《数据库系统原理》课程实验指导

openGauss 数据查询与修改

2023年9月

目录

前 言

实验环境说明

本实验环境为 virtualBOX 虚拟机 openEuler20.03 系统上的 openGauss1.1.0/openGauss2.0.0 数据库和华为云 GaussDB(openGauss)数据库，实验数据采用电商数据库的八张表。

1 查询与修改实验

1.1 实验目的

对前面实验建立的电商数据库关系表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。

1.2 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；

1.3 实验步骤

依次完成以下各个查询实验。

1.3.1 单表查询

查询 1:从订单表 ORDERS 表中,找出由收银员 Clerk#000000951 处理的满足下列条件的所有订单 o_orderkey:

(1) 订单总价位于[？,？]，并且

(2) 下单日期在？至？之间，并且

(3) 订单状态 O_ORDERSTATUS 不为空

，列出这些订单的订单 key（O_ORDERKEY）、客户 key、订单状态、订单总价、下单日期、订单优先级和发货优先级；

要求：对查询结果，按照订单优先级从高到低、发货优先级从高到低排序，并且将 O_ORDERDATE 重新命名为 O_DATE。

说明：？代表由学生自己选择输入条件

查询 2：从订单明细表 LINEITEM 表中，找出满足下列条件的所有订单 L_ORDERKEY：

（1）数量位于[？,？]，

（2）退货标志为‘N’的订单中，价格不小于?

，列出这些订单的 key 和零件供应商 key、价格；要求：对查询结果，按照价格从高到低排序，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。

1.3.2 String 操作

查询 3：从客户表 CUSTOMER 表中，找出满足下列条件的客户：

（1）客户电话开头部分包含‘10’，或者客户市场领域中包含“BUILDING”，并且

（2）客户电话结尾不为‘8’。

查询 4：从客户表 CUSTOMER 表中，找出满足下列条件的客户姓名：

（1）客户 key 由 2 个字符组成，并且

（2）客户地址至少包括 18 个字符，即地址字符串的长度不小于 18。

1.3.3 集合操作

查询 5：使用集合并操作 union、union all，从订单明细表 LINEITEM 查询满足下列条件的订单 key

(1)订单发货日期早于‘2016-01-01’，或者

(2)订单数量大于 100

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，从供应商表 SUPPLIER 中，查询账户余额最大的供应商。

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

1.3.4 多表查询

查询 7：选取两张数据量比较小的表 T_1 和 T_2，如地区表 REGION、国家表 NATION、供应商表 SUPPLIER，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），从订单表 ORDERS、供应商表 SUPPLIER、订单明细表 LINEITEM 中，查询实际到达日期小于预计到达日期的订单，列出这些订单的订单 key、订单总价、下单日期以及该供应商的姓名、地址和手机号。

查询 9：使用多表连接操作，从供应商表 SUPPLIER、零部件表 PART、零部件供应表 PARTSUPP 中，查询供应零件品牌为‘Brand#13’的供应商信息，列出零件供应数量与成本，以及供应商的姓名与手机号。

查询 10：利用订单明细表 LINEITEM，使用教材 3.4.1 节元组变量 as/rename 方式，查询所有比流水号为“1”，订单号为“1”的折扣高的订单 key 和流水号，列出这些订单的零件、折扣，结果按照折扣的降序排列。

1.3.5 聚集函数

查询 11：从订单明细表 LINEITEM、订单表 ORDERS、客户表 CUSTOMER、国家表 NATION，查询客户来自 ALGERIA，下单日期为'2015-01-01'到'2015-02-02'的订单下列信息：

（1）满足条件订单的最大数量、最小数量和平均数量。

（2）具有最大数量且满足上述条件的订单，列出该订单的发货日期、下单日期。

查询 12：根据零部件表 PART 和零部件供应表 PARTSUPP 和供应商表 SUPPLIER，查询有多少零件厂商提供了品牌为 Brand#13 的零件，给出这些零件的类型、零售价和供应商数量，并将查询结果按照零售价降序排列。

查询 13：从零部件表 PART 和零部件供应表 PARTSUPP 中，查询所有零件大小在[7,14]之间的零件的平均零售价，给出零件 key，供应成本，平均零售价，结果按照零售价降序排列。

1.3.6 嵌套查询

查询 14：从订单明细表 LINEITEM、订单表 ORDERS、客户表 CUSTOMER 中，使用 set membership 运算符 in，查询明细折扣小于 0.01 的订单，列出这些订单的 key 和采购订单的客户姓名。

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：从订单明细表 LINEITEM，使用 Set Comparison 运算符 some，查询满足下列条件的订单：该订单的数量大于发货日期在[?,?]之间的部分（至少一个）订单的数量，列出这些订单的流水号、key 和税。

查询 15-2：从订单表 ORDERS，使用 Set Comparison 运算符 some，查询满足下列条件的订单：订单状态为‘O’，订单总价大于部分在 2020 年之后下单的订单。列出这些订单的 key、客户 key、收银员。

查询 16-1：从订单明细表 LINEITEM 中，使用 Set Comparison 运算符>=all，查询满足下列条件的供应商：该供应商在 2019 年出货量大于等于同时段其他供应商的出货量，即 2019 年该供应商的出货量最高。

查询 16-2：供应商表 SUPPLIER，使用 Set Comparison 运算符 all，查询账户余额大于等于其他供应商的供应商。列出该供应商的姓名、key、手机号。

查询 17-1：从供应商表 SUPPLIER、国家表 NATION，使用 Test for Empty Relations 运算符“exists”，查询国家为日本，账户余额大于 5000 的供应商。

查询 17-2：从客户表 CUSTOMER、国家表 NATION、订单表 ORDERS、订单明细表 LINEITEM、供应商表 SUPPLIER 中，使用 Test for Empty Relations 运算符“not exists except”，查询满足下列条件的供应商：该供应商不能供应所有的零件。

查询 18：从国家表 NATION、客户表 CUSTOMER 中，使用“count”，查询满足下列条件的国家：至少有 3 个客户来自这个国家，并列出该国家的国家 key 和国家名。

查询 19：从零部件表 PART 和零部件供应表 PARTSUPP 中，使用 Subqueries in the From Clause 方法，查询满足下列条件的零件：零件由 2 个以上的供应商供应，且零件大小在 20 以上。

1.3.7 with 临时视图查询

查询 20：用 with 临时视图方式，实现查询 19 中查询要求。

查询 21：从零部件供应表 PARTSUPP 中，用 with 临时视图方式，查询零件供应数量最多的供应商 key 和其供应的数量。

1.3.8 键/函数依赖分析

查询 22：在订单明细表 LINEITEM 中，检查订单 key、零件 key、供应商 key、流水号是否组成超键。

查询 23：在订单明细表 LINEITEM 中，利用 SQL 语句检查函数依赖零件 key→价格是否成立；如果不成立，利用 SQL 语句找出导致函数依赖不成立的元组。

1.3.9 关系表的插入/删除/更新

查询 24：向订单表 ORDERS 中插入一条订单数据；

查询 25：将零件 32 的全部供应商，作为零件 20 的供应商，加入到零部件供应表 PARTSUPP 中。

查询 26：在订单明细表 LINEITEM 中，删除已退货的订单记录。(returnflag='R')

查询 27：用订单明细表 LINEITEM 中在 2019 年之后交易中的预计到达日期，替换表中的实际到达日期。

查询 28：针对订单明细表 LINEITEM、订单表 ORDERS，使用 update/case 语句做出如下修改：如果订单的订单优先级低于 medium，则其在订单明细表中的预计到达日期推后 2 天，否则推迟一天。

查询 29：在订单表 ORDERS 中，利用 Rank 函数，按照订单总价对订单进行降序排序，并输出订单 key 和排名。

1.4 实验要求

用 postgreSQL 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

2 实验示例

单表查询

查询 1：从订单表 ORDERS 表中，找出由收银员 Clerk#000000951 处理的满足下列条件的所有订单 o_orderkey：

（1）订单总价位于[？,？]，并且

（2）下单日期在？至？之间，并且

（3）订单状态 O_ORDERSTATUS 不为空

，列出这些订单的订单 key（O_ORDERKEY）、客户 key、订单状态、订单总价、下单日期、订单优先级和发货优先级；

要求：对查询结果，按照订单优先级从高到低、发货优先级从高到低排序，并且将 O_ORDERDATE 重新命名为 O_DATE。

说明：？代表由学生自己选择输入条件

```
select o_orderkey, o_custkey, o_orderstatus, o_totalprice, o_orderdate, o_orderpriority, o_shippriority
from orders
where o_clerk='Clerk#000000951'
and o_totalprice between 0 and 10000000
and o_orderdate between '2015-01-01'::date and '2021-12-30'::date
and o_orderstatus is not null
order by o_orderpriority desc, o_shippriority desc;
```

查询 2：从订单明细表 LINEITEM 表中，找出满足下列条件的所有订单 L_ORDERKEY：

（1）数量位于[？,？]，

（2）退货标志为‘N’的订单中，价格不小于？

，列出这些订单的 key 和零件供应商 key、价格；要求：对查询结果，按照价格从高到低排序，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。

```
select distinct l_orderkey, l_suppkey, l_extendedprice
from lineitem
where l_quantity between 10 and 100
```

```
and l_returnflag='N'
and l_extendedprice >= 10000
order by l_extendedprice desc;
```

String 操作

查询 3：从客户表 CUSTOMER 表中，找出满足下列条件的客户：

（1）客户电话开头部分包含‘10’，或者客户市场领域中包含“BUILDING”，并且

（2）客户电话结尾不为‘8’。

```
select c_custkey, c_name from customer
where c_phone like '10%'
or c_mktsegment like 'BUILDING'
and c_phone not like '%8';
```

查询 4：从客户表 CUSTOMER 表中，找出满足下列条件的客户姓名：

（1）客户 key 由 2 个字符组成，并且

（2）客户地址至少包括 18 个字符，即地址字符串的长度不小于 18。

```
select c_name
from customer
where c_custkey like '__'
and length(c_address) >= 18;
```

集合操作

查询 5：使用集合并操作 union、union all，从订单明细表 LINEITEM 查询满足下列条件的订单 key

(1)订单发货日期早于‘2016-01-01’，或者

(2)订单数量大于 100

对比 union all、union 操作在查询结果、执行时间上的差异。

（1）

```
select l_orderkey
from lineitem
where l_shipdate < '2016-01-01'::date
union all
select l_orderkey
from lineitem
where l_quantity > 100;
```

（2）

```
select l_orderkey
from lineitem
where l_shipdate < '2016-01-01'::date
union
select l_orderkey
from lineitem
where l_quantity > 100;
```

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，从供应商表 SUPPLIER 中，查询账户余额最大的供应商。

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

(1) **except**

```
select s_suppkey, s_name
from supplier
except(
    select t1.s_suppkey, t1.s_name
    from supplier t1, supplier t2
    where t1.s_acctbal < t2.s_acctbal
);
```

(2) **except all**

```
select s_suppkey, s_name
from supplier
except all(
    select t1.s_suppkey, t1.s_name
    from supplier t1, supplier t2
    where t1.s_acctbal < t2.s_acctbal
);
```

(3) **max**

```
select s_suppkey, s_name
from supplier
where s_acctbal=(
    select max(s_acctbal)
    from supplier
```

);

多表查询

查询 7：选取两张数据量比较小的表 T_1 和 T_2，如地区表 REGION、国家表 NATION、供应商表 SUPPLIER，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
```

```
select * from region, nation;
```

查询 8：使用多表连接操作（3.3.3 join/natural join，4.1.1 join），从订单表 ORDERS、供应商表 SUPPLIER、订单明细表 LINEITEM 中，查询实际到达日期小于预计到达日期的订单，列出这些订单的订单 key、订单总价、下单日期以及该供应商的姓名、地址和手机号。

```
select o_orderkey, o_totalprice, o_orderdate, s_name, s_address, s_phone
from lineitem t1 join orders
on l_orderkey = o_orderkey, supplier
where s_suppkey = t1.l_suppkey
and t1.l_receiptdate < t1.l_commitdate;
```

查询 9：使用多表连接操作，从供应商表 SUPPLIER、零部件表 PART、零部件供应表 PARTSUPP 中，查询供应零件品牌为‘Brand#13’的供应商信息，列出零件供应数量与成本，以及供应商的姓名与手机号。

```
select ps_availqty, ps_supplycost, s_name, s_phone
from partsupp t1 join supplier on t1.ps_suppkey = s_suppkey, part
where t1.ps_partkey = p_partkey
and part.p_brand = 'Brand#13';
```

查询 10：利用订单明细表 LINEITEM，使用教材 3.4.1 节元组变量 as/rename 方式，查询所有比流水号为“1”，订单号为“1”的折扣高的订单 key 和流水号，列出这些订单的零件、折扣，结果按照折扣的降序排列。

```
select l_orderkey, l_linenumber as number, l_partkey, l_discount
from lineitem
where l_discount > (
    select l_discount
    from lineitem
    where l_orderkey = '1'
    and l_linenumber = '1'
)
order by l_discount desc;
```

聚集函数

查询 11：从订单明细表 LINEITEM、订单表 ORDERS、客户表 CUSTOMER、国家表 NATION，查询客户来自 ALGERIA，下单日期为'2015-01-01'到'2015-02-02'的订单下列信息：

（1）满足条件订单的最大数量、最小数量和平均数量。

（2）具有最大数量且满足上述条件的订单，列出该订单的发货日期、下单日期。

(1)

```
select max(l_quantity), min(l_quantity), avg(l_quantity)
from lineitem, customer, nation, orders
where lineitem.l_orderkey = orders.o_orderkey
and orders.o_custkey = customer.c_custkey
and customer.c_nationkey = nation.n_nationkey
and nation.n_name = 'ALGERIA'
and orders.o_orderdate between '2015-01-01'::date and '2015-02-02'::date;
```

(2)

```
select l_quantity, l_shipdate, o_orderdate
from lineitem, customer, nation, orders
where lineitem.l_orderkey = orders.o_orderkey
and orders.o_custkey = customer.c_custkey
and customer.c_nationkey = nation.n_nationkey
and nation.n_name = 'ALGERIA'
and orders.o_orderdate between '2015-01-01'::date and '2015-02-02'::date
order by l_quantity desc
limit 1;
```

查询 12：根据零部件表 PART 和零部件供应表 PARTSUPP 和供应商表 SUPPLIER，查询有多少零件厂商提供了品牌为 Brand#13 的零件，给出这些零件的类型、零售价和供应商数量，并将查询结果按照零售价降序排列。

```
select p_mfgr, p_type, p_retailprice, count(s_suppkey) as supplier_number
from part, partsupp, supplier
where partsupp.ps_partkey = part.p_partkey
and partsupp.ps_suppkey = s_suppkey
and part.p_brand='Brand#13'
group by p_mfgr, p_type, p_retailprice
order by p_retailprice desc;
```

查询 13：从零部件表 PART 和零部件供应表 PARTSUPP 中，查询所有零件大小在[7,14]之间的零件的平均零售价，给出零件 key，供应成本，平均零售价，结果按照零售价降序排列。

```
select p_partkey, ps_supplycost, avg(p_retailprice) as avgretailprice
from part, partsupp
where part.p_partkey = partsupp.ps_partkey
and part.p_size between 7 and 14
group by p_partkey, ps_supplycost
order by avgretailprice desc;
```

嵌套查询

查询 14：从订单明细表 LINEITEM、订单表 ORDERS、客户表 CUSTOMER 中，使用 set membership 运算符 in，查询明细折扣小于 0.01 的订单，列出这些订单的 key 和采购订单的客户姓名。

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。

(1)嵌套查询

```
select o_orderkey, c_name
from orders, customer
where o_custkey = c_custkey
and o_orderkey in(
    select l_orderkey
    from lineitem
    where l_discount < 0.01);
```

(2)多表查询

```
select distinct l_orderkey, c_name
from orders, customer, lineitem
where o_orderkey = l_orderkey
and c_custkey = o_custkey
and l_discount < 0.01;
```

查询 15-1：从订单明细表 LINEITEM，使用 Set Comparison 运算符 some，查询满足下列条件的订单：该订单的数量大于发货日期在[?,?]之间的部分（至少一个）订单的数量，列出这些订单的流水号、key 和税。

```
select l_orderkey, l_linenumber, l_tax
from lineitem
where l_quantity > some (
```

```
    select l_quantity
    from lineitem
    where l_shipdate > '2015-01-01'::date
    and l_shipdate < '2015-02-02'::date
);
```

查询 15-2：从订单表 ORDERS，使用 Set Comparison 运算符 some，查询满足下列条件的订单：订单状态为‘O’，订单总价大于部分在 2020 年之后下单的订单。列出这些订单的 key、客户 key、收银员。

```
select o_orderkey, o_custkey, o_clerk
from orders
where o_totalprice > some (
    select o_totalprice
    from orders
    where o_orderstatus = 'O'
    and o_orderdate >= '2020-01-01'::date
);
```

查询 16-1：从订单明细表 LINEITEM 中，使用 Set Comparison 运算符>=all，查询满足下列条件的供应商：该供应商在 2019 年出货量大于等于同时段其他供应商的出货量，即 2019 年该供应商的出货量最高。

```
select l_suppkey
from lineitem
where l_shipdate >= '2019-01-01'::date
and l_shipdate <= '2019-12-30'::date
group by l_suppkey
having sum(l_quantity) >= all(
    select sum(l_quantity)
    from lineitem
    where l_shipdate >= '2019-01-01'::date
    and l_shipdate <= '2019-12-30'::date
    group by l_suppkey
);
```

查询 16-2：供应商表 SUPPLIER，使用 Set Comparison 运算符 all，查询账户余额大于等于其他供应商的供应商。列出该供应商的姓名、key、手机号。

```
select s_suppkey, s_name, s_phone
```

```
from supplier
where s_acctbal >= all(
    select s_acctbal
    from supplier
);
```

查询 17-1：从供应商表 SUPPLIER、国家表 NATION，使用 Test for Empty Relations 运算符“exists”，查询国家为日本，账户余额大于 5000 的供应商。

```
select s_suppkey
from supplier
where exists(
    select *
    from nation
    where n_nationkey = s_nationkey
    and n_name = 'JAPAN'
    and s_acctbal > 5000
);
```

查询 17-2：从客户表 CUSTOMER、国家表 NATION、订单表 ORDERS、订单明细表 LINEITEM、供应商表 SUPPLIER 中，使用 Test for Empty Relations 运算符“not exists except”，查询满足下列条件的供应商：该供应商不能供应所有的零件。

```
select s_suppkey
from supplier
where not exists(
    select ps_partkey
    from partsupp
    where ps_suppkey = s_suppkey
    except(
        select p_partkey
        from part
    )
);
```

查询 18：从国家表 NATION、客户表 CUSTOMER 中，使用“count”，查询满足下列条件的国家：至少有 3 个客户来自这个国家，并列出该国家的国家 key 和国家名。

```
select n_nationkey, n_name
from nation, customer
where n_nationkey = c_nationkey
group by n_nationkey
having count(c_custkey) >= 3;
```

查询 19：从零部件表 PART 和零部件供应表 PARTSUPP 中，使用 Subqueries in the From Clause 方法，查询满足下列条件的零件：零件由 2 个以上的供应商供应，且零件大小在 20 以上。

```
select ps_partkey
from (
    select ps_partkey, p_size
    from part, partsupp
    where p_partkey = ps_partkey
    group by ps_partkey, p_size
    having count(ps_suppkey) > 2
) where p_size > 20;
```

with 临时视图查询

查询 20：用 with 临时视图方式，实现查询 19 中查询要求。

```
with temp as(
    select ps_partkey, p_size
    from part, partsupp
    where p_partkey = ps_partkey
    group by ps_partkey, p_size
    having count(ps_suppkey) > 2)
select ps_partkey
from temp
where p_size > 20;
```

查询 21：从零部件供应表 PARTSUPP 中，用 with 临时视图方式，查询零件供应数量最多的供应商 key 和其供应的数量。

```
with supnum as(
    select s_suppkey, max(ps_availqty) as quantity
    from supplier, part, partsupp
    where s_suppkey = ps_suppkey
```

```
    and p_partkey = ps_partkey
    group by s_suppkey
)
select *
from supnum;
```

键/函数依赖分析

查询 22：在订单明细表 LINEITEM 中，检查订单 key、零件 key、供应商 key、流水号是否组成超键。

```
select l_orderkey, l_partkey, l_suppkey, l_linenumber
from lineitem
group by l_orderkey, l_partkey, l_suppkey, l_linenumber
having count(*) > 1;
```

若结果为空，则以上属性组成超键；否则不能组成超键。

查询 23：在订单明细表 LINEITEM 中，利用 SQL 语句检查函数依赖零件 key→价格是否成立；如果不成立，利用 SQL 语句找出导致函数依赖不成立的元组。

```
select l_partkey
from lineitem
group by l_partkey
having count(distinct l_extendedprice) > 1;
```

若结果为空，则函数依赖成立；反之不成立，

若不成立，找出导致不成立的元组：

```
select *
from lineitem a
where a.l_partkey in(
    select b.l_partkey
    from lineitem b
    group by b.l_partkey
    having count(distinct l_extendedprice) > 1);
```

关系表的插入/删除/更新

查询 24：向订单表 ORDERS 中插入一条订单数据；

```
insert into orders values('1200001', '20045', 'F', 61365.24, '2017-03-19'::date, '2-HIGH',
'Clerk#000000098', 0, 'furiously special f');
```

查询 25：将零件 32 的全部供应商，作为零件 20 的供应商，加入到零部件供应表 PARTSUPP 中。

```
insert into partsupp(
    select '20', ps_suppkey, ps_availqty, ps_supplycost, ps_comment
    from partsupp
    where ps_partkey = '32'
    and ps_suppkey not in(
        select ps_suppkey
        from partsupp
        where ps_partkey = '20'
    )
);
```

查询 26：在订单明细表 LINEITEM 中，删除已退货的订单记录。(returnflag='R')

```
delete from lineitem
where l_returnflag = 'R';
```

查询 27：用订单明细表 LINEITEM 中在 2019 年之后交易中的预计到达日期，替换表中的实际到达日期。

```
update lineitem
set l_receiptdate = l_commitdate
from orders
where l_orderkey = o_orderkey
and o_orderdate >= '2020-01-01'::date;
```

查询 28：针对订单明细表 LINEITEM、订单表 ORDERS，使用 update/case 语句做出如下修改：如果订单的订单优先级低于 medium，则其在订单明细表中的预计到达日期推后 2 天，否则推迟一天。

```
update lineitem
set l_commitdate =
case when l_orderkey in (
    select o_orderkey
    from orders
    where o_orderpriority = '3-MEDIUM'
)
```

```
then l_commitdate + '2 day'
else
    l_commitdate + '1 day'
end;
```

查询 29：在订单表 ORDERS 中，按照订单总价对订单进行降序排序，并输出订单 key 和排名。

```
select O_ORDERKEY ,O_TOTALPRICE,rank() over(order by O_TOTALPRICE desc) as  'Rank'
from ORDERS;
```