

实验三：数据基本查询

1. 实验环境

- 华为 ECS+openGauss 数据库服务器平台
- **前提：**openGauss 数据库服务器正常运行
- 已建立带样例数据的 SALES 数据库

2. 实验目的

- 熟练掌握 openGauss 单表查询的语法结构及其使用方法
- 掌握设计正确查询语句以实现查询要求的方法
 - 简单单表查询（**此处指不涉及模糊、集合、聚集、分组、排序的查询**）
 - 模糊查询、聚集函数、分组统计和排序
- 掌握 Group by 的使用
- 正确区分元组过滤条件（WHERE 子句）和分组过滤条件（HAVING 短语）的异同
- 掌握 Order by 的使用
- 掌握使用 DISTINCT 实现查询结果的去重方法
- 掌握空值 NULL 的使用方法
- 掌握表别名的使用场合及方法
- 掌握自身连接的使用方法

3. 实验要求

- 设计正确的 SQL 查询语句并测试其结果是否满足查询要求。
- 完成实验内容并按实验报告模板撰写实验报告，实验报告提交到 FTP 上的相应文件夹“实验三”。
- 实验报告提交截至日期：**2022 年 4 月 14 日星期四**。

4. 实验内容与步骤

4.1 实验内容

- (1) 查询顾客表中的顾客号（customer_id）、顾客名（name）和信用卡额度（credit_limit）
- (2) 查询顾客的所有信息
- (3) 查询订单表中的订单号，顾客号，状态，订单日期，并按订单日期降序显示结果
- (4) 查询联系表中的名（first name）和姓（last name），并按名升序，姓降序显示
- (5) 执行以下语句并观察 state 列 NULL 值的显示位置，得出结论

```
SELECT country_id, city, state FROM locations ORDER BY city, state;

SELECT country_id, city, state FROM locations ORDER BY state ASC NULLS FIRST;

SELECT country_id, city, state FROM locations ORDER BY state ASC NULLS LAST;
```

- (6) 查询订单细节表中 (order_items) 的产品号和数量，查询结果应无重复元组
- (7) 查询产品表中的产品名为 ‘Kingston’ 的产品名，产品描述和价格
- (8) 查询产品表中所有价格大于 500 且 category_id 为 4 的产品名和价格
- (9) 查询产品表中所有价格在 650 和 680 之间的产品名和价格并按价格升序显示结果
- (10) 查询雇员表中的名和姓，名和姓的字段分别显示为 "First Name" 和 "Family Name"
- (11) 查询产品表中的产品名及毛利，并按毛利结果降序显示，毛利名为 gross_profit，毛利 = list_price - standard_cost
- (12) 查询雇员表中每个雇员对应的经理名，要求第一列字段名为 employee_name，第二列字段名为 manager_name（雇员和经理的姓名同一格式为 ‘first_name, last_name’）

说明： 此处查询涉及字符串的连接。字符串连接符号为 “||”，即两个连续的竖线。如

```
openGauss=# SELECT 'MPP' || 'DB' AS RESULT;

result
-----
MPPDB
(1 row)
```

语法参考：<https://opengauss.org/zh/docs/3.0.0/docs/Developerguide/%E5%AD%A7%E7%AC%A6%E5%A4%E7%90%86%E5%87%BD%E6%95%B0%E5%92%8C%E6%93%8D%E4%BD%9C%E7%AC%A6.html>

- (13) 查询产品表中所有以 Asus 开头的产品名和价格，并以价格降序显示
- (14) 查询联系表中电话号码不是以 ‘+1’ 开头的名、姓和电话号码，并以名升序显示
- (15) 查询联系表中的电话号码和电子邮件，要求名 (first_name) 的长度为 4 且以 'Je' 开头，以 'i' 结尾，按名升序显示
- (16) 查询联系表中所有以开头 'Je' 的名，且至少包含 3 个字符的名，姓，电子邮件和电话
- (17) 查询订单表中所有没有销售员负责的订单 (i.e., query all sales orders that do not have a responsible salesman)
- (18) 统计每个顾客的订单总数（查询订单表）
- (19) 统计每个订单的总价格大于 1000000 的订单号和总价格，并按总价格降序显示结果（查询订单细节表 order_items，总价格=unit_price*quantity）
- (20) 创建一个折扣表 discounts

```
CREATE TABLE discounts

( product_id NUMBER, --产品号，主码

  discount_message VARCHAR2( 255 ) NOT NULL, --折扣信息

  PRIMARY KEY( product_id ) );
```

插入 3 条数据：

```
INSERT INTO discounts(product_id, discount_message) VALUES(1, 'Buy 1 and Get 25% OFF on 2nd ');  
  
INSERT INTO discounts(product_id, discount_message) VALUES(2, 'Buy 2 and Get 50% OFF on 3rd ');  
  
INSERT INTO discounts(product_id, discount_message) VALUES(3, 'Buy 3 Get 1 free');
```

要求：查询折扣表中折扣信息出现“25%”的产品号和折扣信息。

4.2 实验步骤

(1) 以 root 用户登录到 ECS 服务器>以 omm 操作系统管理员身份登录数据库>使用 gsql 连接到数据库。操作步骤见《openGauss 开发者指南》中的 3.2.3 节。

(2) 依次完成 4.1 中的实验内容。

5. 实验思考

- 使用 GROUP BY 时 SELECT 关键词后面的列及其次序有何要求？请上机验证你的看法。

6. 参考资料

- openGauss 开发者指南.pdf 之 SELECT 语法（16.14.165 节 SELECT）
- 实验一教程
- openGauss 松鼠会： openGauss.org
- 墨天轮： <https://www.modb.pro/tag/openGauss>
- 数据库原理实验(openGauss)交互式 SQL 语句 <https://bokai.blog.csdn.net/article/details/117871367>