

# 第一章 数据查询与修改实验

## 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。

## 实验环境

采用 GaussDB(MySQL)数据库管理系统作为实验平台。

## 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 T<sub>1</sub> 和 T<sub>2</sub>，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/natural 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标识和切换目标小区, 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/alias 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 1~3 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 2020/07/17-2020/07/19 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\_ID→PCI 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

1. 用 Transact\_SQL 语句完成以上操作。
2. 要求学生独立完成以上内容。
3. 实验完成后完成要求的实验报告内容。

## 实验总结