**GaussDB(for MySQL)/openGuass**

数据查询与修改

实验指导书

目录

[前 言 3](#_Toc59569720)

[实验环境说明 3](#_Toc59569721)

[1． 实验介绍 5](#_Toc59569722)

[1.1 实验目的 5](#_Toc59569723)

[1.2 实验内容 5](#_Toc59569724)

[2． 实验要求 5](#_Toc59569725)

[2.1 实验步骤 5](#_Toc59569726)

[2.2 实验要求 9](#_Toc59569727)

[3． 实验示例 10](#_Toc59569728)

[String操作 12](#_Toc59569729)

[集合操作 13](#_Toc59569730)

[多表查询 15](#_Toc59569731)

[聚集函数 18](#_Toc59569732)

[嵌套查询【参考教科书3.8节】 22](#_Toc59569733)

[with临时视图查询【参考教科书3.8.6节】 33](#_Toc59569734)

[键/函数依赖分析【参考第8章作业题8.9】 35](#_Toc59569735)

[关系表的插入/删除/更新 37](#_Toc59569736)

前 言

实验环境说明

本实验环境可选以下三种：

1. opneGuass数据库

利用指导教师提供的安装镜像或从openGaussDB社区下载，安装在学生实验用笔记本电脑或台式机的opneGuass数据库。

1. 华为云数据库GaussDB(for MySQL)

以IAM或IP方式登录华为云数据库。

本次实验华为方面提供四个华为主账号，每个主账号可以支持：

i) 最多50个IAM子账号; ii) 100个IP登录方式子账号。

主账号下的1个子账号支持一个数据库实例，供一组学生实验使用。

实验指导教师/助教已经在四个主账号上创建了以IAM方式、IP方式登录的多个子账号，并将账号分配和分发给参加实验的学生。

1. GuassDB100/200数据库

实验指导/助教在计算机学院院实验中心的鲲鹏服务器和应用中心实验室的Intel服务器上安装了GuassDB100/200数据库，创建多组数据库实例并分配给参加实验的学生。学生根据分配的账号，连接、登录数据库实例，完成实验。

为了满足数据库原理课程实验需要，建议每套实验环境采用以下配置：

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 软件版本/硬件配置 |
| 数据库 | openGauss | openGauss1.0.0及以上，openEuler 20.3LTS；  或者：openGauss1.0.0及以上，VirtualBox/VMvare，Windows操作系统；  至少：4核CPU，8G 内存，1T硬盘；  Huawei JDK 1.8.0、psmisc、bzip2 |
| GaussDB(for MySQL) 16 核 | 64 GB | GaussDB(for MySQL)云服务；  客户端至少：4核CPU，8G 内存，1T硬盘；  Huawei JDK 1.8.0、psmisc、bzip2 |
| GaussDB 100/200 | GaussDB 100/200，Redhat 7.2；  服务器端：三个服务器结点，每个结点至少4核CPU，32G 内存，2T硬盘  客户端至少：4核CPU，8G 内存，1T硬盘；  Huawei JDK 1.8.0、psmisc、bzip2 |

# 实验介绍

## 实验目的

对前两个实验建立的LTE数据库关系表和视图进行各种类型的查询操作和修改操作，加深对SQL语言中DML的了解，掌握相关查询语句和数据修改语句的使用方法。

## 实验内容

1. 单表简单查询，包括复合选择条件、结果排序、结果去重、结果重命名查询；
2. 多表查询，包括等值连接、自然连接、元组变量查询；
3. 统计查询，包括带有分组、聚集函数的查询；
4. 嵌套查询，包括带有**in/some/all、 exists、unique**的嵌套查询，from中子查询；
5. with临时视图查询；
6. 键/函数依赖分析；
7. 表的插入、删除、更新；

# 实验要求

## 实验步骤

依次完成以下各个查询实验。

**单表查询**

**查询1**：从小区/基站信息表tbCell表中，找出“sanxia”市满足下列条件的所有小区cell：

（1）所属基站的经纬度范围分别位于[？,？]、[？,？？]，并且

（2）PCI值在？至？之间，并且

（3）设备厂家VENDOR不为空

，列出这些小区的小区标识（Sector\_ID）、小区名、所属基站的基站ID和基站名、基站经纬度、小区PCI、小区天线的方位角(azimuth)和高度(height)；

要求：对查询结果，按照经度范围从大到小、纬度范围从大到小、频点(RARFCN)从高到低排序，并且将PCI重新命名为Physical Cell Identity。

说明：？代表由学生自己选择输入条件

**查询2**：从小区/基站信息表tbCell表中，找出“sanxia”市满足下列条件的所有基站ENodeB：

（1）所属基站的经纬度范围分别位于[？-？]、[？,？？]，

（2）属于该基站的小区中，至少有一个小区的PCI值在？至？之间

，列出这些基站的基站ID和基站名、基站经纬度、基站类型(Style)、设备生产厂家(Vendor)；要求：对查询结果，按照基站位置从北到南、从东到西排序，并且对查询结果使用distinct去重。

比较对查询结果去重和不去重，在查询时间和查询结果上的差异。

**String操作**

**查询3**：从小区/基站信息表tbCell表中，找出满足下列条件的小区：

（1）小区名开头部分包含“A池”或“高铁”，或者基站名中包含“医院”或“实验高中”，并且

（2） 不是所属基站的第1小区，即小区名结尾部分不是“-1”

**查询4**：从小区/基站信息表tbCell表中，找出满足下列条件的小区：

（1）小区标识由5个字符组成，并且

（2）小区所属基站的名字/标识至少包括8个字符，即名字字符串的长度不小于8。

**集合操作**

**查询5**：使用集合并操作union、union all，从小区KPI指标表tbCellKPI查询满足下列条件的小区

(1)小区RRC建立成功率qf (%)大于95%，或者

(2)E-RAB建立成功率2 (%)大于99%

对比union all、union操作在查询结果、执行时间上的差异。

**查询6**：结合教材3.4.1节元组变量样例，使用集合操作except、except all，从小区/基站信息表tbCell表中，查询位于最北端（具有最大纬度）的基站。

对比使用except、except all、聚集函数max，完成此查询在执行时间、查询结果上的异同。

**多表查询**

**查询7**：选取两张数据量比较大的表T1和T2，如tbMROData、tbCellTraffic、tbC2I、tbHandover，执行如下无连接条件的笛卡尔积操作，观察数据库系统的反应和查询结果：

Select \* from T1, T2

**查询8**：使用多表连接操作（3.3.3 join/natual join，4.1.1 join），从小区/基站信息表tbCell表、小区一阶邻区关系表tbAdjCell、小区二阶（同频）邻区关系表tbSecAdjCell中，查询有相同的一阶邻小区和二阶邻小区的主小区，列出这些主小区的小区标识、小区名称、小区频点，以及该小区的一阶邻小区和二阶邻小区的小区标识及其频点。

**查询9**：使用多表连接操作，从小区/基站信息表tbCell表、路测ATU C2I干扰矩阵表tbATUC2I、路测ATU切换统计矩阵表tbATUHandover中，查询小区标识ID为“238397-1”的主小区的同站干扰小区的小区和切换目标小区，列出主小区名称和ID、同站干扰小区的ID、切换目标小区的ID。

**查询10**：利用MR测量报告干扰分析表tbC2I表，使用教材3.4.1节元组变量as/rename方式，查询所有比主小区ID为“124673-0”，邻小区ID为“259772-0”的小区间C2I干扰均值高的主小区、邻小区，列出这些主邻小区的名称、ID和C2I干扰值，结果按照C2I干扰均值的降序排列。

**聚集函数**

**查询11**：从小区小时级话务量表tbCellTraffic、从小区/基站信息表tbCell表，查询2020年5月期间，每天忙时时段（包括早9点-11点2个小时、晚19点-21点，共4个小时）的位于经度范围[？,?]、纬度范围[?,?]内的

（1）全部小区的最大月忙时话务量、最小月忙时话务量、平均月忙时话务量;

（2）具有最大月忙时话务量的小区，列出该小区ID、名称、经纬度位置，以及月忙时话务量。

月忙时话务量=月内各天的忙时话务量累加

**查询12**：根据优化小区/保护带小区表tbOptCell和小区一阶邻区关系表tbAdjCell，查询一阶邻区数大于10的优化小区，给出这些优化小区的标识、名称，以及邻区数量，并将查询结果按照邻区数目降序排列。

**查询13**：从小区话务量表tbCellTraffic、小区/基站信息表tbCell表中，查询所有包含38400频点的基站的年平均小时级话务量avgTraffic，给出年平均话务量超出avgTraffic的基站ID名称、基站年平均话务量，结果按照年平均话务量降序排列。

注：1个基站ENodeB有13个38400频点小区。

**嵌套查询【参考教科书3.8节】**

**查询14**：从小区/基站信息表tbCell、优化小区/保护带小区tbOptCell和小区PCI优化调整结果表tbPCIAssignment中，使用set membership运算符in，查询小区类型为“优化区”的小区、PCI调整前后没有发生变化的小区，列出这些优化小区的ID和名称、调整前后的PCI。

对比使用多表连接、非嵌套的查询在执行时间、查询结果上的异同。

**查询15-1**：从小区/基站信息表tbCell，使用Set Comparison 运算符some，查询满足下列条件的小区：该小区的天线高度height高于位于经度在[?,?]、纬度在[?,?]区域内的部分（至少一个）小区的天线高度，列出这些小区的名称、标识和天线高度。

**查询15-2**：从路测ATU数据表tbATUdata，使用Set Comparison 运算符some，查询满足下列条件的小区：在路测数据中作为主小区/服务小区CELLID，其参考信号接收功率RSRP，大于部分（基站标识ENodeBID=253903的小区作为主小区/服务小区时的参考信号接收功率RSRP。列出这些小区的ID、名称、在测量报告中作为主服务小区的RSRP。

**查询16-1**：从小区小时级话务量表tbCellTraffic中，使用Set Comparison 运算符>=all，查询全年小时级话务量总和满足下列条件的小区：该小区的全年话务量总和大于等于其它小区的全年话务量总和，即该小区的全年话务量总和最高。

**查询16-2**：切换统计表tbHandOver，使用Set Comparison 运算符all，查询作为源小区SCELL与周边目标/邻小区NCELL发生切换次数(HOATT)最多的小区。列出这些源小区的ID、目标/邻小区ID、发生的切换次数。

**查询17-1**：从切换统计表tbHandOver表，使用Test for Empty Relations运算符“not exists”，查询作为源小区SCELL，其切换邻小区NCELL包含了{15290-128, 259595-1, 124711-0, 47444-1}中的全部四个小区。

**查询17-2**：从一阶邻区表tbAdjCell、二阶邻区表tbSecAdjCell中，使用Test for Empty Relations 运算符“not exists”，查询满足下列条件的源小区S\_SECTOR\_ID：该小区的一阶邻小区包含其全部二阶邻小区，或者该小区的二阶邻小区包含其全部一阶邻小区。

**查询18**：从小区/基站信息表tbCell、一阶邻区表tbAdjCell中，使用Test for Absence of Duplicate Tuples运算符not unique，查询满足下列条件的源小区：ENodeBID=15114的基站下有多个小区，该源小区至少是基站15114下2个小区的邻小区。

**查询19**：从小区KPI性能表tbCellKPI中，使用Subqueries in the From Clause方法，查询满足下列条件的小区：小区在2020/07/17-2020/0719这三天的平均RRC建立成功率大于0.992，给出这些小区的ID和其三天平均连接成功率。

**with临时视图查询【参考教科书3.8.6节】**

**查询20**：用with临时视图方式，实现查询19中查询要求。

**查询21**：从小区/基站信息表tbCell、一阶邻区表tbAdjCell中，用with临时视图方式，查询一阶邻小区最多的主小区，给出这些主小区的ID、邻小区数目。

**键/函数依赖分析【参考第8章作业题8.9】**

**查询22**：在MRO测量报告数据表tbMROData中，检查TimeStamp、ServingSector、InterferingSector是否组成超键。

**查询23**：在PCI优化分配表tbPCIAssignment中，利用SQL语句检查函数依赖ENODEB\_IDPCI是否成立；如果不成立，利用SQL语句找出导致函数依赖不成立的元组。

**关系表的插入/删除/更新**

**查询24**：向小区一阶邻区关系表中插入一条邻区数据；

**查询25**：将小区124673-0的全部二阶邻小区，作为该小区的一阶邻小区，加入到一阶邻区表tbAdjCell中。

**查询26**：在小区切换统计性能表tbHandover中，删除切换次数最少的那些切换数据。

**查询27**：用小区PCI优化调整结果表tbPCIAssignmeng给出的各个小区的优化调整后的PCI值，替换小区/基站消息表tbCell中对应小区的物理小区标识PCI。

**查询28**：针对路测ATU干扰矩阵表tbATUC2I，使用update/case语句做出如下修改：对主小区SECTOR\_ID=238397-1，如果该小区与干扰小区N\_SECTOR\_ID为同站小区(cosite=1)且干扰强度排序rank不小于1，则干扰强度排序减1；如果主小区与干扰小区不同站，干扰强度排序加1。

## 实验要求

用Transact\_SQL语句完成以上操作。

每组独立完成以上内容。

根据实验内容完成实验报告，实验报告包括：实验环境、目的、内容/步骤、总结等内容

# 实验示例

查询1：从小区/基站信息表tbCell表中，找出“sanxia”市满足下列条件的所有小区cell：

（1）所属基站的经纬度范围分别位于[？,？]、[？,？？]，并且

（2）PCI值在？至？之间，并且

（3）设备厂家VENDOR不为空

，列出这些小区的小区标识（Sector\_ID）、小区名、所属基站的基站ID和基站名、基站经纬度、小区PCI、小区天线的方位角(azimuth)和高度(height)；

要求：对查询结果，按照经度范围从大到小、纬度范围从大到小、频点(EARFCN)从高到低排序，并且将PCI重新命名为Physical Cell Identity。

说明：？代表由学生自己选择输入条件

SQL查询语句：

SELECT SECTOR\_ID, SECTOR\_NAME, ENODEB\_NAME, ENODEBID, LATITUDE

, LONGITUDE, pci AS Physical\_Cell\_Identity, AZIMUTH, HEIGHT

FROM tbcell

WHERE CITY = 'sanxia'

AND LATITUDE BETWEEN 33 AND 34

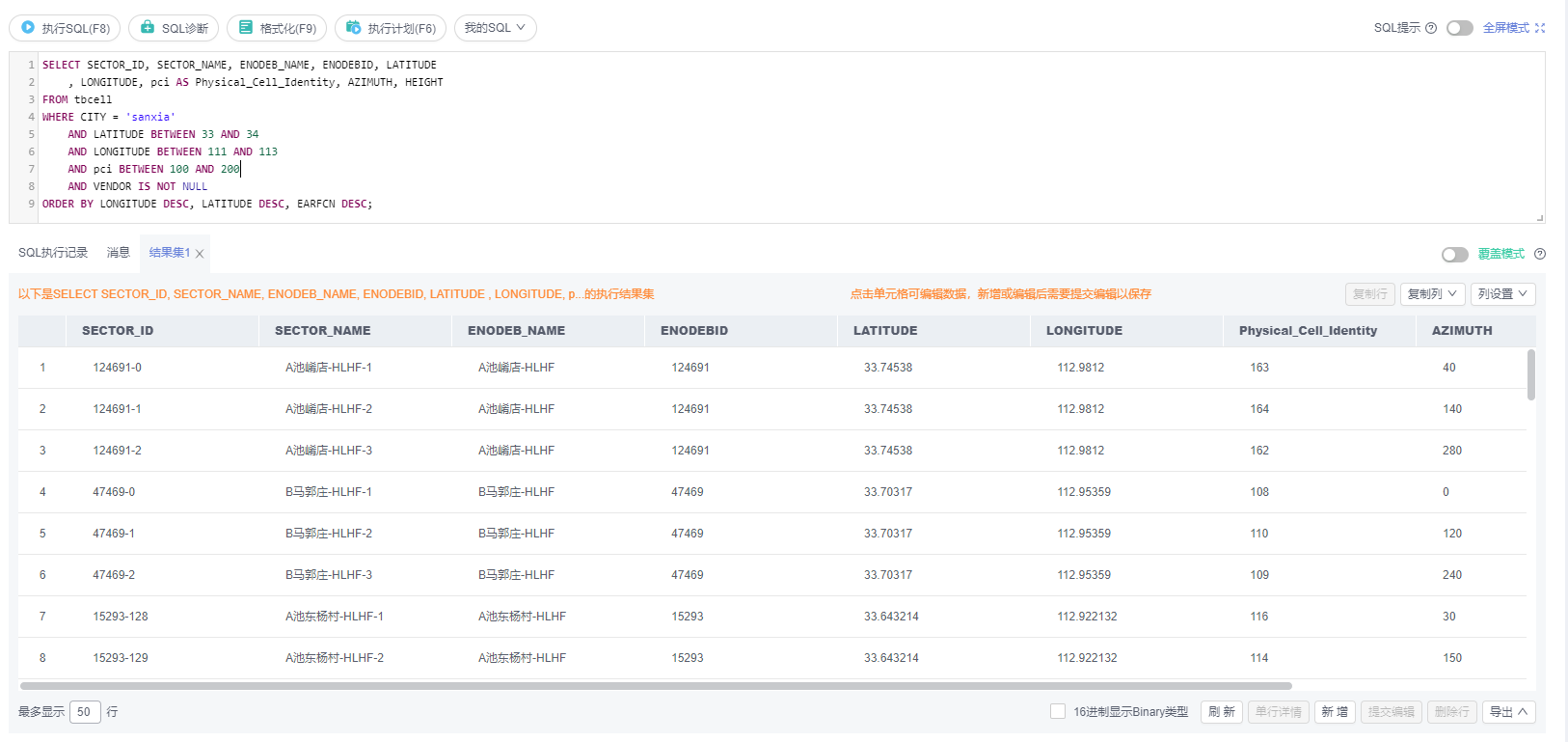
AND LONGITUDE BETWEEN 111 AND 113

AND pci BETWEEN 100 AND 200

AND VENDOR IS NOT NULL

ORDER BY LONGITUDE DESC, LATITUDE DESC, EARFCN DESC

**执行结果如下：**



查询2：从小区/基站信息表tbCell表中，找出“sanxia”市满足下列条件的所有基站ENodeB：

（1）所属基站的经纬度范围分别位于[？-？]、[？,？？]，

（2）属于该基站的小区中，至少有一个小区的PCI值在？至？之间

，列出这些基站的基站ID和基站名、基站经纬度、基站类型(Style)、设备生产厂家(Vendor)；要求：对查询结果，按照基站位置从北到南（从大到小）、从东到西排序（从大到小），并且对查询结果使用distinct去重。比较对查询结果去重和不去重，在查询时间和查询结果上的差异。

SQL查询语句：

SELECT DISTINCT ENODEBID, ENODEB\_NAME, LATITUDE, LONGITUDE, STYLE

, VENDOR, ROW\_NUMBER() OVER (ORDER BY ENODEBID) AS 行号

FROM tbcell t1

WHERE CITY = 'sanxia'

AND EXISTS (

SELECT \*

FROM tbcell t2

WHERE t1.ENODEBID = t2.ENODEBID

AND PCI BETWEEN 100 AND 200

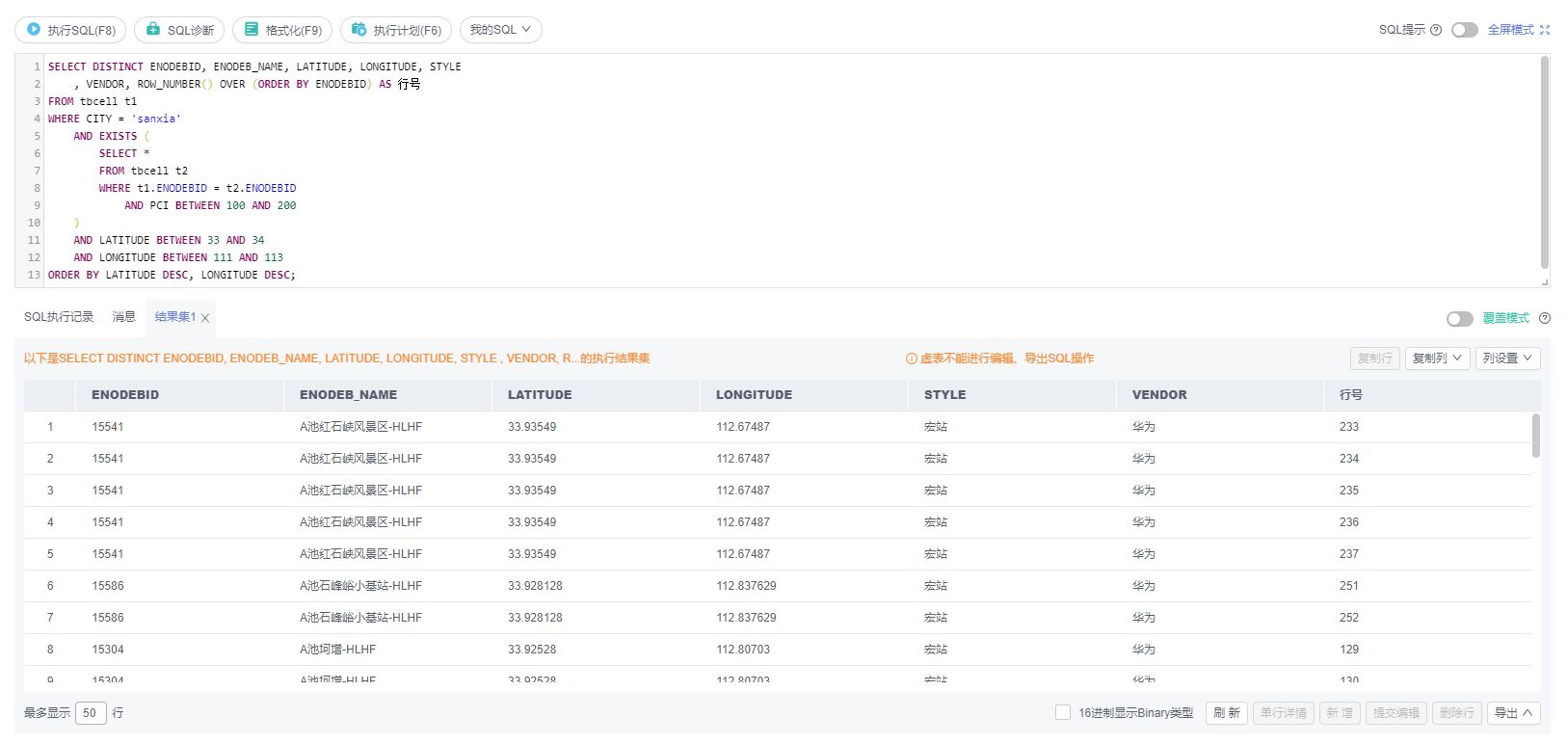
)

AND LATITUDE BETWEEN 33 AND 34

AND LONGITUDE BETWEEN 111 AND 113

ORDER BY LATITUDE DESC, LONGITUDE DESC

**执行结果如下：**



#### **String操作**

查询3：从小区/基站信息表tbCell表中，找出满足下列条件的小区：

（1）小区名开头部分包含“A池”或“高铁”，或者基站名中包含“医院”或“实验高中”，并且

（2） 不是所属基站的第1小区，即小区名结尾部分不是“-1”

SQL查询语句：

SELECT SECTOR\_ID, SECTOR\_NAME

FROM tbcell

WHERE SECTOR\_NAME LIKE 'A池%'

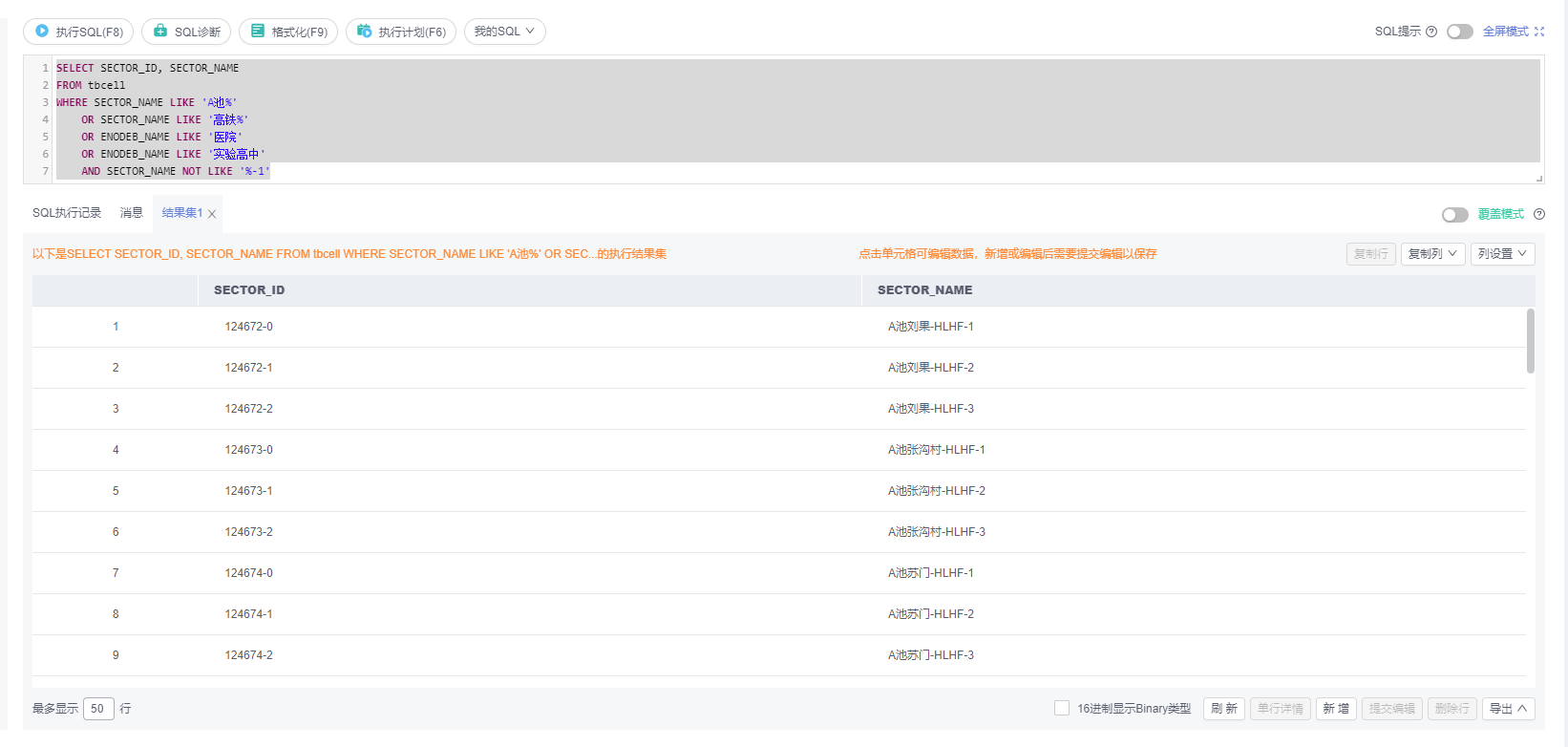
OR SECTOR\_NAME LIKE '高铁%'

OR ENODEB\_NAME LIKE '医院'

OR ENODEB\_NAME LIKE '实验高中'

AND SECTOR\_NAME NOT LIKE '%-1'

**执行结果如下：**



查询4：从小区/基站信息表tbCell表中，找出满足下列条件的小区：

（1）小区标识由5个字符组成，并且

（2）小区所属基站的名字/标识至少包括8个字符，即名字字符串的长度不小于8。

SQL查询语句：

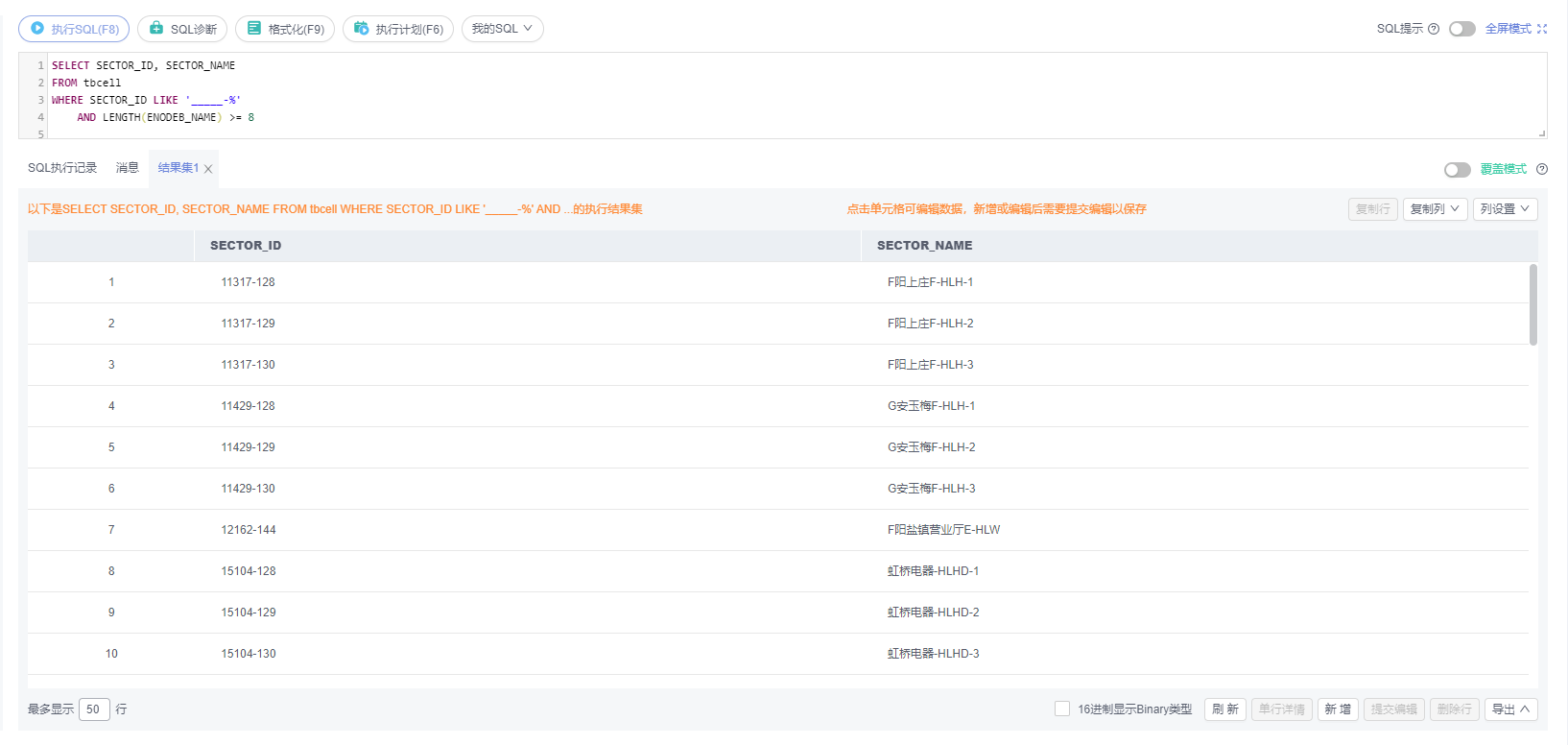
SELECT SECTOR\_ID, SECTOR\_NAME

FROM tbcell

WHERE SECTOR\_ID LIKE '\_\_\_\_\_-%'

AND LENGTH(ENODEB\_NAME) >= 8

**执行结果如下：**



#### **集合操作**

查询5：使用集合并操作union、union all，从小区KPI指标表tbCellKPI查询满足下列条件的小区

(1)小区RRC建立成功率qf (%)大于95%，或者

(2)E-RAB建立成功率2 (%)大于99%。

对比union all、union操作在查询结果、执行时间上的差异。

SQL查询语句：(union all)

SELECT 小区

FROM tbcellkpi

WHERE RRC建立成功率qf > 0.95

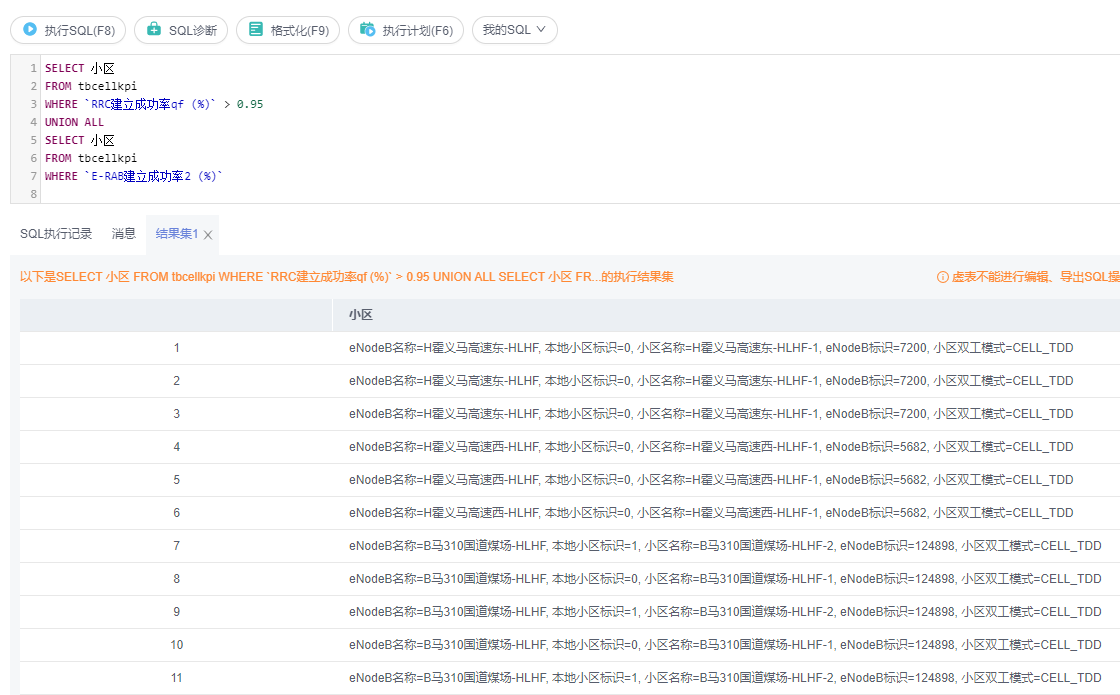
UNION ALL

SELECT 小区

FROM tbcellkpi

WHERE ERAB建立成功率2 > 0.99

**执行结果如下：**



SQL查询语句：(union)

SELECT 小区

FROM tbcellkpi

WHERE RRC建立成功率qf > 0.95

UNION

SELECT 小区

FROM tbcellkpi

WHERE ERAB建立成功率2 > 0.99

**执行结果如下：**

查询6：结合教材3.4.1节元组变量样例，使用集合操作except、except all，从小区/基站信息表tbCell表中，查询位于最北端（具有最大纬度）的基站。对比使用except、except all、聚集函数max，完成此查询在执行时间、查询结果上的异同。

SQL查询语句：

SELECT ENODEBID, ENODEBID

FROM tbcell

WHERE LATITUDE = (

SELECT MAX(LATITUDE)

FROM tbcell

)

**执行结果如下：**



#### **多表查询**

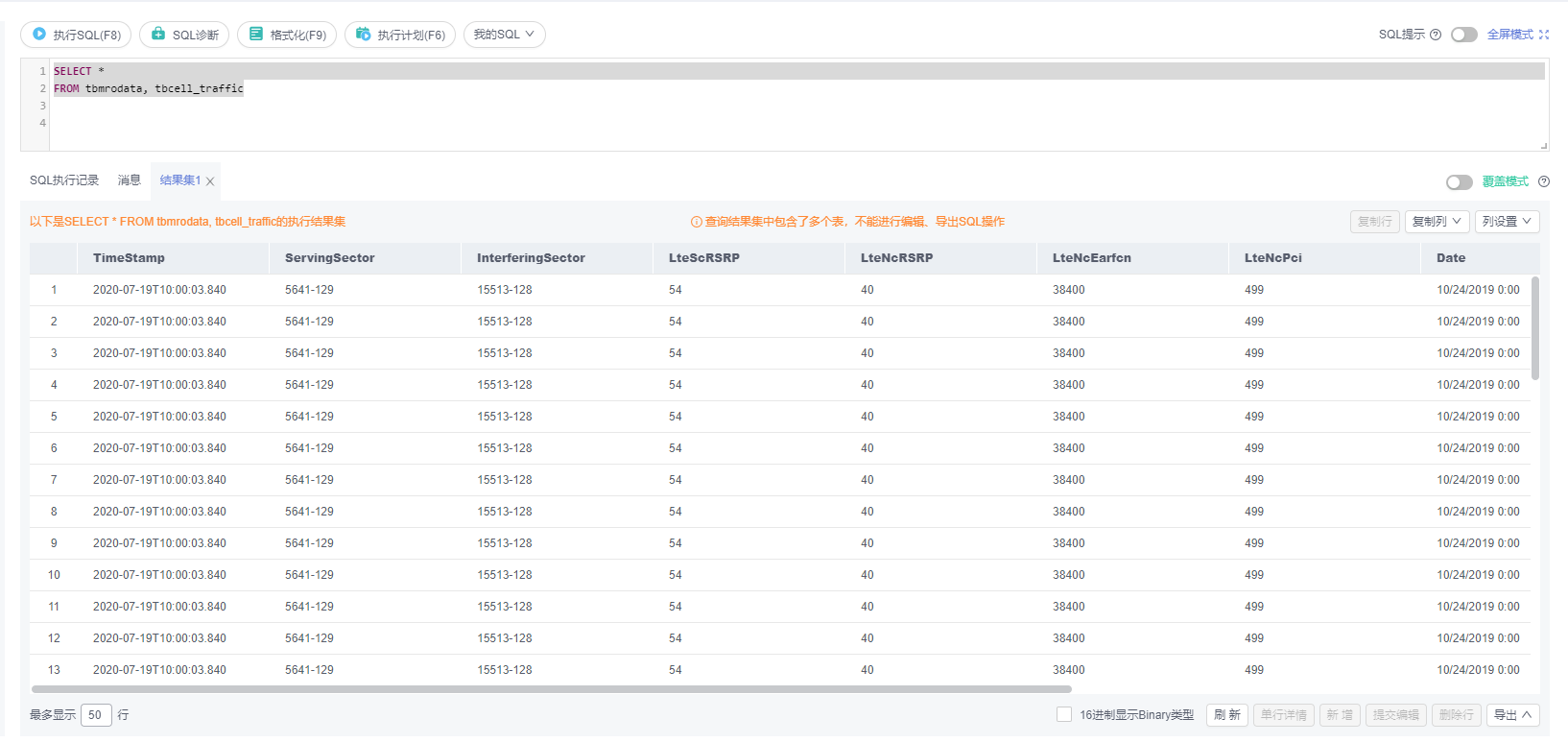
查询7：选取两张数据量比较大的表T1和T2，如tbMROData、tbCellTraffic、tbC2I、tbHandover，执行如下无连接条件的笛卡尔积操作，观察数据库系统的反应和查询结果：

SQL查询语句：

SELECT \*

FROM tbmrodata, tbcelltraffic

**执行结果如下：**



查询8：使用多表连接操作（3.3.3 join/natual join，4.1.1 join），从小区/基站信息表tbCell表、小区一阶邻区关系表tbAdjCell、小区二阶（同频）邻区关系表tbSecAdjCell中，查询有相同的一阶邻小区和二阶邻小区的主小区，列出这些主小区的小区标识、小区名称、小区频点，以及该小区的一阶邻小区和二阶邻小区的小区标识及其频点。

SQL查询语句：

SELECT SECTOR\_ID, SECTOR\_NAME, EARFCN, N\_SECTOR\_ID, N\_EARFCN

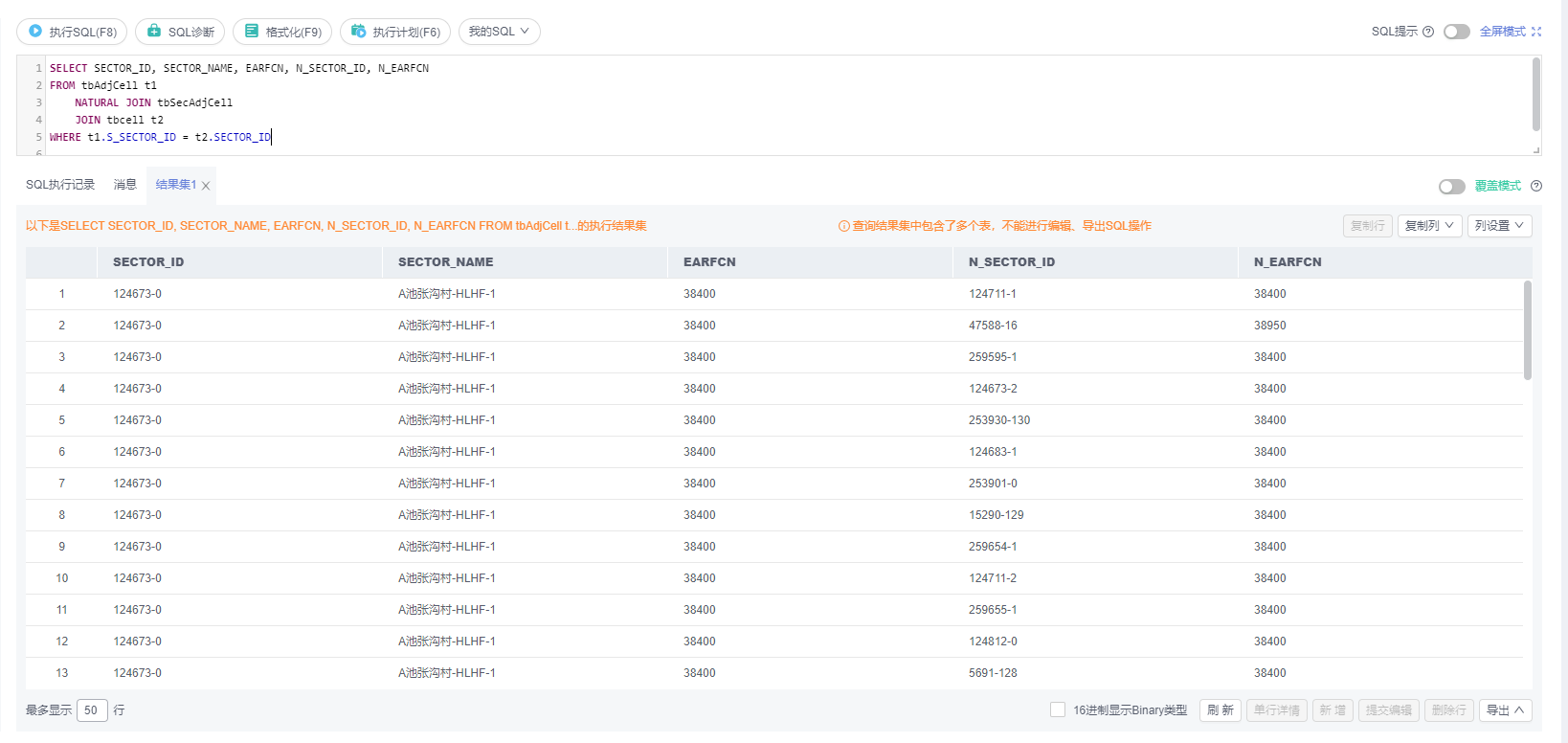
FROM tbAdjCell t1

NATURAL JOIN tbSecAdjCell

JOIN tbcell t2

WHERE t1.S\_SECTOR\_ID = t2.SECTOR\_ID

**执行结果如下：**



查询9：使用多表连接操作，从小区/基站信息表tbCell表、路测ATU C2I干扰矩阵表tbATUC2I、路测ATU切换统计矩阵表tbATUHandover中，查询小区标识ID为“238397-1”的主小区的同站干扰小区的小区和切换目标小区，列出主小区名称和ID、同站干扰小区的ID、切换目标小区的ID。

SQL查询语句：

SELECT SECTOR\_ID, SECTOR\_NAME, NCELL\_ID AS NSECTOR\_ID

FROM tbcell

NATURAL JOIN tbatuc2i

JOIN tbatuhandover

WHERE tbcell.SECTOR\_ID = tbatuhandover.SSECTOR\_ID

AND tbatuc2i.COSITE = 1

AND tbcell.SECTOR\_ID = '238397-1'

**执行结果如下：**



查询10：利用MR测量报告干扰分析表tbC2I表，使用教材3.4.1节元组变量as/rename方式，查询所有比主小区ID为“124673-0”，邻小区ID为“259772-0”的小区间C2I干扰均值高的主小区、邻小区，列出这些主邻小区的名称、ID和C2I干扰值，结果按照C2I干扰均值的降序排列。

SQL查询语句：

SELECT scell, ncell, c2i\_mean

FROM tbc2i

WHERE c2i\_mean > (

SELECT c2i\_mean

FROM tbc2i t

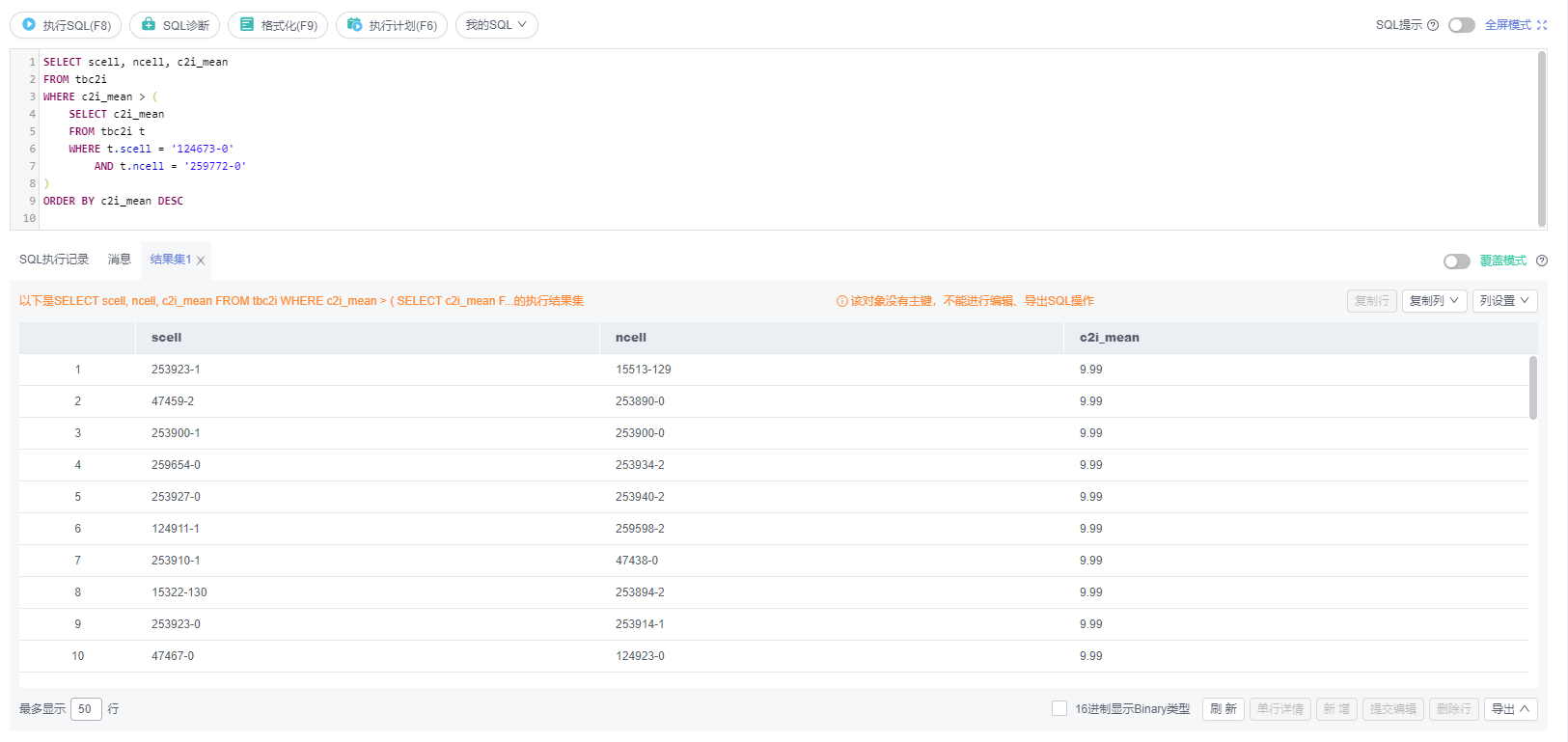
WHERE t.scell = '124673-0'

AND t.ncell = '259772-0'

)

ORDER BY c2i\_mean DESC

**执行结果如下：**



#### **聚集函数**

查询11：从小区小时级话务量表tbCellTraffic、从小区/基站信息表tbCell表，查询2020年5月期间，每天忙时时段（包括早9点-11点2个小时、晚19点-21点，共4个小时）的位于经度范围[？,?]、纬度范围[?,?]内的

（1）全部小区的最大月忙时话务量、最小月忙时话务量、平均月忙时话务量;

（2）具有最大月忙时话务量的小区，列出该小区ID、名称、经纬度位置，以及月忙时话务量。

月忙时话务量=月内各天的忙时话务量累加

(1)SQL查询语句：

WITH Sector\_traffic AS (

SELECT Sector\_ID, SUM(traffic) AS s\_traffic

FROM tbcell\_traffic

WHERE date BETWEEN '2020-05-01' AND '2020-05-31'

AND (hour BETWEEN 9 AND 11

OR hour BETWEEN 19 AND 21)

AND Sector\_ID IN (

SELECT Sector\_ID

FROM tbcell

WHERE LONGITUDE BETWEEN 111 AND 112

AND LATITUDE BETWEEN 33 AND 34

)

GROUP BY Sector\_ID

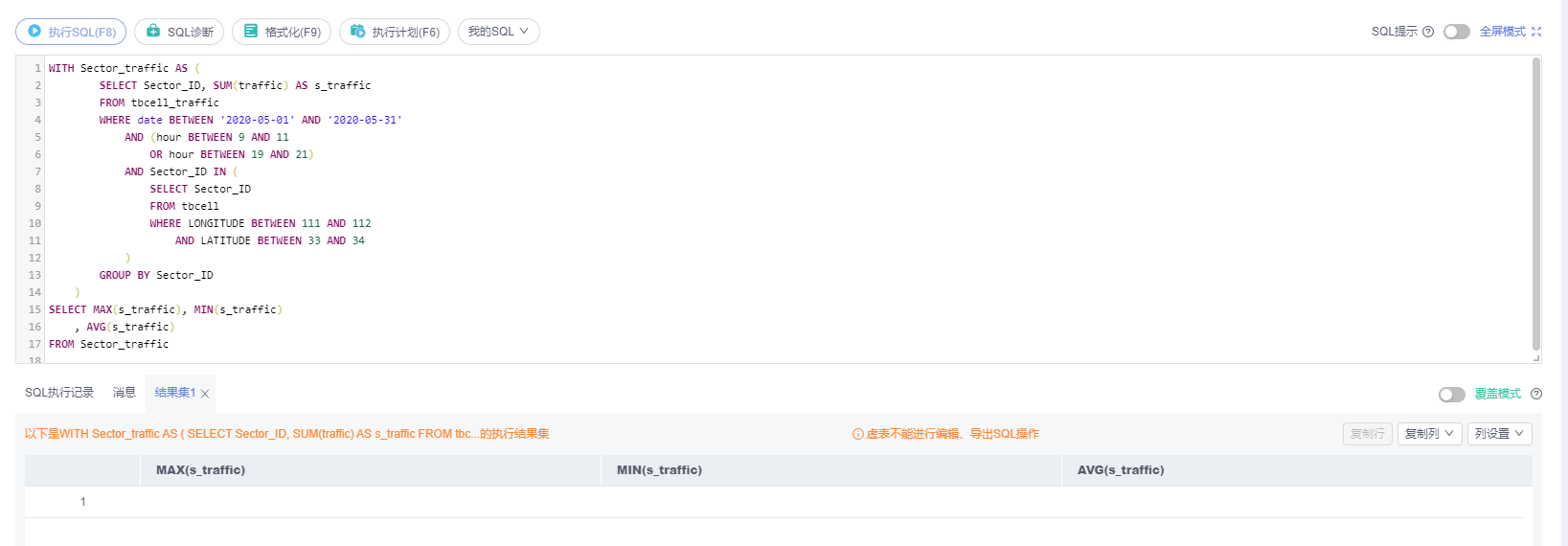
)

SELECT MAX(s\_traffic), MIN(s\_traffic)

, AVG(s\_traffic)

FROM Sector\_traffic

**执行结果如下：**



(2)SQL查询语句：

WITH Sector\_traffic AS (

SELECT Sector\_ID, SUM(traffic) AS s\_traffic

FROM tbcell\_traffic

WHERE date BETWEEN '2020-05-01' AND '2020-05-31'

AND (hour BETWEEN 9 AND 11

OR hour BETWEEN 19 AND 21)

AND Sector\_ID IN (

SELECT Sector\_ID

FROM tbcell

WHERE LONGITUDE BETWEEN 111 AND 112

AND LATITUDE BETWEEN 33 AND 34

)

GROUP BY Sector\_ID

)

SELECT Sector\_ID, SECTOR\_NAME, Longitude, Latitude, s\_traffic

FROM tbcell

NATURAL JOIN Sector\_traffic

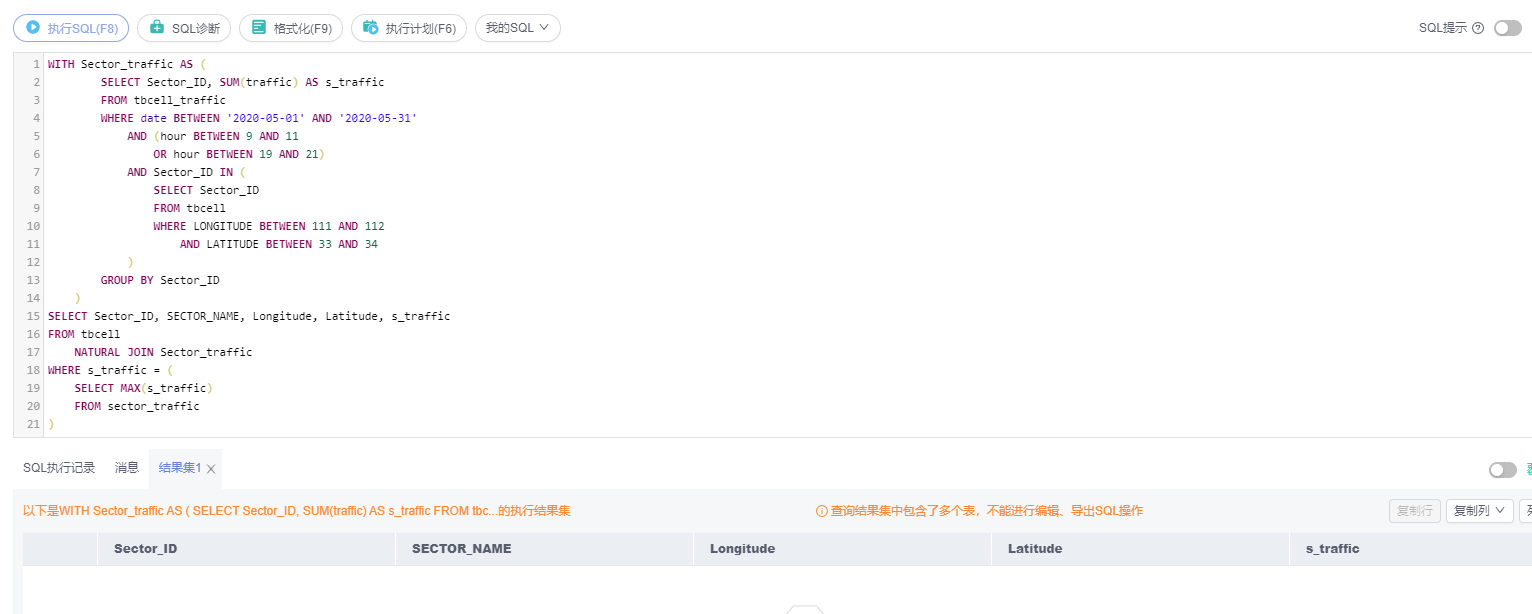
WHERE s\_traffic = (

SELECT MAX(s\_traffic)

FROM sector\_traffic

)

**执行结果如下：**



查询12：根据优化小区/保护带小区表tbOptCell和小区一阶邻区关系表tbAdjCell，查询一阶邻区数大于10的优化小区，给出这些优化小区的标识、名称，以及邻区数量，并将查询结果按照邻区数目降序排列。

SQL查询语句：

SELECT S\_SECTOR\_ID, SECTOR\_NAME, COUNT(N\_sECTOR\_ID) AS c

FROM tbadjcell

JOIN tbcell

WHERE S\_SECTOR\_ID = SECTOR\_ID

AND s\_sector\_id IN (

SELECT sector\_id

FROM tboptcell

WHERE cell\_type = ’优化区’

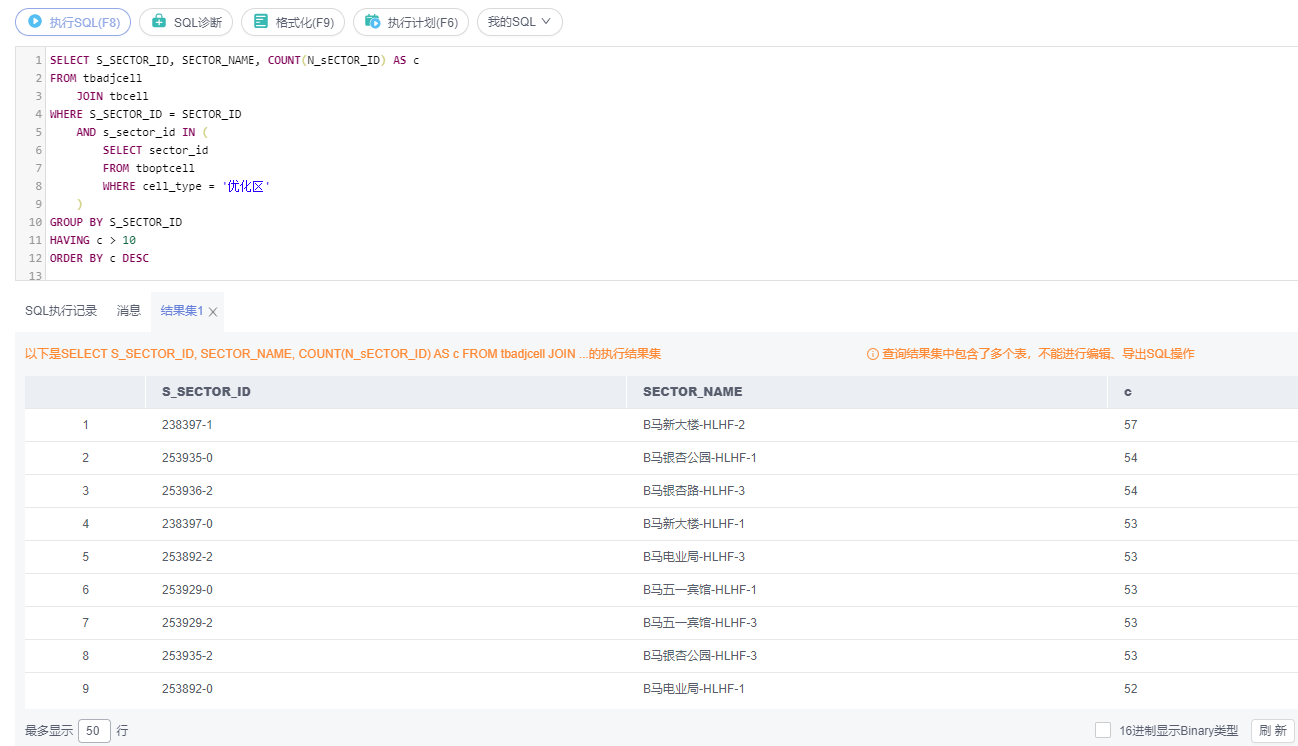
)

GROUP BY S\_SECTOR\_ID

HAVING c > 10

ORDER BY c DESC

**执行结果如下：**



查询13：从小区话务量表tbCellTraffic、小区/基站信息表tbCell表中，查询所有包含38400频点的基站的年平均小时级话务量avgTraffic，给出年平均话务量超出avgTraffic的基站ID名称、基站年平均话务量，结果按照年平均话务量降序排列。

注：1个基站ENodeB有13个38400频点小区。

SQL查询语句：

WITH tbcell1 AS (

SELECT \*

FROM tbcell

WHERE EARFCN = 38400

)

SELECT ENODEBID, AVG(traffic) AS avgTraffic

FROM tbcell\_traffic

NATURAL JOIN tbcell1

GROUP BY ENODEBID

ORDER BY avgTraffic DESC

以下也可：

SELECT ENODEBID, AVG(traffic) AS avgTraffic

FROM tbcell\_traffic

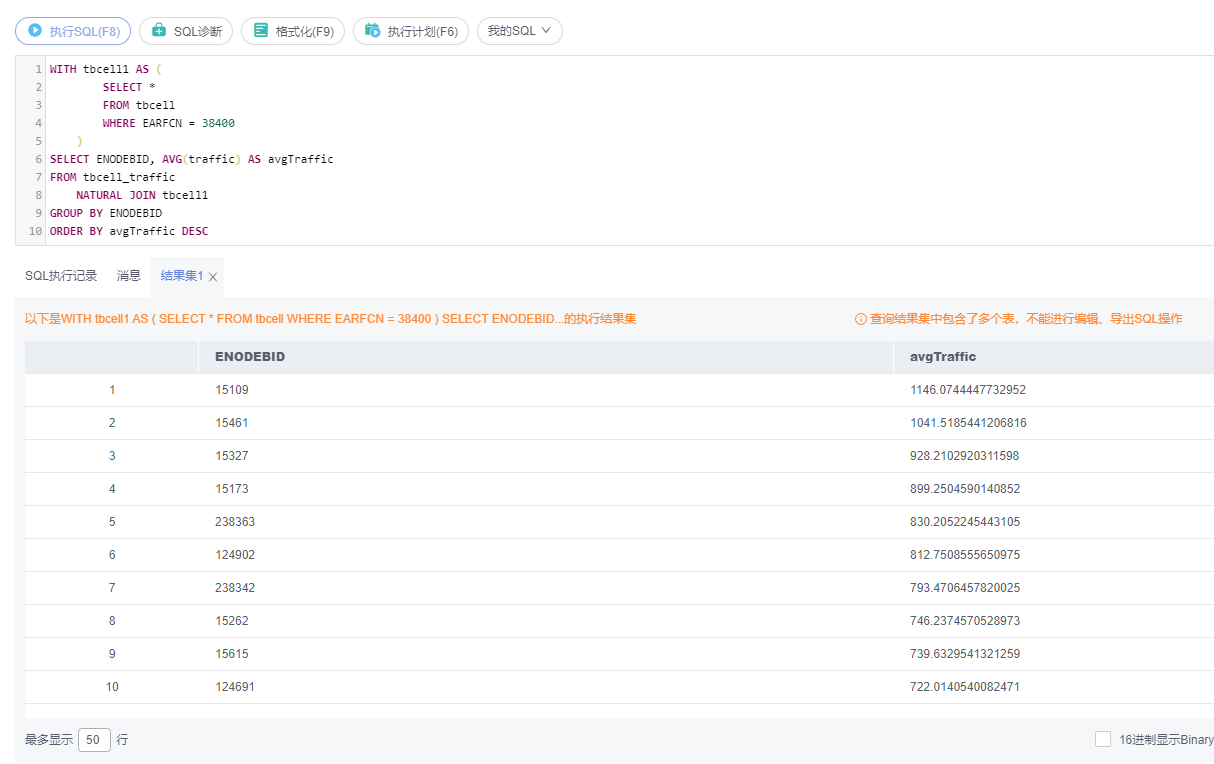
NATURAL JOIN tbcell

WHERE EARFCN = 38400

GROUP BY ENODEBID

ORDER BY avgTraffic DESC

**执行结果如下：**



#### **嵌套查询【参考教科书3.8节】**

查询14：从小区/基站信息表tbCell、优化小区/保护带小区tbOptCell和小区PCI优化调整结果表tbPCIAssignment中，使用set membership运算符in，查询小区类型为“优化区”的小区、PCI调整前后没有发生变化的小区，列出这些优化小区的ID和名称、调整前后的PCI。

对比使用多表连接、非嵌套的查询在执行时间、查询结果上的异同。（结果为空）

SQL查询语句：

1. 嵌套查询

SELECT Sector\_ID, SECTOR\_NAME, PCI

FROM tboptcell

NATURAL JOIN tbcell

WHERE CELL\_TYPE = '优化区'

AND PCI IN (

SELECT PCI

FROM tbpciassignment

WHERE tbpciassignment.SECTOR\_ID = tbcell.SECTOR\_ID

)

**执行结果如下：**



1. 非嵌套查询

SQL查询语句：

SELECT tbcell.Sector\_ID, tbcell.SECTOR\_NAME, tbcell.PCI

FROM tboptcell

NATURAL JOIN tbcell natural

JOIN tbpciassignment

WHERE CELL\_TYPE = '优化区'

**执行结果如下：**



1. 多表查询

SQL查询语句：

SELECT tbcell.Sector\_ID, tbcell.SECTOR\_NAME, tbcell.PCI

FROM tboptcell

NATURAL JOIN tbcell

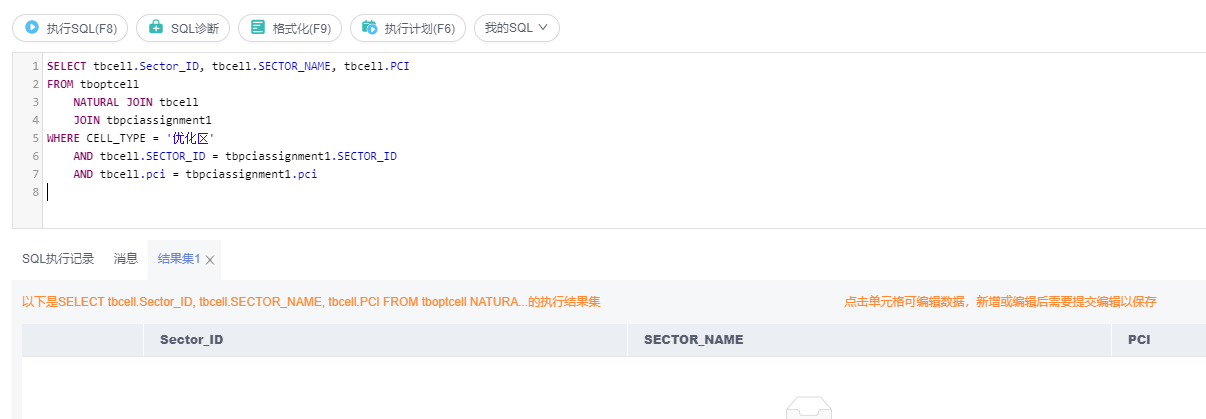
JOIN tbpciassignment

WHERE CELL\_TYPE = '优化区'

AND tbcell.SECTOR\_ID = tbpciassignment.SECTOR\_ID

AND tbcell.pci = tbpciassignment.pci

**执行结果如下：**



查询15-1：从小区/基站信息表tbCell，使用Set Comparison 运算符some，查询满足下列条件的小区：该小区的天线高度height高于位于经度在[?,?]、纬度在[?,?]区域内的部分（至少一个）小区的天线高度，列出这些小区的名称、标识和天线高度。

(1)SQL查询语句：

SELECT SECTOR\_ID, SECTOR\_NAME, HEIGHT

FROM tbcell

WHERE height > SOME (

SELECT height

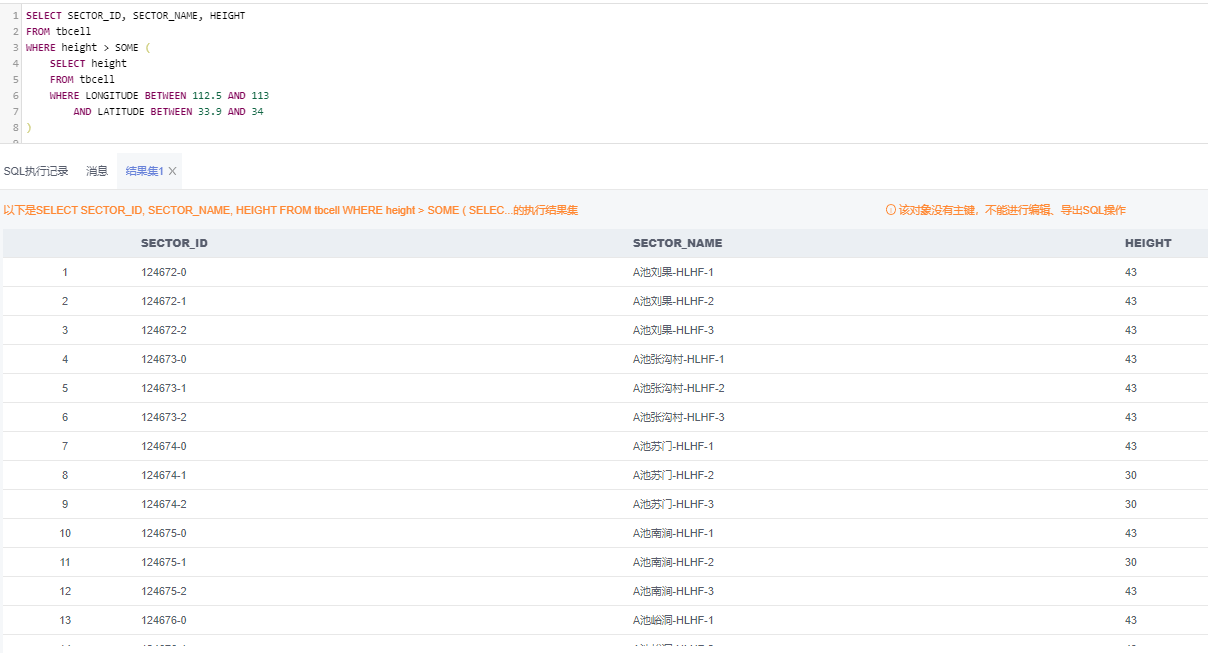
FROM tbcell

WHERE LONGITUDE BETWEEN 112.5 AND 113

AND LATITUDE BETWEEN 33.9 AND 34

)

**执行结果如下：**



查询15-2：从路测ATU数据表tbATUdata，使用Set Comparison 运算符some，查询满足下列条件的小区：在路测数据中作为主小区/服务小区CELLID，其参考信号接收功率RSRP，大于部分（基站标识ENodeBID=253903的小区作为主小区/服务小区时的参考信号接收功率RSRP。列出这些小区的ID、名称、在测量报告中作为主服务小区的RSRP。（不用再连接，直接让CellID=253903)就可以）

(2)SQL查询语句：

SELECT DISTINCT CellID, SECTOR\_NAME, RSRP

FROM tbatudata

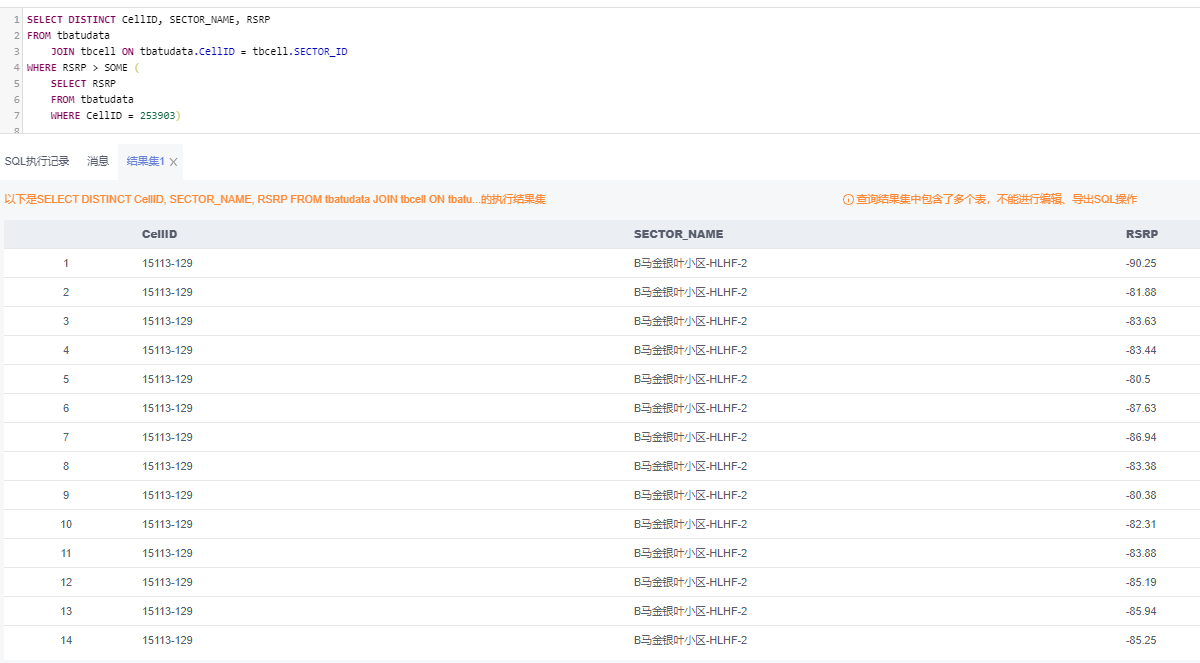
JOIN tbcell ON tbatudata.CellID = tbcell.SECTOR\_ID

WHERE RSRP > SOME (

SELECT RSRP

FROM tbatudata

WHERE CellID = 253903

) 

查询16-1：从小区小时级话务量表tbCellTraffic中，使用Set Comparison 运算符>=all，查询全年小时级话务量总和满足下列条件的小区：该小区的全年话务量总和大于等于其它小区的全年话务量总和，即该小区的全年话务量总和最高。

(1)SQL查询语句：

SELECT Sector\_ID, SUM(Traffic)

FROM tbcell\_traffic

GROUP BY Sector\_ID

HAVING SUM(Traffic) >= ALL (

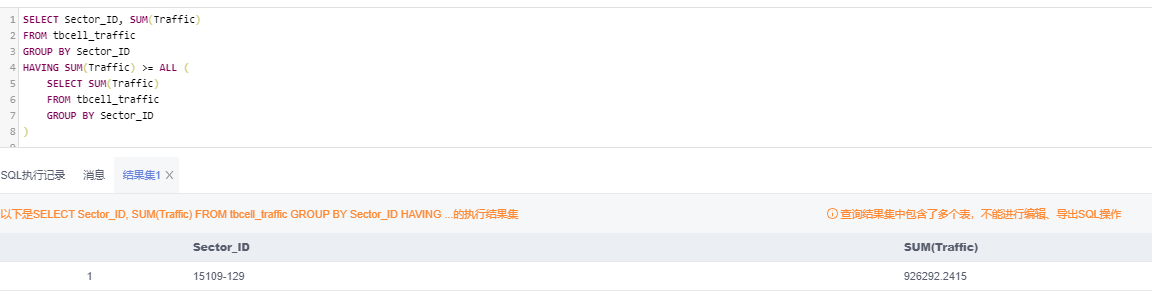
SELECT SUM(Traffic)

FROM tbcell\_traffic

GROUP BY Sector\_ID

)

**执行结果如下：**



查询16-2：切换统计表tbHandOver，使用Set Comparison 运算符all，查询作为源小区SCELL与周边目标/邻小区NCELL发生切换次数(HOATT)最多的小区。列出这些源小区的ID、目标/邻小区ID、发生的切换次数。

(2)SQL查询语句：

SELECT SCELL, NCELL, HOATT

FROM tbhandover

WHERE HOATT >= ALL (

SELECT HOATT

FROM tbhandover

)

**执行结果如下：**



查询17-1：从切换统计表tbHandOver表，使用Test for Empty Relations运算符“not exists”，查询作为源小区SCELL，其切换邻小区NCELL包含了{15290-128, 259595-1, 124711-0, 47444-1}中的全部四个小区。（mysql不支持except，但sql server中可以）

(1)SQL查询语句：exists

SELECT DISTINCT scell

FROM tbhandover t1

WHERE EXISTS (

SELECT \*

FROM tbhandover t2

WHERE t1.scell = t2.scell

AND t2.ncell = '15290-128'

)

AND EXISTS (

SELECT \*

FROM tbhandover t2

WHERE t1.scell = t2.scell

AND t2.ncell = '259595-1'

)

AND EXISTS (

SELECT \*

FROM tbhandover t2

WHERE t1.scell = t2.scell

AND t2.ncell = '124711-0'

)

AND EXISTS (

SELECT \*

FROM tbhandover t2

WHERE t1.scell = t2.scell

AND t2.ncell = '47444-1'

)

**执行结果如下：**



查询17-2：从一阶邻区表tbAdjCell、二阶邻区表tbSecAdjCell中，使用Test for Empty Relations 运算符“not exists”，查询满足下列条件的源小区S\_SECTOR\_ID：该小区的一阶邻小区包含其全部二阶邻小区，或者该小区的二阶邻小区包含其全部一阶邻小区。

(2)SQL查询语句：

SELECT k.S\_SECTOR\_ID

FROM tbadjcell AS k

WHERE NOT EXISTS (

(SELECT q.N\_SECTOR\_ID as N\_SECTOR\_ID

FROM tbadjcell AS q

WHERE q.S\_SECTOR\_ID = k.S\_SECTOR\_ID)

except(

SELECT z.N\_SECTOR\_ID as N\_SECTOR\_ID

FROM tbsecadjcell AS z

WHERE z.S\_SECTOR\_ID = k.S\_SECTOR\_ID))

**执行结果为空**

查询18：从小区/基站信息表tbCell、一阶邻区表tbAdjCell中，使用Test for Absence of Duplicate Tuples运算符not unique（不是mysql关键字），查询满足下列条件的源小区：ENodeBID=15114的基站下有多个小区，该源小区至少是基站15114下2个小区的邻小区。

SQL查询语句：(mysql不支持not unique，所以用count计数)

SELECT DISTINCT S\_SECTOR\_ID

FROM tbadjcell t1

WHERE 2 <= (

SELECT COUNT(\*)

FROM tbadjcell t2

WHERE t1.S\_SECTOR\_ID = t2.S\_SECTOR\_ID

AND t2.N\_SECTOR\_ID = 15114

);

SELECT DISTINCT S\_SECTOR\_ID

FROM tbadjcell t1

WHERE 2 <= (

SELECT COUNT(\*)

FROM tbadjcell t2, tbcell

WHERE t1.S\_SECTOR\_ID = t2.S\_SECTOR\_ID

AND t2.N\_SECTOR\_ID = tbcell.SECTOR\_ID

AND tbcell.ENODEBID = 15114

)

**执行结果如下：**



查询19：从小区KPI性能表tbCellKPI中，使用Subqueries in the From Clause方法，查询满足下列条件的小区：小区在2020/07/17-2020/0719这三天的平均RRC建立成功率大于0.992，给出这些小区的ID和其三天平均连接成功率。

“DMY”需改为“MDY”，即：月日年

SQL查询语句：嵌套

SELECT 小区名称, AVG\_QF

FROM (

SELECT 小区名称, AVG(RRC建立成功率qf) AS AVG\_QF

FROM tbCellKPI

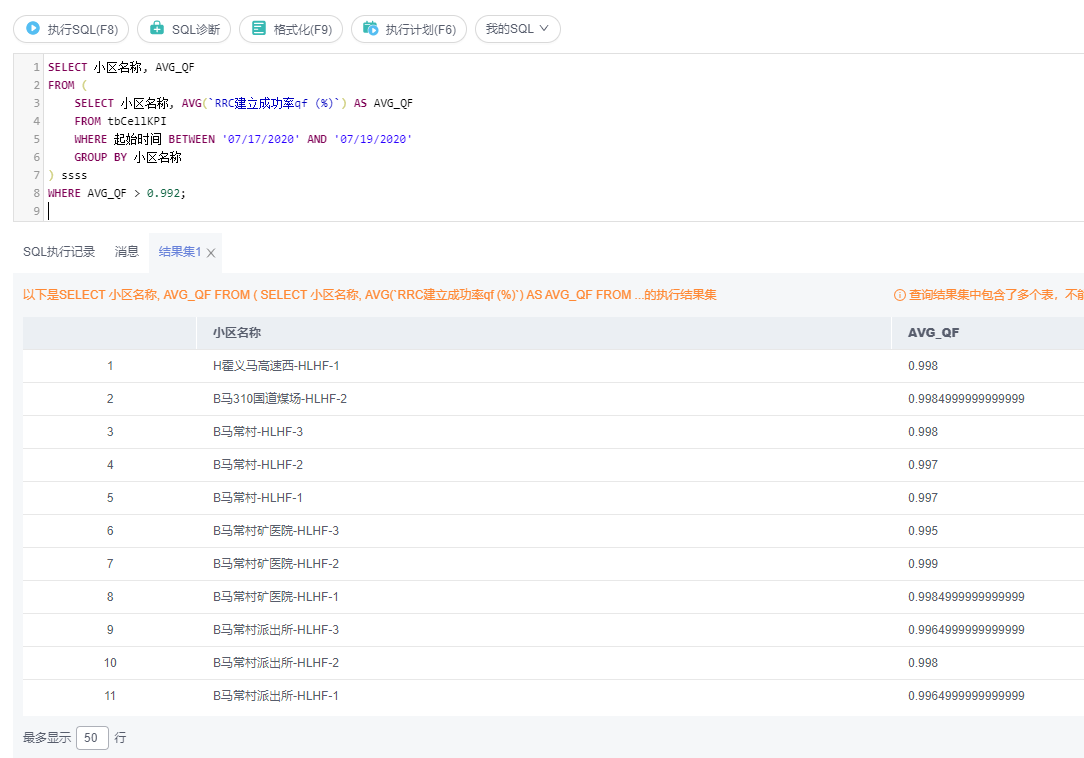
WHERE 起始时间 BETWEEN '2020-07-17' AND '2020-07-19'

GROUP BY 小区名称

) ssss（需有别名）

WHERE AVG\_QF > 0.992;

**执行结果如下：**



SQL查询语句：having by

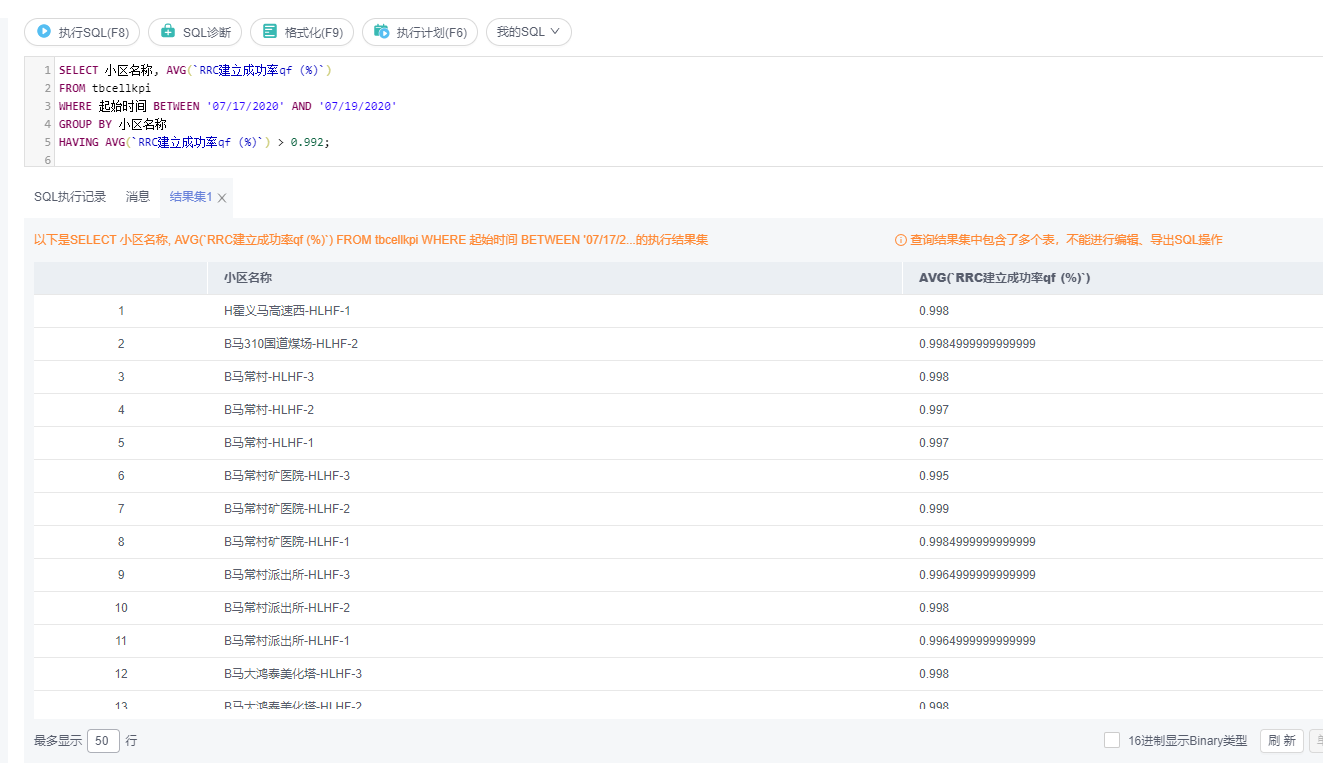
SELECT 小区名称, AVG(RRC建立成功率qf)

FROM tbcellkpi

WHERE 起始时间 BETWEEN '07/17/2020' AND '07/19/2020'

GROUP BY 小区名称

HAVING AVG(RRC建立成功率qf) > 0.992

**执行结果如下：**

#### **with临时视图查询【参考教科书3.8.6节】**

查询20：用with临时视图方式，实现查询19中查询要求。

SQL查询语句：

WITH temp (xiaoqu, chenggonglv) AS (

SELECT `小区`, AVG(`RRC建立成功率qf (%)`)

FROM `tbcellkpi`

GROUP BY `小区`

)

SELECT xiaoqu, chenggonglv

FROM temp

WHERE chenggonglv > 0.992;

**执行结果如下：**



查询21：从小区/基站信息表tbCell、一阶邻区表tbAdjCell中，用with临时视图方式，查询一阶邻小区最多的主小区，给出这些主小区的ID、邻小区数目。

SQL查询语句：

WITH adj\_number (SECTOR\_ID, number) AS (

SELECT S\_SECTOR\_ID, COUNT(N\_SECTOR\_ID)

FROM `tbadjcell`

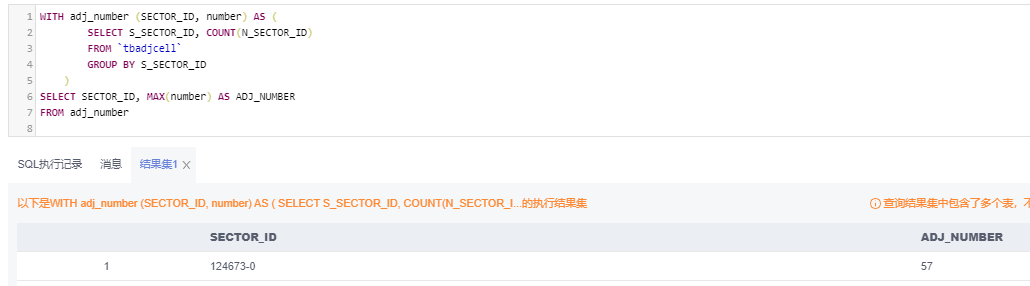
GROUP BY S\_SECTOR\_ID

)

SELECT SECTOR\_ID, MAX(number) AS ADJ\_NUMBER

FROM adj\_number

**执行结果如下：**



#### **键/函数依赖分析【参考第8章作业题8.9】**

查询22：在MRO测量报告数据表tbMROData中，检查TimeStamp、ServingSector、InterferingSector是否组成超键。

SQL查询语句：

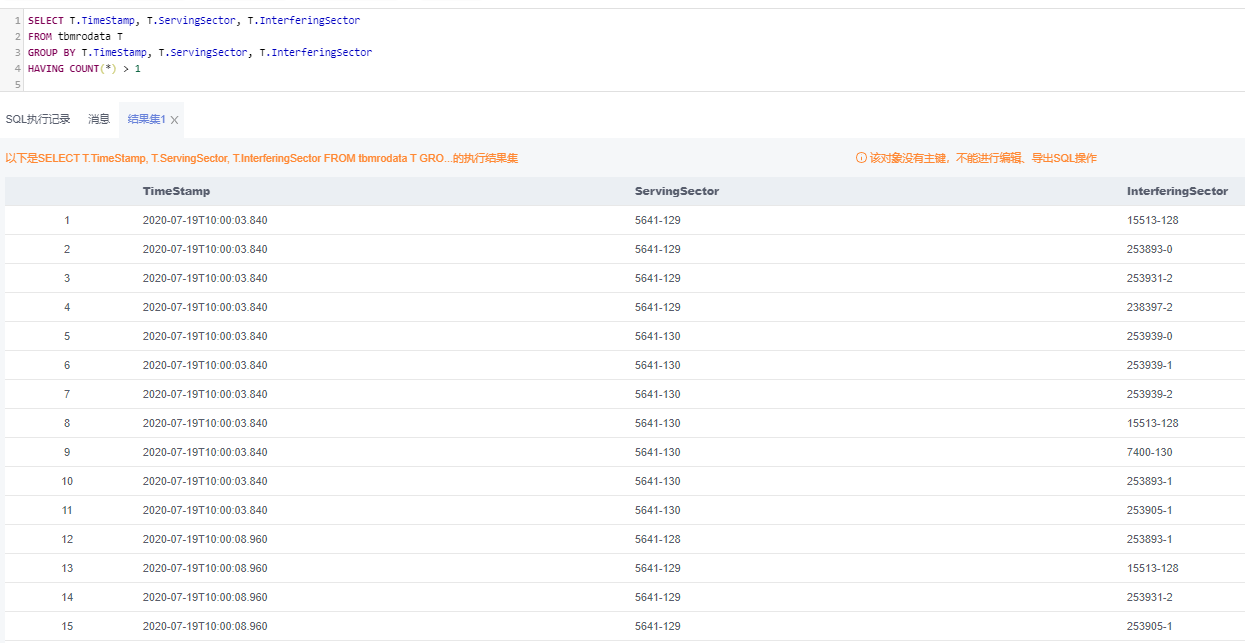
SELECT T.TimeStamp, T.ServingSector, T.InterferingSector

FROM tbmrodata T

GROUP BY T.TimeStamp, T.ServingSector, T.InterferingSector

HAVING COUNT(\*) > 1

**执行结果如下：**



查询23：在PCI优化分配表tbPCIAssignment中，利用SQL语句检查函数依赖ENODEB\_IDPCI是否成立；如果不成立，利用SQL语句找出导致函数依赖不成立的元组。

1. 检查函数依赖是否成立：

SQL查询语句：

SELECT ENODEB\_ID

FROM tbpciassignment

GROUP BY ENODEB\_ID

HAVING COUNT(DISTINCT PCI) > 1;

**执行结果如下：**



可以看出，函数依赖ENODEB\_ID->PCI不成立。

1. 找出导致函数依赖不成立的元组：

SQL查询语句：

SELECT \*

FROM tbpciassignment a

WHERE a.ENODEB\_ID IN (

SELECT b.ENODEB\_ID

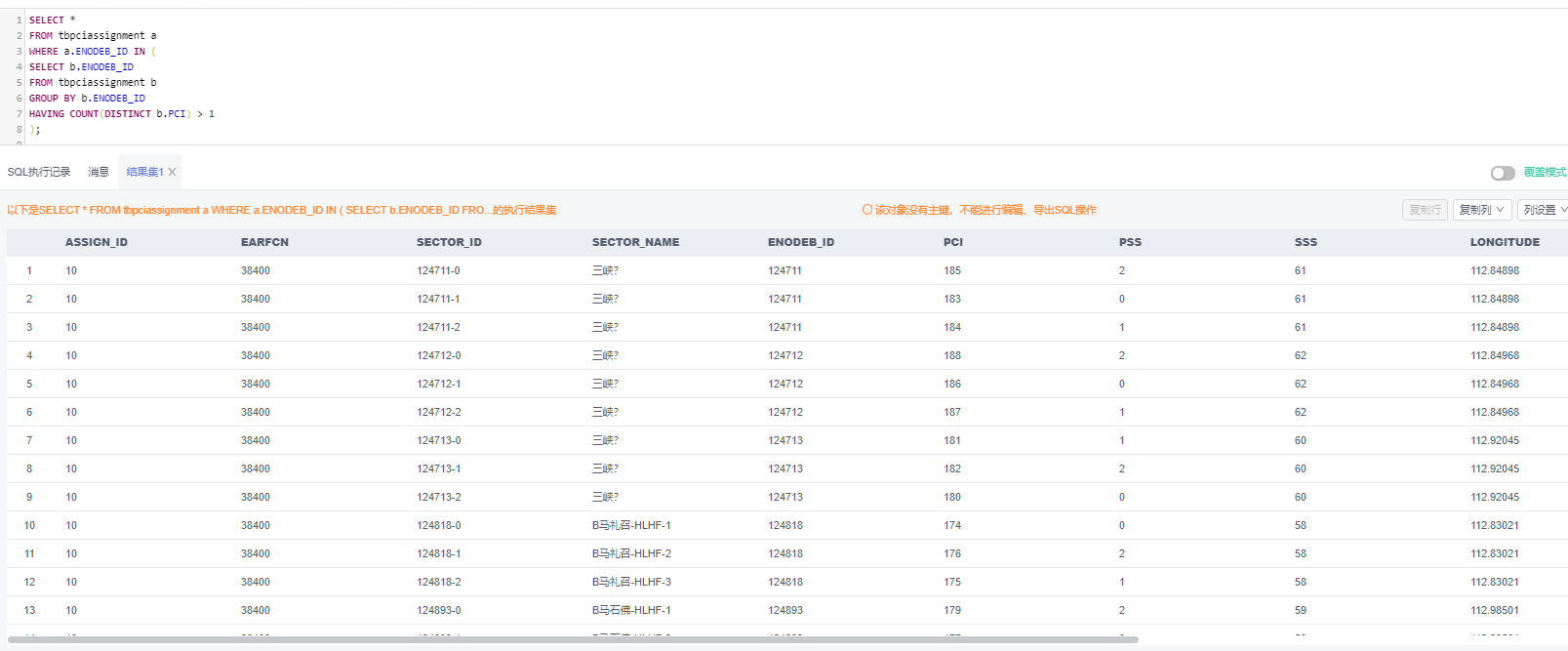
FROM tbpciassignment b

GROUP BY b.ENODEB\_ID

HAVING COUNT(DISTINCT b.PCI) > 1

);

**执行结果如下：**



#### **关系表的插入/删除/更新**

查询24：向小区一阶邻区关系表中插入一条邻区数据；

1. 插入一条邻区数据：

SQL查询语句：

INSERT INTO tbadjcell (S\_SECTOR\_ID, N\_SECTOR\_ID, S\_EARFCN, N\_EARFCN)

VALUES ('124673-0', '5691-128', '38400', '38400');

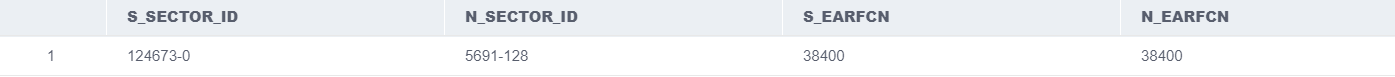
（2）查看插入结果：

SELECT \*

FROM tbadjcell

WHERE S\_SECTOR\_ID LIKE '124673-0';

**执行结果如下：**



查询25：将小区124673-0的全部二阶邻小区，作为该小区的一阶邻小区，加入到一阶邻区表tbAdjCell中。

SQL查询语句：

INSERT INTO tbadjcell

SELECT C.S\_SECTOR\_ID, C.N\_SECTOR\_ID, A.EARFCN, B.EARFCN

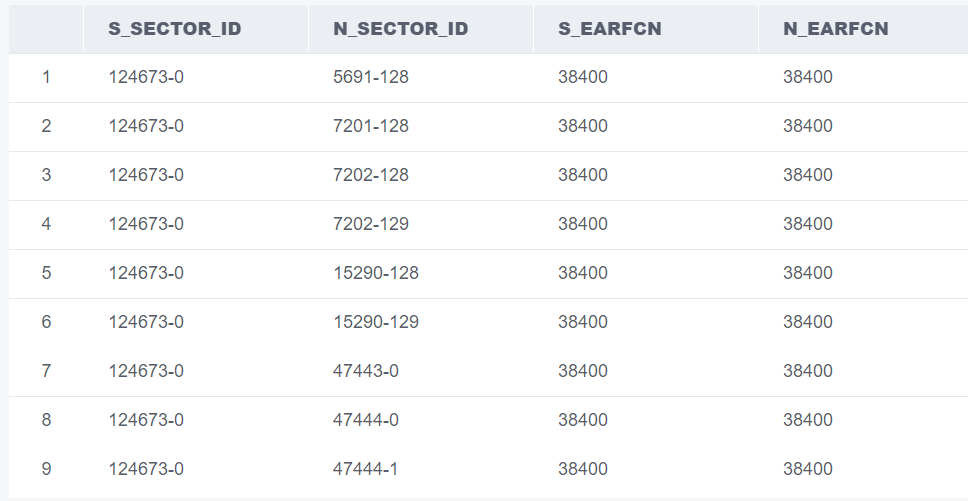
FROM tbsecadjcell C

LEFT JOIN tbcell A ON C.S\_SECTOR\_ID = A.SECTOR\_ID

LEFT JOIN tbcell B ON C.N\_SECTOR\_ID = B.SECTOR\_ID

WHERE C.S\_SECTOR\_ID = '124673-0'

**执行结果如下：**

****

查询26：在小区切换统计性能表tbHandover中，删除切换次数最少的那些切换数据。

SQL查询语句：

DELETE FROM `tbhandover` a

WHERE a.HOATT = (

SELECT c.hoatt

FROM (

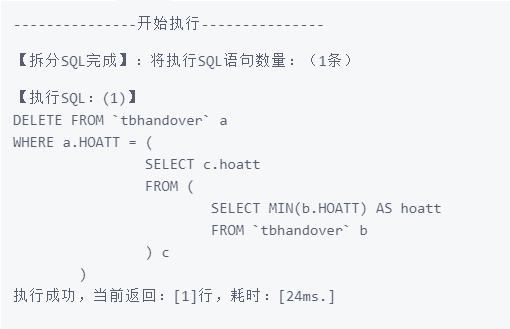
SELECT MIN(b.HOATT) AS hoatt

FROM `tbhandover` b

) c

);

**执行结果如下：**



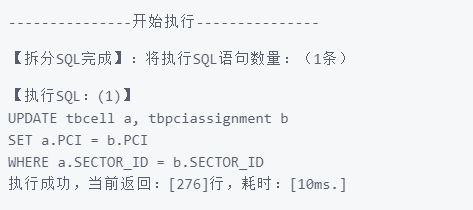
查询27：用小区PCI优化调整结果表tbPCIAssignmeng给出的各个小区的优化调整后的PCI值，替换小区/基站消息表tbCell中对应小区的物理小区标识PCI。

SQL查询语句：

UPDATE tbcell a, tbpciassignment b

SET a.PCI = b.PCI

WHERE a.SECTOR\_ID = b.SECTOR\_ID;



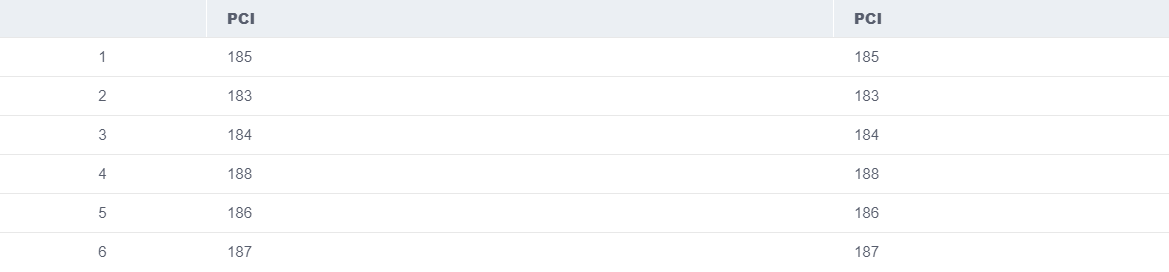
检查替换结果：

SELECT a.PCI, b.PCI

FROM tbcell a, tbpciassignment b

WHERE a.SECTOR\_ID = b.SECTOR\_ID;

**执行结果如下：**



查询28：针对路测ATU干扰矩阵表tbATUC2I，使用update/case语句做出如下修改：对主小区SECTOR\_ID=238397-1，如果该小区与干扰小区N\_SECTOR\_ID为同站小区(cosite=1)且干扰强度排序rank不小于1，则干扰强度排序减1；如果主小区与干扰小区不同站，干扰强度排序加1。

SQL查询语句：

UPDATE tbatuc21

SET tbatuc21.RANK = CASE

WHEN tbatuc21.COSITE = 1 THEN tbatuc21.RANK - 1

ELSE tbatuc21.RANK + 1

END

WHERE tbatuc21.SECTOR\_ID = '238397-1'

AND tbatuc21.RANK >= 1

**执行结果如下：**

