



数据库系统课程实验报告

| | |
|-------|--|
| 实验名称: | 实 验 九 ： 使 用 JDBC 连 接 openGauss 数据库 |
| 实验日期: | 2024.5.24--2024.5.31 |
| 实验地点: | 文宣楼 B313 |
| 提交日期: | 2024.6.3 |

| | |
|-------|------------------|
| 学号: | 37220222203791 |
| 姓名: | 魏一诺 |
| 专业年级: | 软工 2022 级 |
| 学年学期: | 2023-2024 学年第二学期 |

1.实验目的

- 理解并掌握 openGauss 的查询处理过程
- 理解并掌握 openGauss 的查询优化过程
- 掌握使用 EXPLAIN 命令查看执行计划的方法
- 理解 EXPLAIN 输出结果的含义，包括：全表扫描、索引扫描、位图扫描、条件过滤、嵌套循环、

2.实验内容和步骤

阅读并实现《PostgreSQL 修炼之道-从小工到专家》材料中的第 9 章之 9.1

1) 连接到数据库

```
cd /opt/software/openGauss/script/
//步骤 1 在数据库主节点服务器上，切换至 omm 操作系统用户环境
su - omm
gs_om -t status    //步骤 2 查看服务是否启动
gs_om -t start     //步骤 3 启动数据库服务
gsql -d sales -p 26000 -U wyn -W Bigdata@123 -r //连接到数据库
SET search_path TO sales;
```

```
Successfully started.
[omm@ecs-ald4 ~]$ gsql -d sales -p 26000 -U wyn -W Bigdata@123 -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sales=> █
```

2)

```
EXPLAIN SELECT * FROM employee;
```

```
sales=> EXPLAIN SELECT * FROM employees;
               QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.07 rows=107 width=78)
(1 row)
```

使用 EXPLAIN 命令可以显示一个查询的执行计划，但不实际执行该查询

3)

```
EXPLAIN ANALYZE SELECT * FROM employees;
```

ANALYZE 选项会实际执行 SQL 并提供实际的运行时间和行数。

```
sales=> EXPLAIN ANALYZE SELECT * FROM employees;
               QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.07 rows=107 width=78) (actual time=0.884..0.952 rows=107 loops=1)
Total runtime: 0.995 ms
(2 rows)
```

4)

```
EXPLAIN (VERBOSE) SELECT * FROM employees;
```

VERBOSE 选项提供额外的详细信息。

```
sales=> EXPLAIN (VERBOSE) SELECT * FROM employees;
               QUERY PLAN
-----
Seq Scan on sales.employees (cost=0.00..5.07 rows=107 width=78)
Output: employee_id, first_name, last_name, email, phone, hire_date, manager_id, job_title
(2 rows)
```

5)

```
EXPLAIN (COSTS) SELECT * FROM employees;
```

使用 COSTS 选项可以显示每个计划节点的成本。

```
sales=> EXPLAIN (COSTS) SELECT * FROM employees;
               QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.07 rows=107 width=78)
(1 row)
```

6)

```
EXPLAIN (ANALYZE, BUFFERS) SELECT * FROM employees;
```

BUFFERS 选项与 ANALYZE 一起使用，显示缓冲区的使用情况

```
sales=> EXPLAIN (ANALYZE, BUFFERS) SELECT * FROM employees;
               QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.07 rows=107 width=78) (actual time=0.009..0.050 rows=107 loops=1)
  (Buffers: shared hit=4)
Total runtime: 0.101 ms
(3 rows)
```

7)

```
EXPLAIN (FORMAT TEXT) SELECT * FROM employees;
```

FORMAT 选项允许将输出格式化为 TEXT、XML、JSON 或 YAML。

```
sales=> EXPLAIN (FORMAT TEXT) SELECT * FROM employees;
          QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.07 rows=107 width=78)
(1 row)
```

8)

```
EXPLAIN (ANALYZE, BUFFERS, VERBOSE) SELECT * FROM employees;
```

可以结合使用多个选项来获取更详细的执行计划。

```
sales=> EXPLAIN (ANALYZE, BUFFERS, VERBOSE) SELECT * FROM employees;
          QUERY PLAN
-----
Seq Scan on sales.employees (cost=0.00..5.07 rows=107 width=78) (actual time=0.008..0.051 rows=107 loops=1)
  Output: employee_id, first_name, last_name, email, phone, hire_date, manager_id, job_title
  (Buffers: shared hit=4)
Total runtime: 0.103 ms
(4 rows)
```

9) 查看特定类型操作的执行计划：

例如，查看全表扫描 (Seq Scan)、索引扫描 (Index Scan) 或位图扫描 (Bitmap Scan)。

```
EXPLAIN SELECT * FROM employees WHERE employee_id<10;
```

```
sales=> EXPLAIN SELECT * FROM employees WHERE employee_id<10;
          QUERY PLAN
-----
Seq Scan on employees (cost=0.00..5.34 rows=10 width=78)
  Filter: (employee_id < 10)
(2 rows)
```

10) 查看不同类型的 JOIN 操作，如嵌套循环连接 (Nestloop Join)、哈希连接 (Hash Join) 和合并连接 (Merge Join)。

```
EXPLAIN SELECT * FROM inventories JOIN warehouses ON inventories.WAREHOUSE_ID
=warehouses.WAREHOUSE_ID;
```

```
sales=> EXPLAIN SELECT * FROM inventories JOIN warehouses ON inventories.WAREHOUSE_ID =warehouses.WAREHOUSE_ID;
      QUERY PLAN
-----
Hash Join (cost=13.31..78.39 rows=2224 width=536)
  Hash Cond: (inventories.warehouse_id = warehouses.warehouse_id)
    -> Seq Scan on inventories (cost=0.00..35.24 rows=2224 width=12)
    -> Hash (cost=11.47..11.47 rows=147 width=524)
          -> Seq Scan on warehouses (cost=0.00..11.47 rows=147 width=524)
(5 rows)
```

3 实验总结

3.1 完成的工作

进行了一系列 openGauss 的查询操作，使用 EXPLAIN 命令查看执行计划和各类信息

3.2 对实验的认识

在上手实践中对 openGauss 的连接、查询等操作更加熟悉，学习了 EXPLAIN 命令的相关知识。

3.3 遇到的困难及解决方法

EXPLAIN (FORMAT TEXT) SELECT * FROM employees;一开始写的是 json 格式，但是数据库不支持它，换成 text 格式就好了。