|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 《数据库原理》实验报告 | | | | |
| 题目：实验一  数据库和表的创建与管理 | 学号 | 姓名 | 班级 | 日期 |
| **2021302111** | **禹泽海** | **JKC012101** | **2023/10/5** |

**一 .头歌平台截图**



头歌平台实验二完成截图

**二 .实验内容、步骤以及结果（线下实验）**

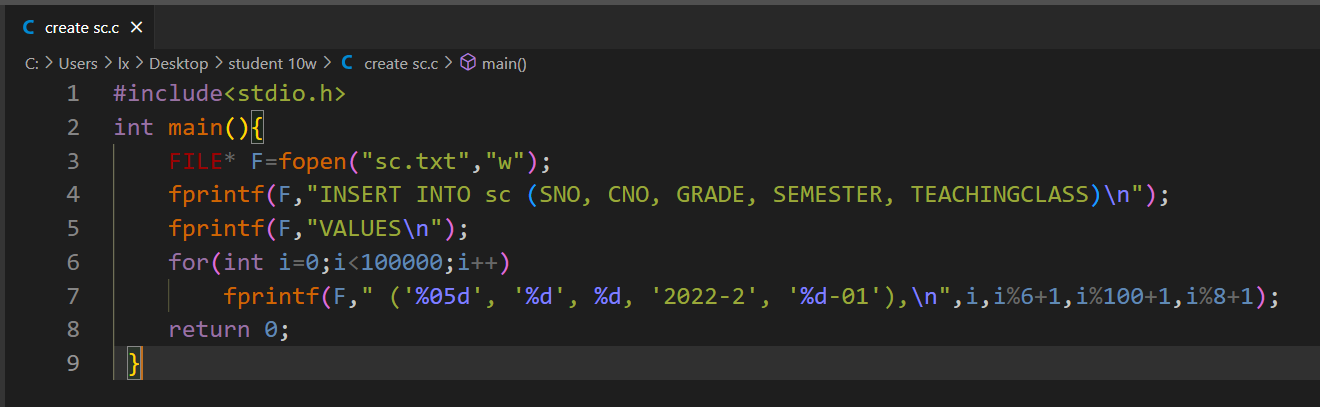
必做实验 实验内容：

针对Student数据库用至少三种不同的**SQL语句**进行查询：查询选修了课程名为“数据库”的学生学号和姓名，然后自己设计实验，用数据比较分析三种查询的效率，并分析原因。

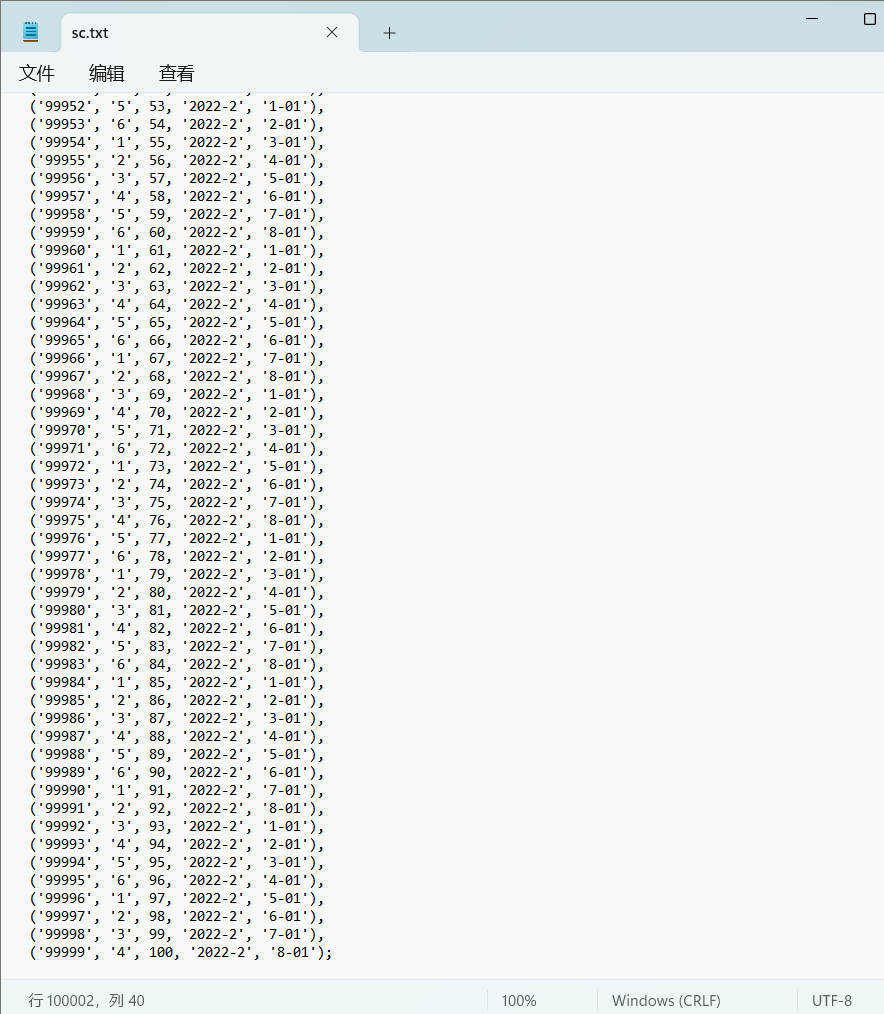
说明：查询性能验证最好在10W以上的数据条件下进行测试。构造10W以上的数据可以自行编程插入数据或者用导入导出方式实现。

实验步骤：**必做实验**

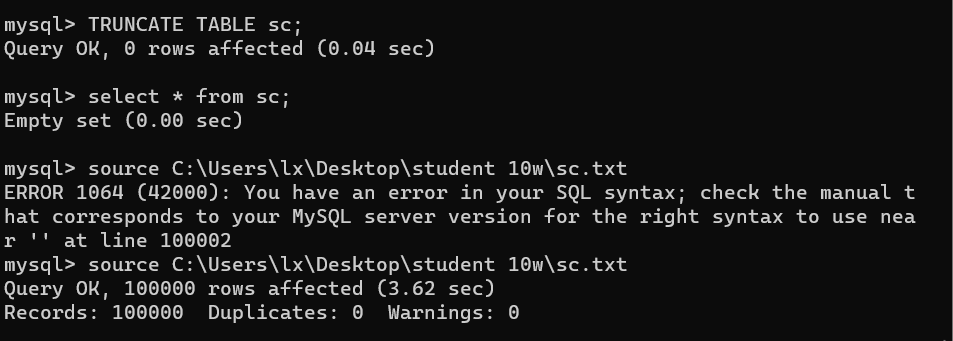
首先使用c语言编写生成表数据txt的代码，核心在于规定数据范围和格式；



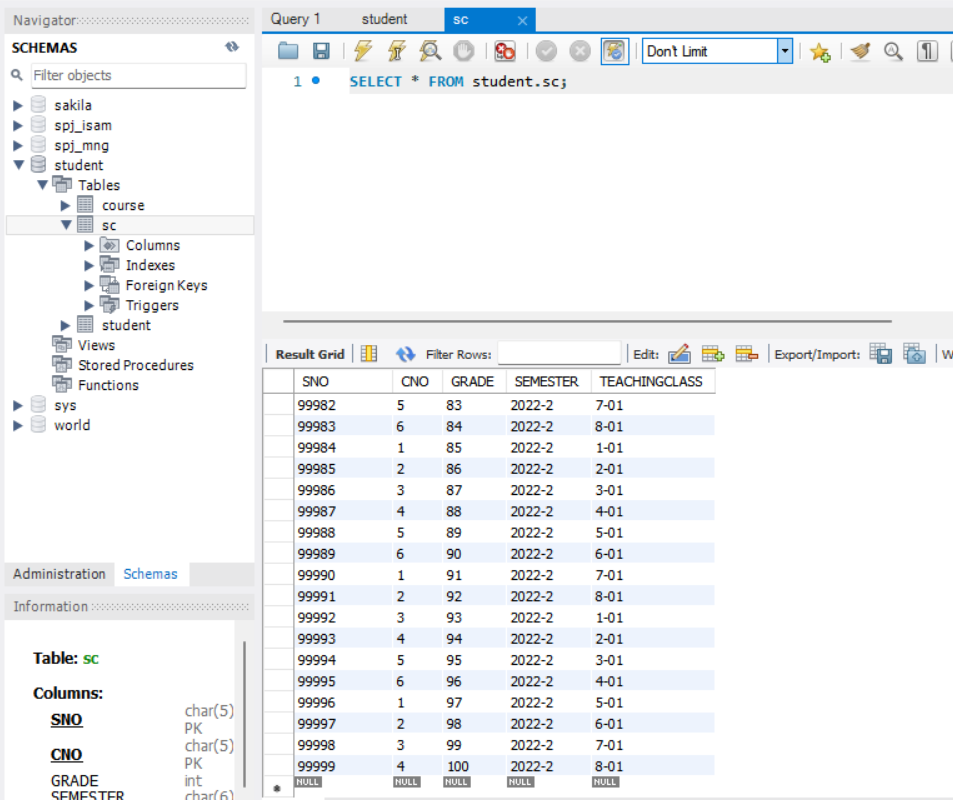
生成的txt文件如下，值得注意的是生成数据最后一条是逗号结尾，应该改为分号即可运行；



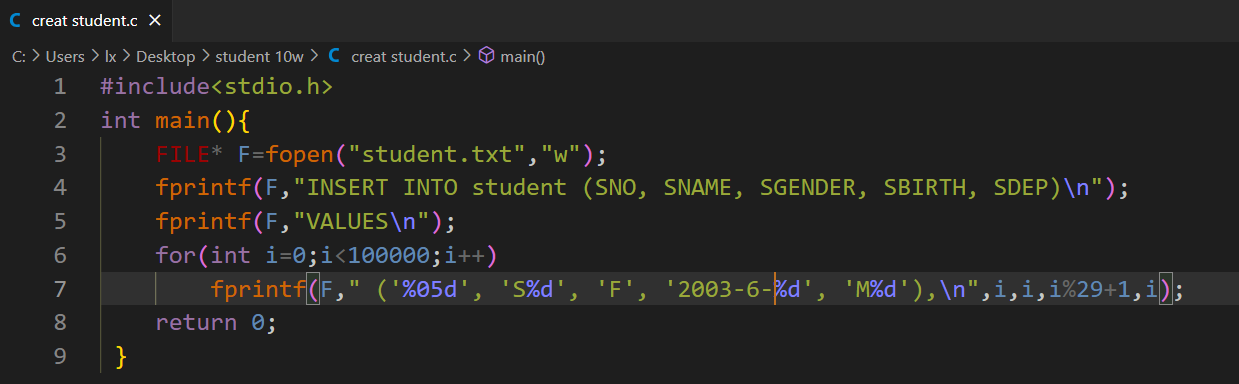
接下来将sc表清空并导入txt文件：



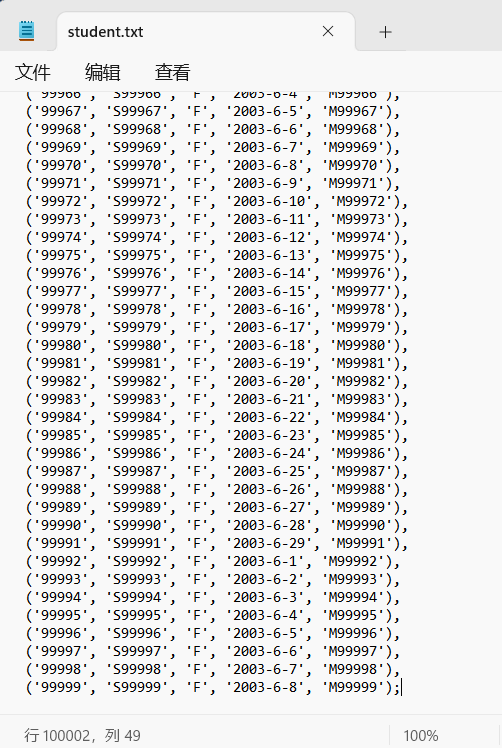
本地查看数据库，数据确实加到了十万条：



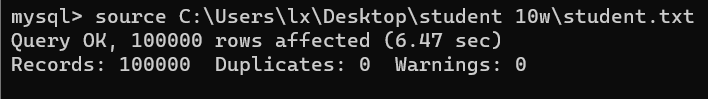
效仿sc的添加过程现在进行student表的数据添加：

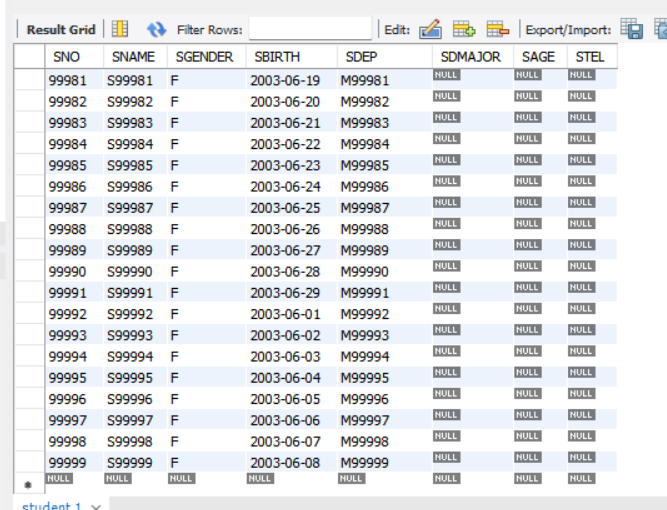


运行生成txt文件：



导入数据库：



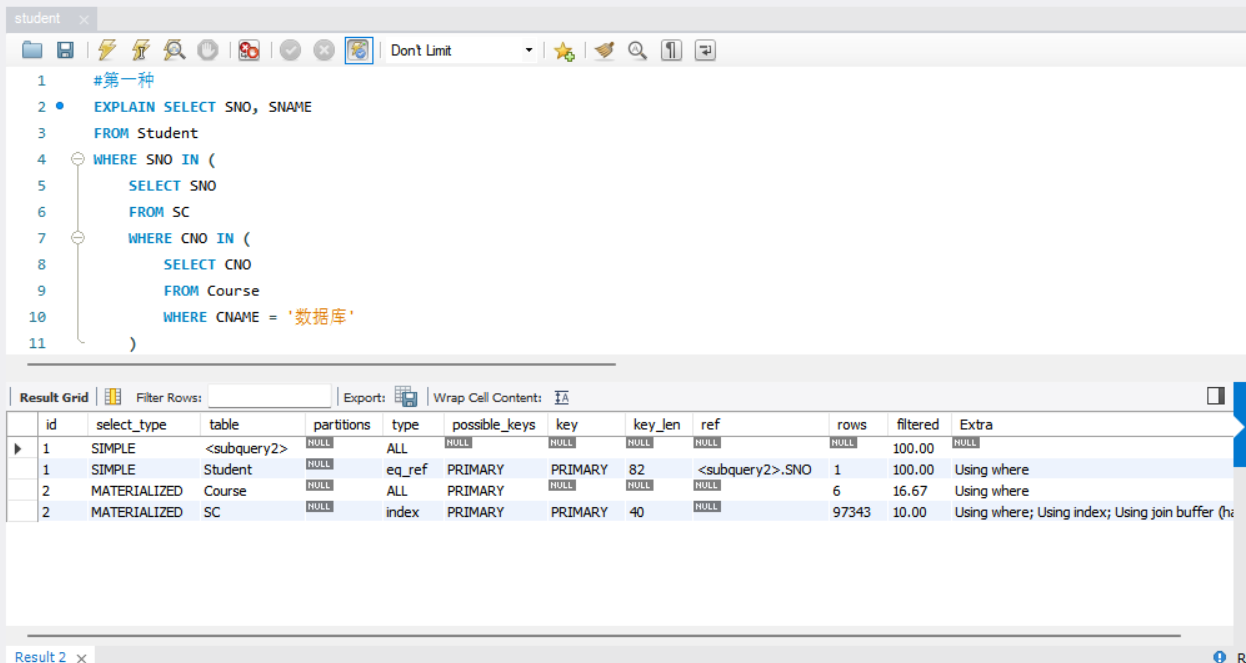
同样前往查看是否加载到本地，直接命令行显示过于浪费资源：  


可以看到十万条数据都已插入完成。

接下来开始三种查询：

题目规定为利用三种不同方式查询选择数据库课程的学生的学号和姓名；

接下来使用三种方法运行查看查询效率。



select\_type：

第一个子查询的 select\_type 为 SIMPLE，表示简单查询。

第二个子查询的 select\_type 为 MATERIALIZED，表示使用临时表存储结果。

table：

第一个子查询涉及 <subquery2>，表示第二个子查询。

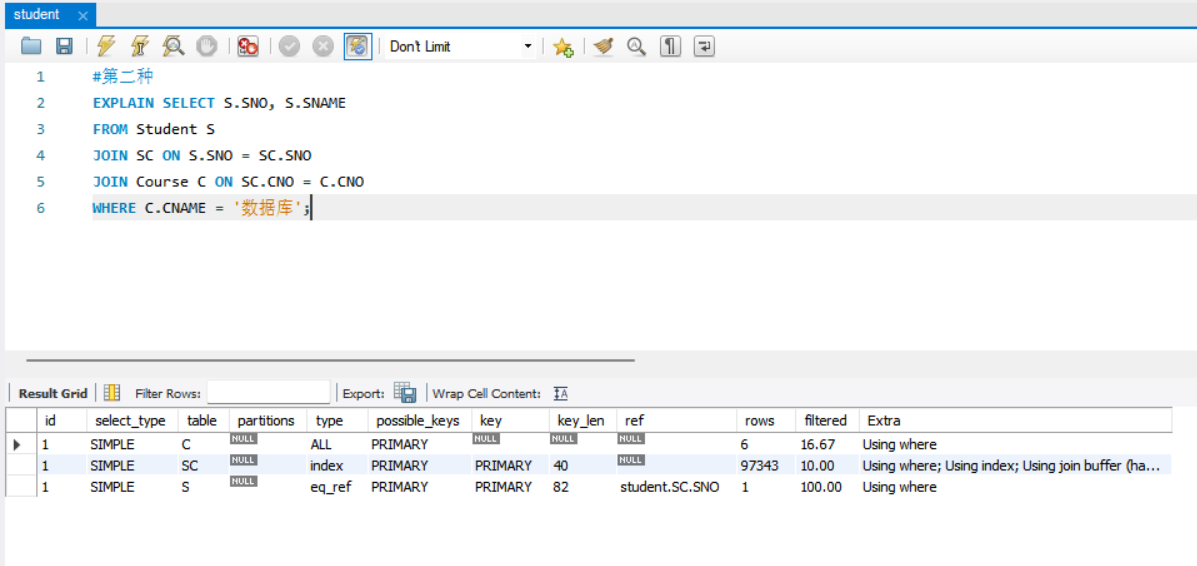
第二个子查询涉及 Course 表和 SC 表。

type：

第一个子查询的 type 为 ALL，表示全表扫描。

第二个子查询的 type 为 index，表示使用索引。

从 EXPLAIN 表可以看出，第一种查询方法使用了多层子查询，而且多次进行全表扫描，性能较低。



select\_type：

三个表的 select\_type 都为 SIMPLE，表示简单查询。

table：

查询涉及 Student、SC 和 Course 表。

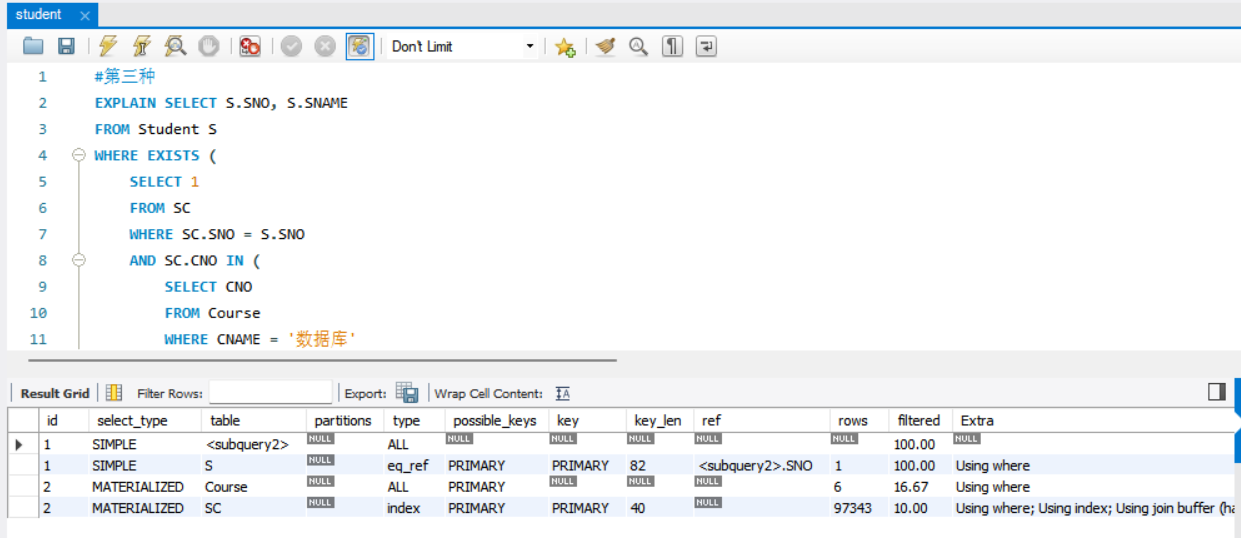
type：

对于 Course 表，type 为 ALL，表示全表扫描。

对于 SC 表，type 为 index，表示使用索引。

对于 Student 表，type 为 eq\_ref，表示使用索引。

从 EXPLAIN 表可以看出，第二种查询方法使用了 JOIN 连接，避免了多层子查询，提高了查询效率。



select\_type：

第一个子查询的 select\_type 为 SIMPLE，表示简单查询。

第二个子查询的 select\_type 为 MATERIALIZED，表示使用临时表存储结果。

table：

第一个子查询涉及 <subquery2>，表示第二个子查询。

第二个子查询涉及 Course 表和 SC 表。

type：

第一个子查询的 type 为 ALL，表示全表扫描。

第二个子查询的 type 为 index，表示使用索引。

从 EXPLAIN 表可以看出，第三种查询方法与第一种类似，仍然使用了多层子查询，性能较低。

**选做实验：**

TPC-H是TPC国际组织发布的数据库基准测试基准之一。针对TPC-H数据库按照以下需求自己设计查询查询并写出SQL语句，并载入少量数据进行写验证。该数据库模拟了一个典型企业中：零件，顾客，零件，供应商，产品，订单等数据，数据库定义语句参考tcp-h-ddl.sql文件。

单表查询语句（实现投影和选择操作）；

分组统计查询语句（不带分组过滤条件，带分组过滤条件）；

单表自身的连接查询；

多个表的连接查询语句；

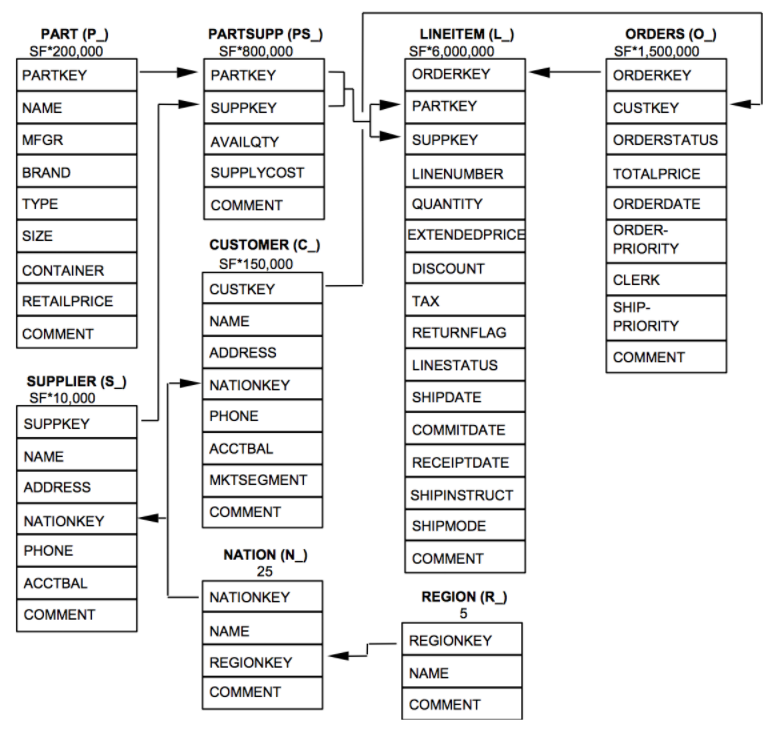
IN嵌套查询；

EXISTS嵌套查询；

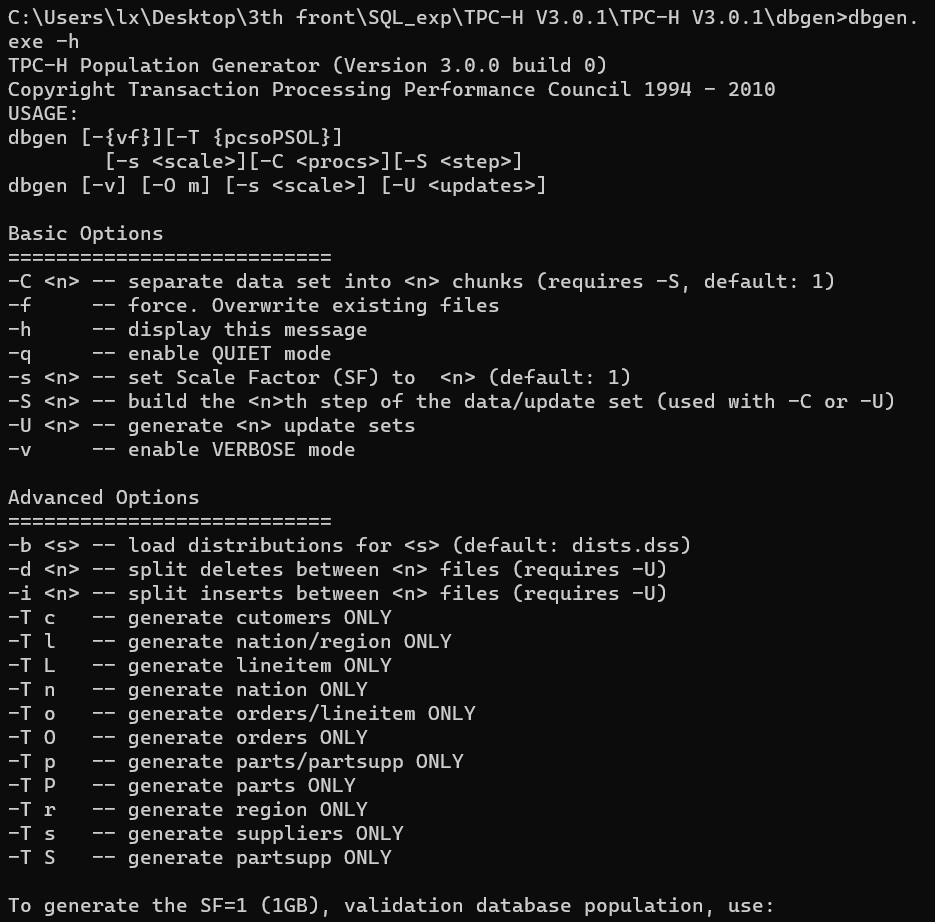
FROM中加入嵌套的查询；

集合查询（交，并，差）。

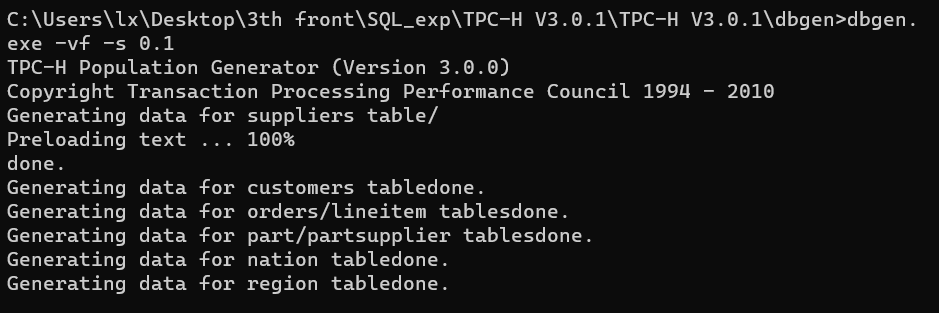
数据库设计图如下：



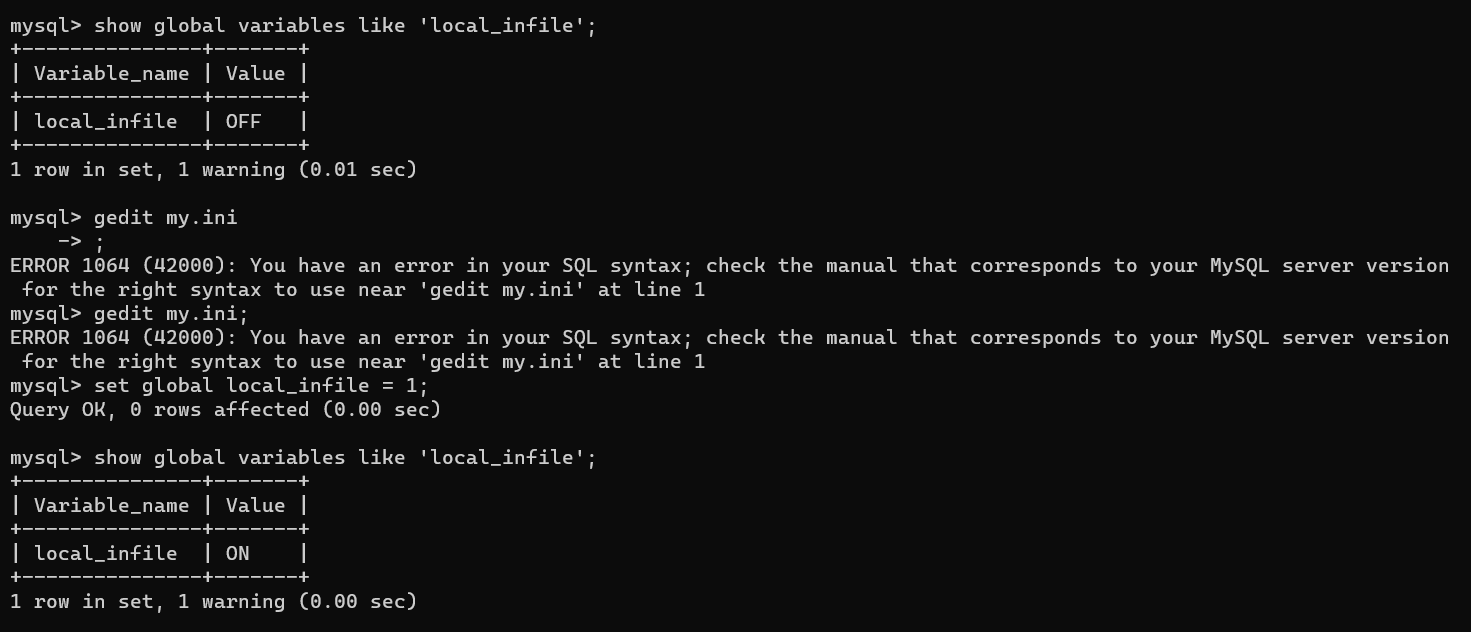
官网下载TPC-H V3.0.1



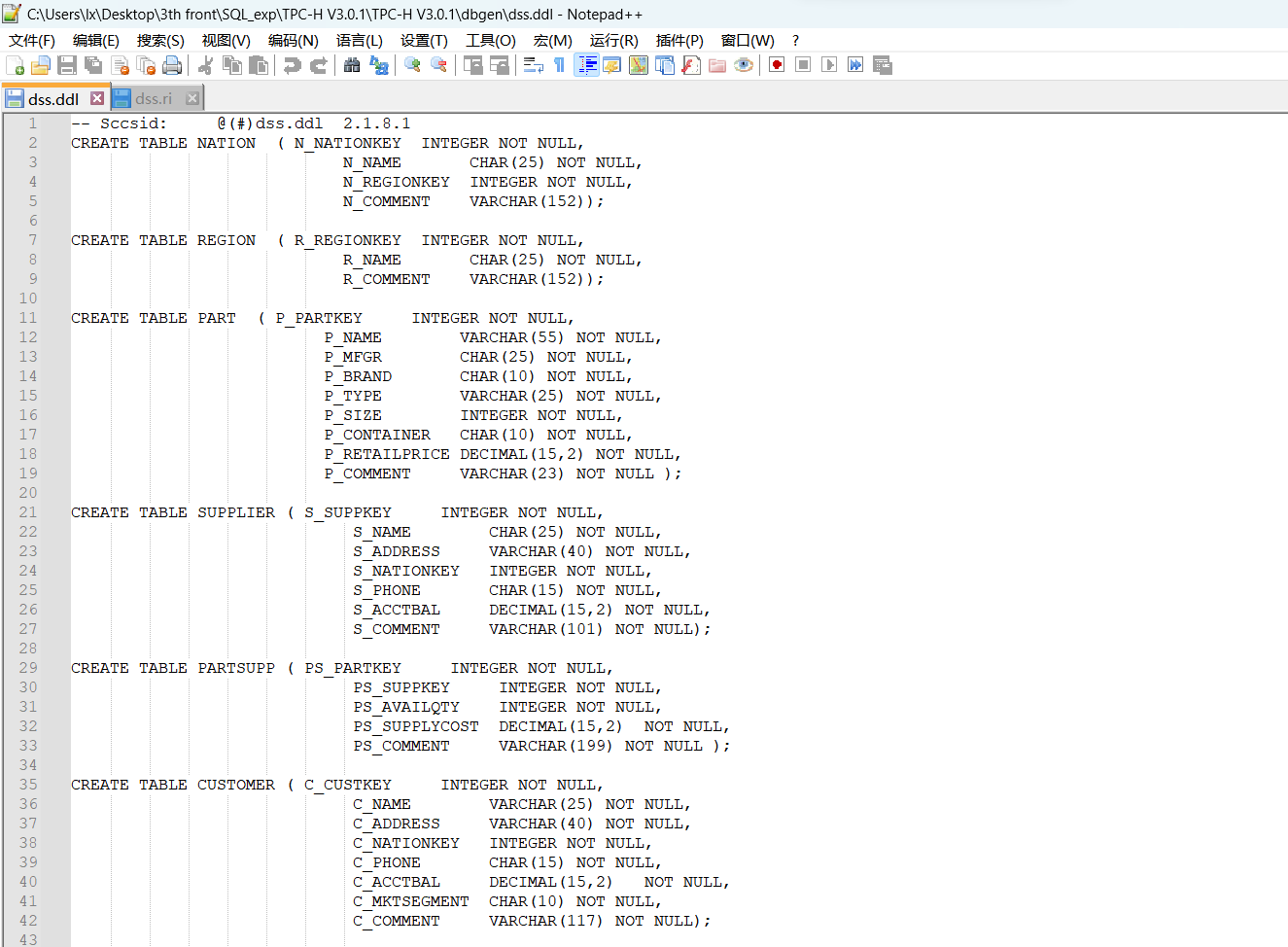
指令加载0.1G的数据



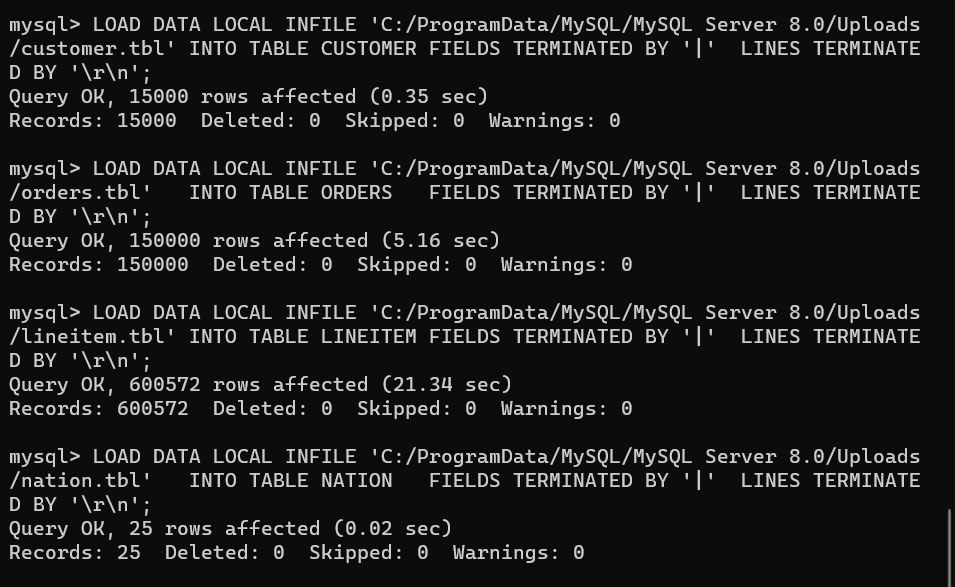
期间遇到error，需要设置local\_infile

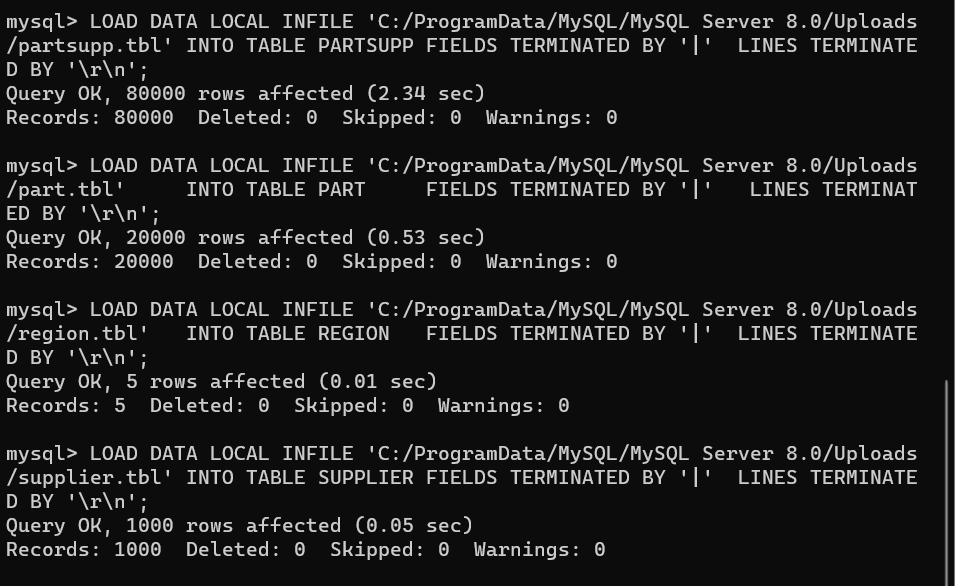


创建database，并添加表

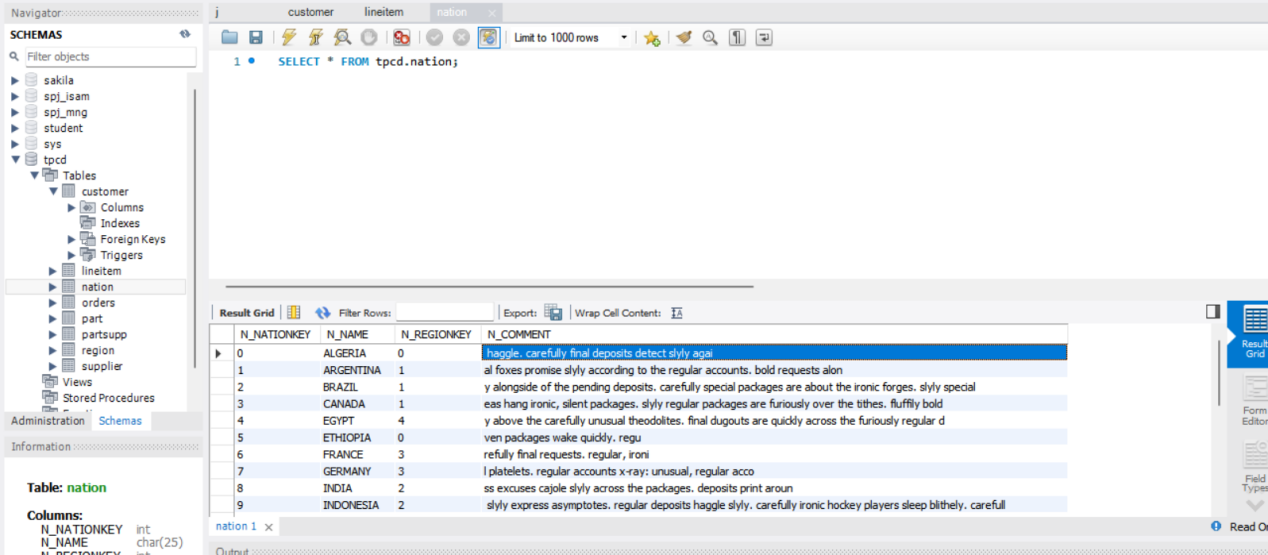


执行后即可添加数据：



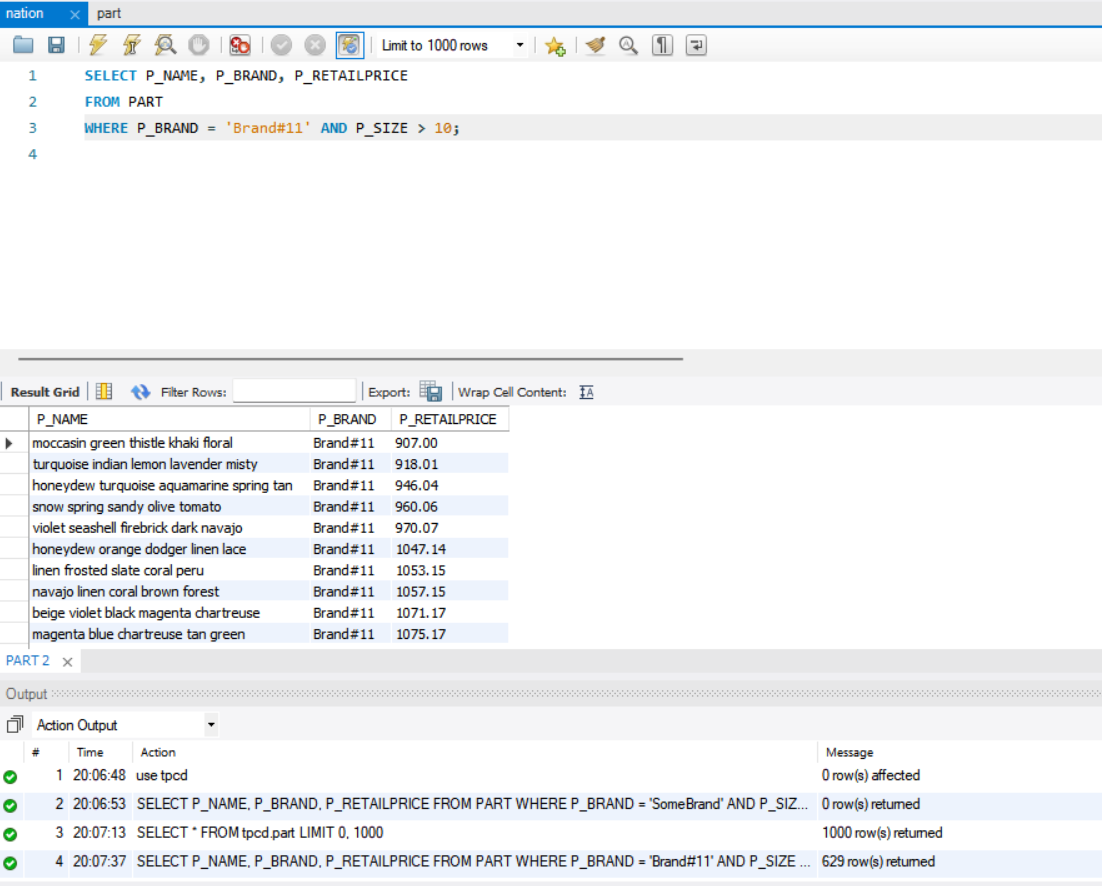


可以查看数据是否添加成功：



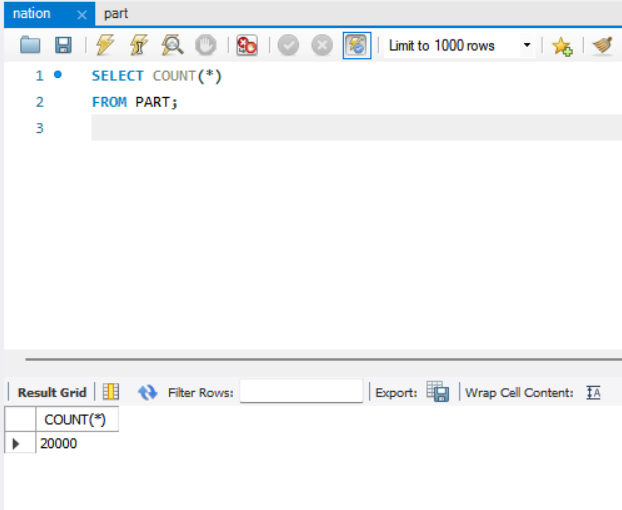
接下来开始题目中的查询：

1 .单表查询语句（实现投影和选择操作，运行结果如下）：

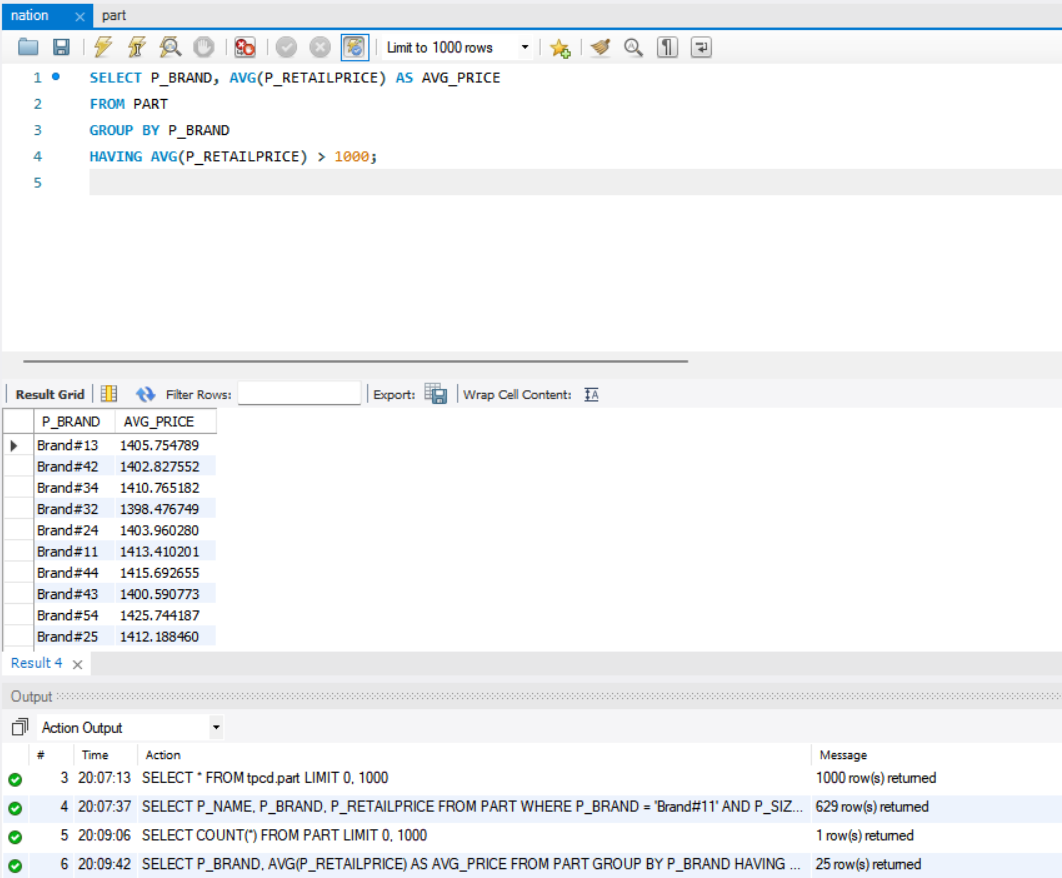


2 .分组统计查询语句（不带分组过滤条件，带分组过滤条件）；

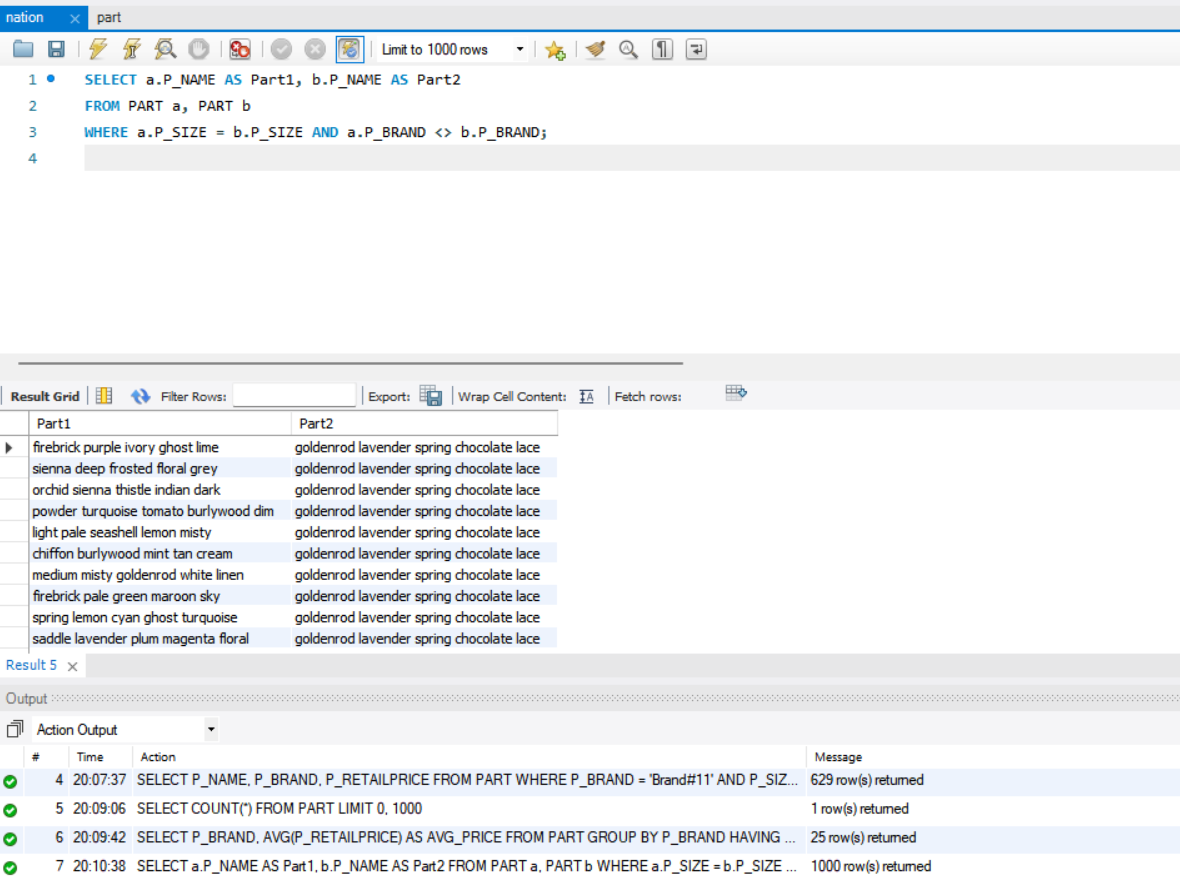
不带过滤分组：



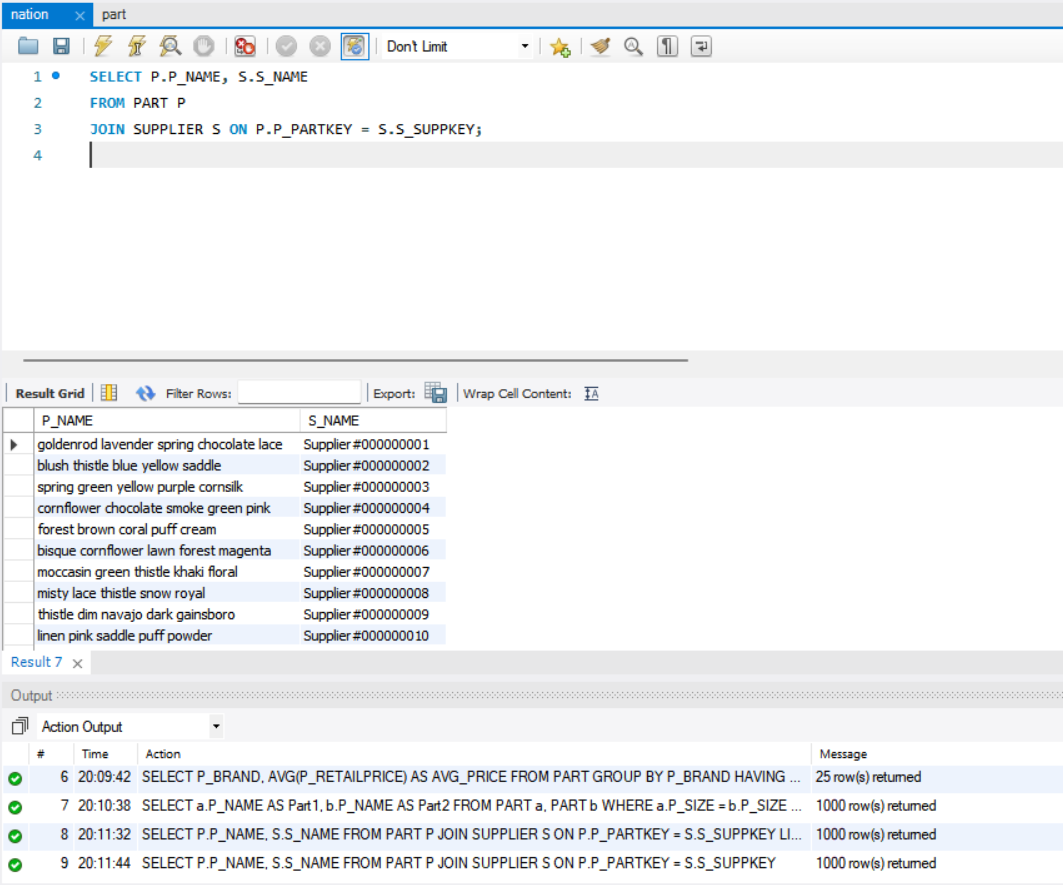
带过滤分组：



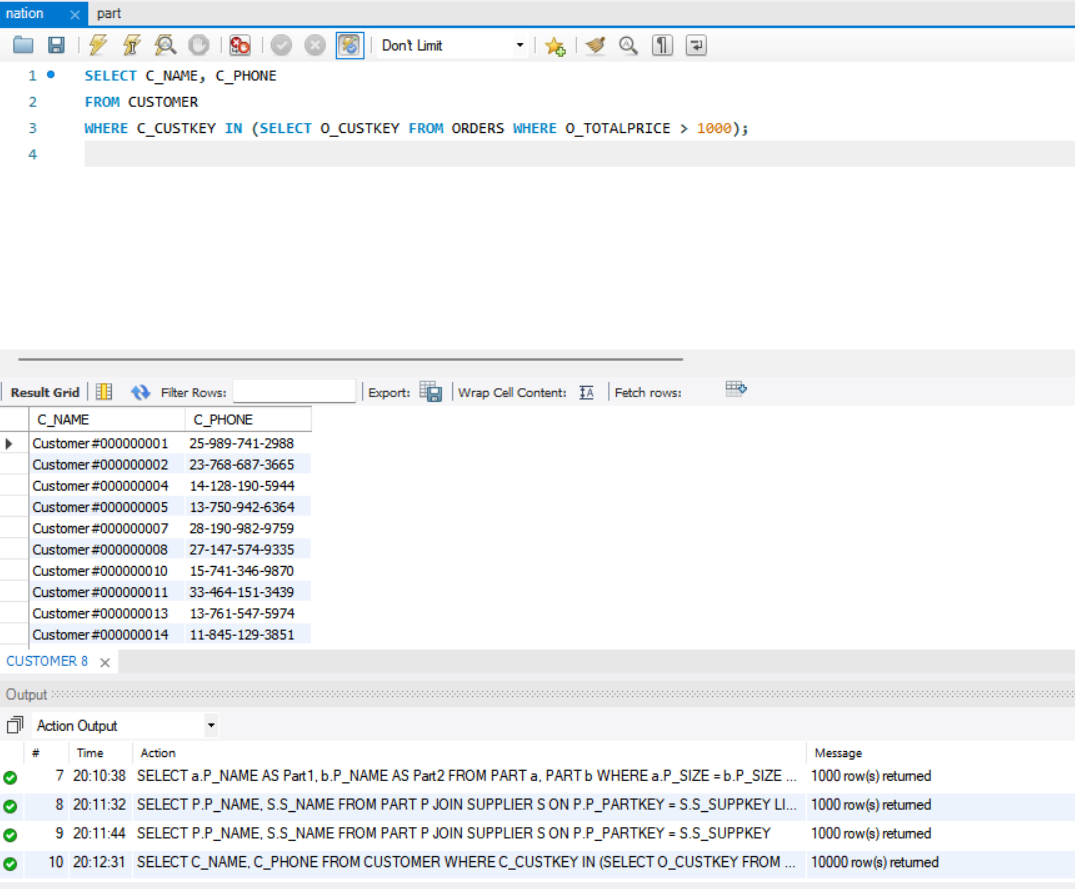
3 .单表自身的连接查询；



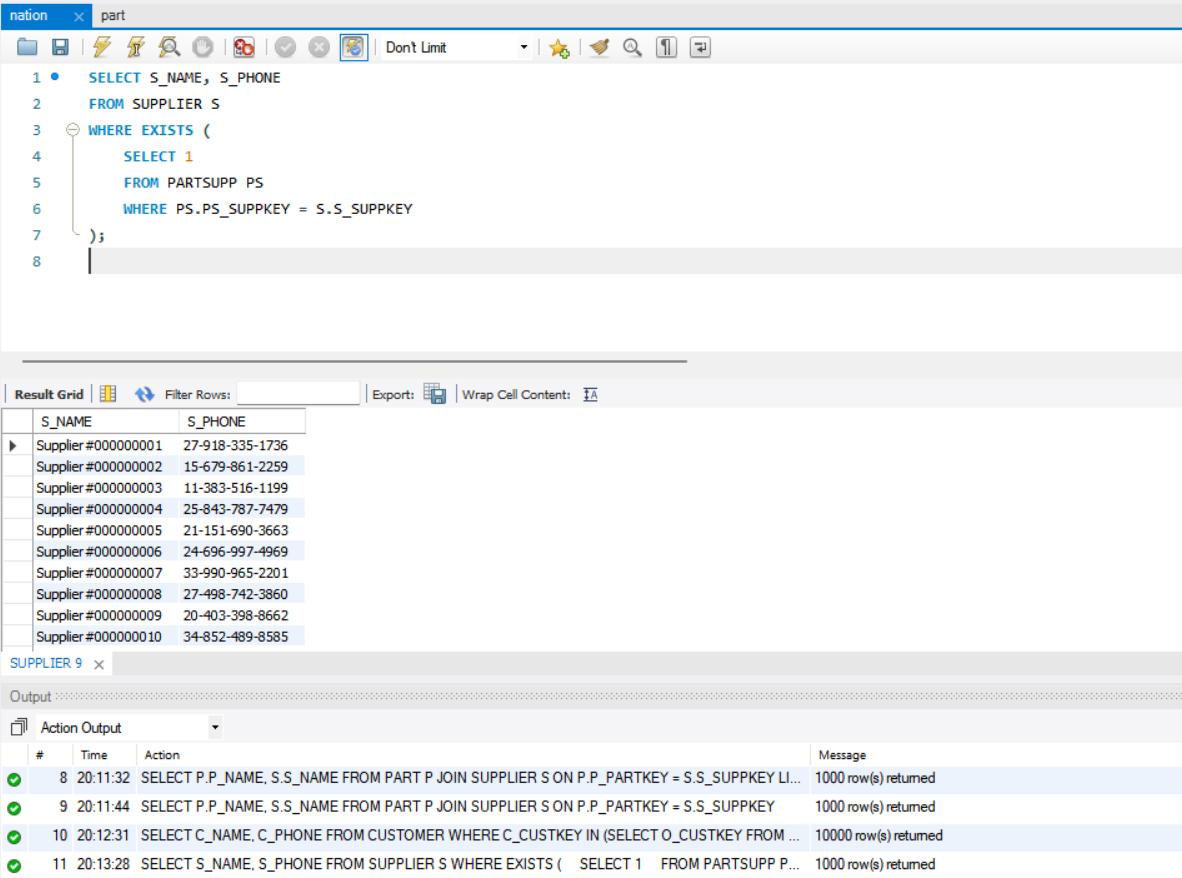
4 .多个表的连接查询语句；



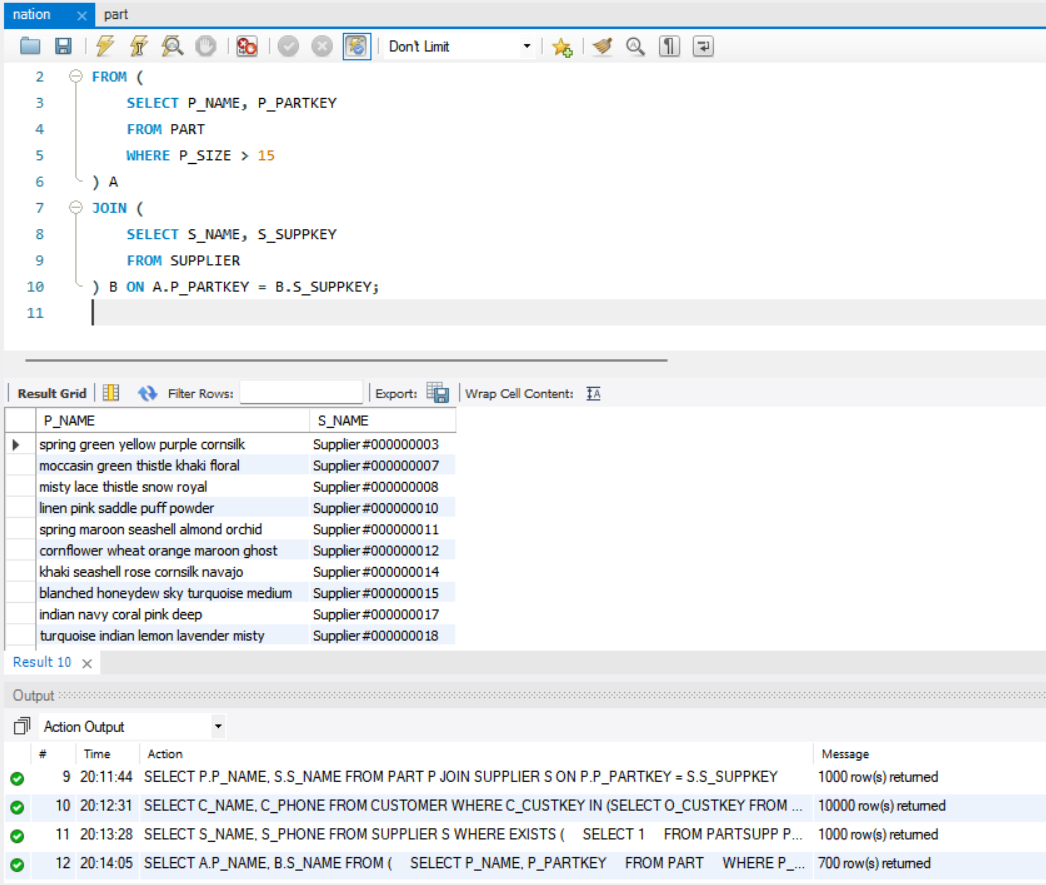
5 .IN嵌套查询



6 .EXIST嵌套查询

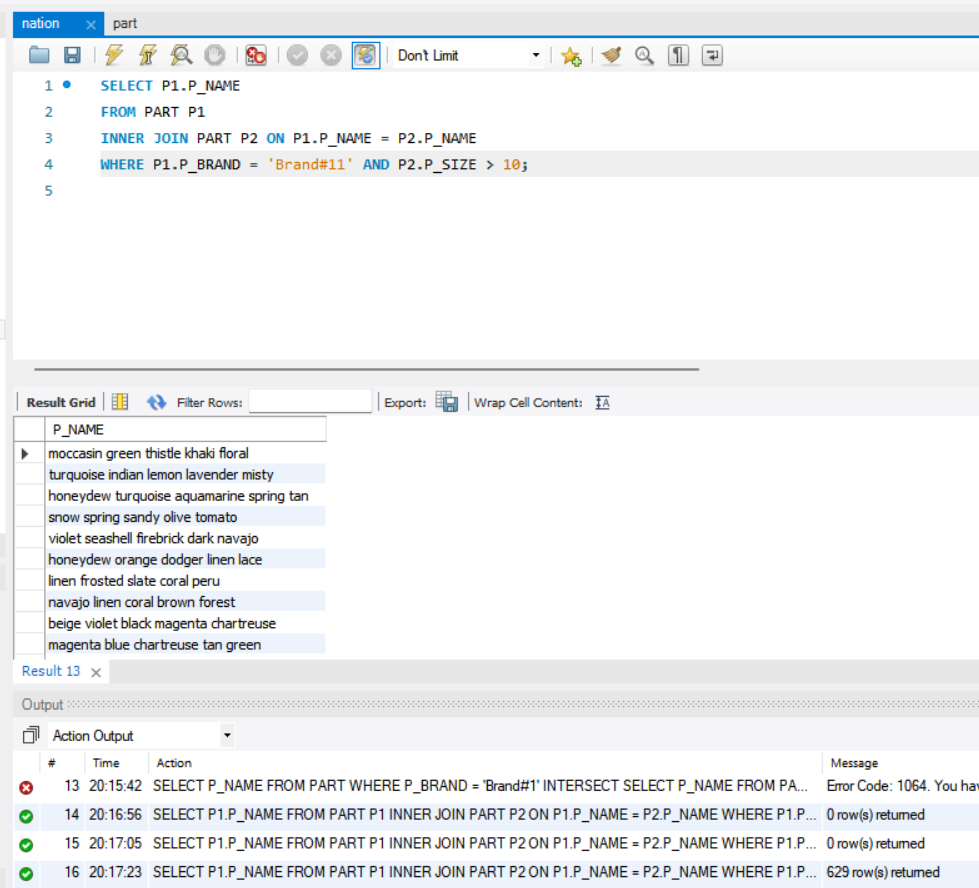


7 .FROM子句中加入嵌套的查询

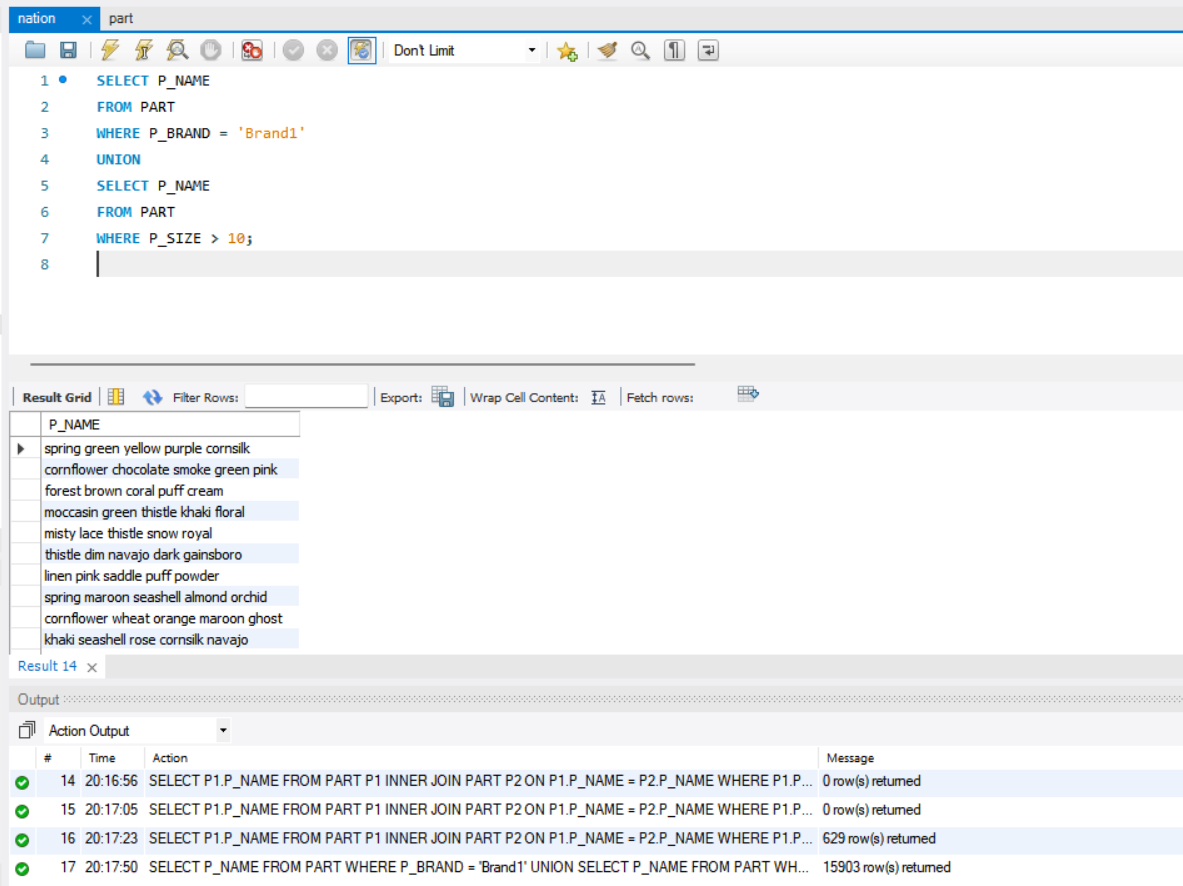


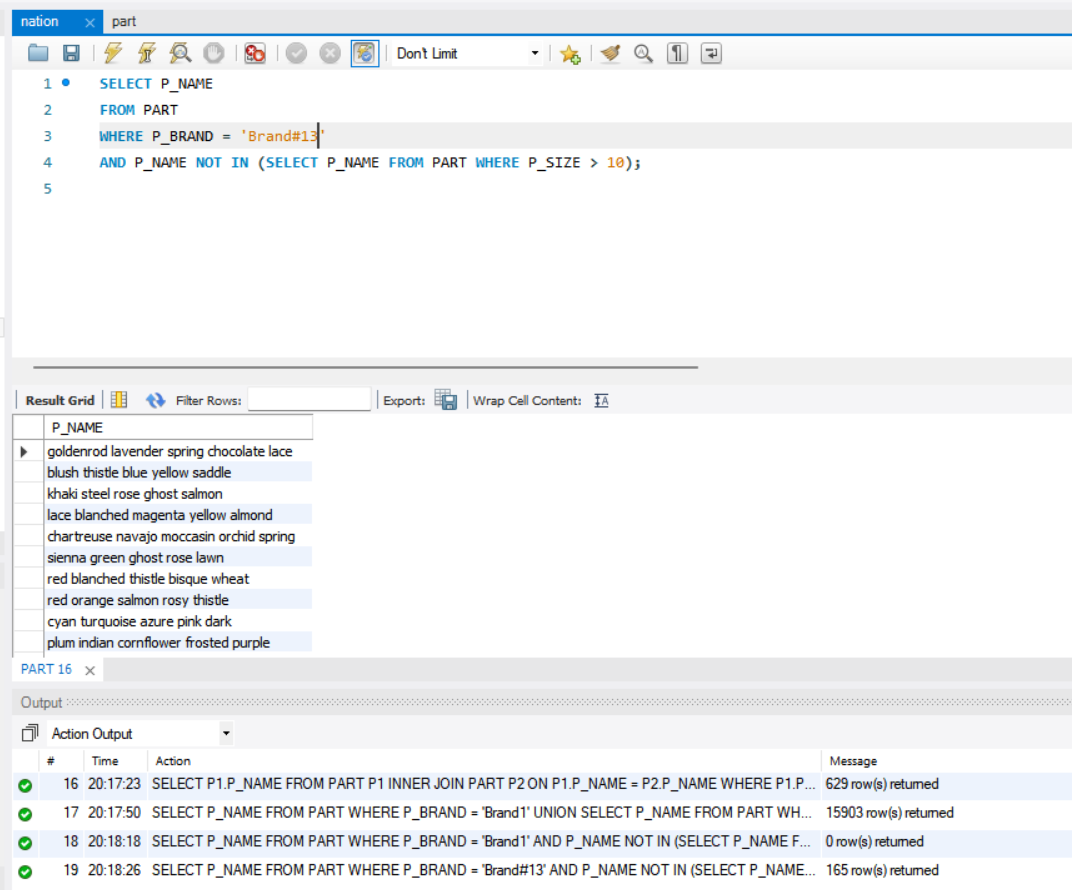
8 .

交集：



并集：



差集：

至此，必做与选做实验全部解决完毕。

**三 .实验中出现的问题以及解决方案**

**问题及解决：此次实验的难点在于10万条数据的载入和业界大型数据库的下载本地，关于10万条数据的载入我使用了c程序生成txt文件再于数据库中运行得到解决。而大型数据库的下载则是借鉴了博文：**

**https://blog.csdn.net/dayuner/article/details/80445772#comments\_20950594**

**感受与建议：数据库的查询需要缜密的逻辑和清晰的思考，在实验中我也深刻地感受到数据库源于生活，可以再这个数据庞杂的时代处理他们并实现结果，是很贴近生活的一门学科。其次，查询中的优化在面对巨量数据时显得格外重要，所以优化和自我反思能力也是我们很有必要去学习培养的。**

**批阅者：**

**批阅日期：**

**实验成绩：**

**批注：**