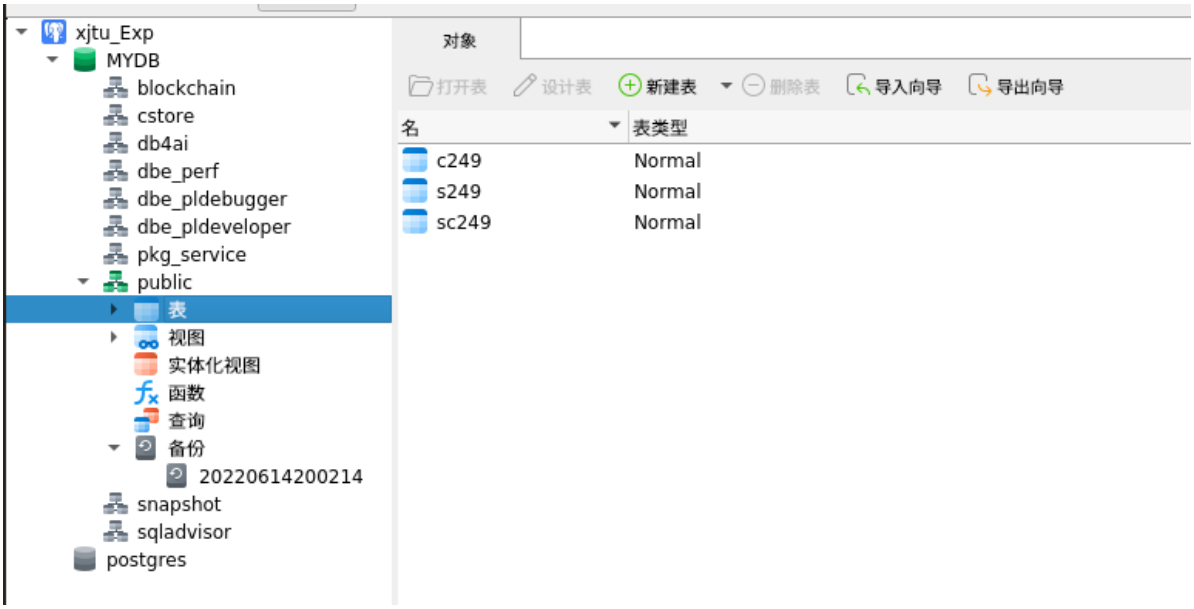
Navicat中备份及恢复




* 100% - 新建备份
常规
对象选择
高级
信息日志
对象: 3
已处理对象: 3
已处理记录: 36046
时间: 00:00.570
[BAK] Backup schema public start
[BAK] Getting types
[BAK] Getting sequences
[BAK] Getting tables
[BAK] Getting functions
[BAK] Getting views
[BAK] Start to back up table `c249`
[BAK] Get table data for "c249"
[BAK] Start to back up table `s249`
[BAK] Get table data for "s249"
[BAK] Start to back up table `sc249`
[BAK] Get table data for "sc249"
[BAK] Finished successfully
100%
保存
开始
关闭


xjtu_Exp
MYDB
blockchain
cstore
db4ai
dbe_perf
dbe_pldebugger
dbe_pldeveloper
pkg_service
public
表
c249
s249
视图
实体化视图
函数
查询
备份
snapshot
sqladvisor
postgres
对象
打开表
设计表
新建表
删除表
导入向导
导出向导
名
表类型
c249 Normal
s249 Normal


Open
Look in:
/home/radiance/.config...L/xjtu_Exp/MYDB/public
Computer
root
Name
Size
Type
Date Modified
20220614200214.nb3
322...KiB
nb3 File
2022/...午8:02
File name:
20220614200214.nb3
Open
Files of type:
Backup file V3 (*.nb3)
Cancel


xjtu_Exp
MYDB
blockchain
cstore
db4ai
dbe_perf
dbe_pldebugger
dbe_pldeveloper
pkg_service
public
表
视图
实体化视图
函数
查询
备份
20220614200214
snapshot
sqladvisor
postgres
对象
打开表
设计表
新建表
删除表
导入向导
导出向导
名
表类型
c249
Normal
s249
Normal
sc249
Normal