**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实 验 人：** | **凌国明** | | **学 号：** | **21307077** | **日 期：** | **2023年9月21日** |
| **院（系）：** | **计算机学院** | | | **专业（班级）：** | **计算机科学与技术** | |
| **联系方式：** |  | | |  |  | |
| **实验题目：** | | **1.4 数据更新实验** | | | | |

* + - 1. **实验目的**

熟悉数据库的数据更新操作，能够使用 SQL 语句对数据库进行数据的插入、修改、 删除操作。

* + - 1. **实验内容和要求**

针对 TPC-H 数据库设计单个元组插入、批量数据插入、修改数据和删除数据等 SQL 语句。理解和掌握 insert、update 和 delete 语法结构的各个组成成分，结合嵌套 SQL 子查询，分别设计几种不同形式的插入、修改和删除数据的语句并调试成功。

* + - 1. **实验重点和难点**

实验重点：插入、修改和删除数据的 SQL。

实验难点：与嵌套 SQL 子查询相结合的插入、修改和删除数据的 SQL 语句；利用一 个表的数据来插入、修改和删除另外一个表的数据。

* + - 1. **实验工具**

MySQL、SQL Server、Navicat

* + - 1. **实验过程**

（1）insert 基本语句（插入全部列的数据）

插入一条顾客记录，要求每列都给一个合理的值。如：values(30303030, '张三', '北京市',40,'010-51001199',0.00, 'Northeast', 'VIP Customer')

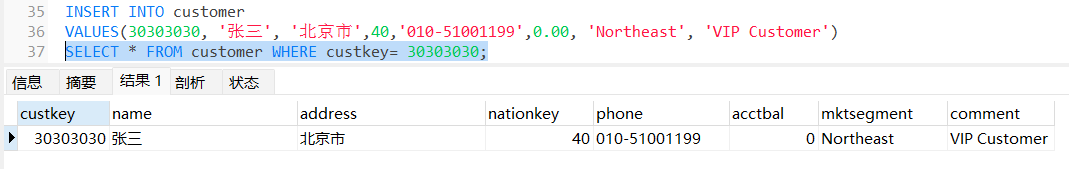
执行以下SQL语句：

INSERT INTO customer

VALUES(30303030, '张三', '北京市',40,'010-51001199',0.00, 'Northeast', 'VIP Customer')

将其结果查询后截屏如下：

SELECT \* FROM customer WHERE custkey= 30303030;



（2）insert 基本语句（插入部分列的数据）

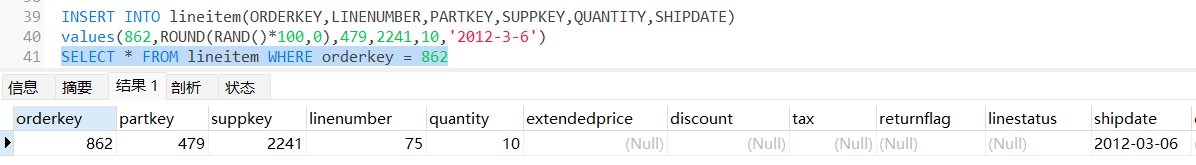
插入一条订单记录，给出必要的几个字段值。如：(ORDERKEY,LINENUMBER,PARTKEY,SUPPKEY,QUANTITY,SHIPDATE)values(862,ROUND(RAND()\*100,0),479,2241,10,'2012-3-6')

执行以下SQL语句：

INSERT INTO lineitem(ORDERKEY,LINENUMBER,PARTKEY,SUPPKEY,QUANTITY,SHIPDATE)

values(862,ROUND(RAND()\*100,0),479,2241,10,'2012-3-6')

将其结果查询后截屏如下：



说明：由于在创建八个基本表的时候，已将所有列添加了约束：非空（不能为 NULL 值），因此在给 lineitem 插入部分列的时候会失败。

（3）批量数据 insert 语句

1. 创建一个新的顾客表NewCustomer，把所有来自“中国”的顾客插入到这个表NewCustomer

执行以下SQL语句：

CREATE TABLE newcustomer LIKE customer

INSERT INTO newcustomer

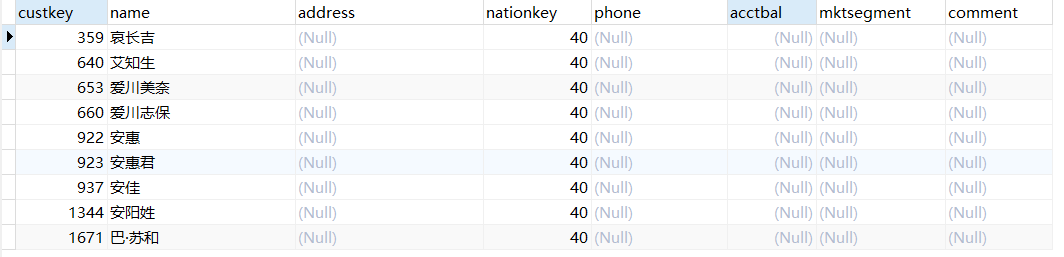
SELECT \*

FROM customer

WHERE nationkey = (SELECT nationkey FROM nation WHERE `name` = '中国')

将其结果查询后部分截屏如下：

SELECT \* FROM newcustomer;



2. 创建一个顾客购物统计表，记录每个顾客及其购物总数和总价等信息。

执行以下SQL语句：

CREATE TABLE shoppingstat(custkey INTEGER, quantity REAL, totalprice REAL)

INSERT INTO shoppingstat

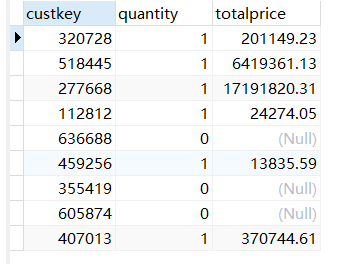
SELECT custkey, count(totalprice), SUM(totalprice)

FROM orders

GROUP BY custkey

将其结果查询后截屏如下：

SELECT \* FROM ShoppingStat;



3. 倍增零件表的数据，多次重复执行，直到总记录数达到 50 万为止。

insert into PART

select PARTKEY+(select max(partkey) from PART), NAME, MFGR, BRAND, TYPE, SIZE, CONTAINER, RETAILPRICE, COMMENT

from PART;

多次执行上述语句后，观察表格可以发现，其最终结果满足要求。

将其结果查询后截屏如下：

SELECT \* from part WHERE partkey>500000;



（4）update 语句（修改部分记录的部分列值）

将“双思 集团”供应的所有零件的供应成本价下降10%。

执行以下SQL语句：

UPDATE partsupp

SET supplycost = supplycost \* 0.9

WHERE suppkey in (SELECT suppkey FROM Supplier WHERE name='双思 集团')

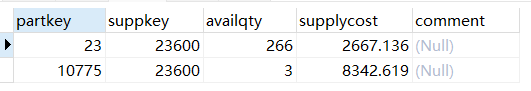
部分查询结果截屏如下：

select \*

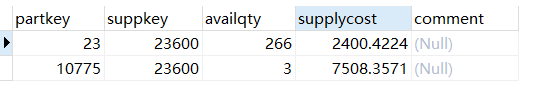
from PartSupp

WHERE suppkey in (SELECT suppkey FROM Supplier WHERE name='双思 集团') ;

1 更新前：



2 更新后：



比较两者的 SUPPLYCOST 可以发现， SUPPLYCOST 的供应成本价较 update 前确实下降了 10%，其最终结果满足要求。

（5）update 语句（利用一个表的数据修改另外一个表的数据）

利 用 part 表 中 的 零 售 价 格 来 修 改 lineitem 中 的 extendedprice ， 其 中 extendedprice = part.retailprice\*quantity。

执行以下SQL语句：

update LINEITEM L INNER JOIN PART P USING(partkey)

set L.EXTENDEDPRICE = P.RETAILPRICE \* L.QUANTITY

或者

UPDATE LINEITEM L

INNER JOIN PART P ON L.partkey = P.partkey

SET L.EXTENDEDPRICE = P.RETAILPRICE \* L.QUANTITY;

部分查询结果截屏如下：



在备份tpch2上运行的

（6）delete 基本语句（删除给定条件的所有记录）

删除顾客“阿波罗”的所有订单记录。

执行以下SQL语句：

DELETE l FROM lineitem AS l

JOIN orders AS o ON l.orderkey = o.orderkey

WHERE o.custkey = (SELECT custkey FROM customer WHERE `name` = '阿波罗')

DELETE FROM orders

WHERE orders.custkey = (SELECT custkey FROM customer WHERE `name` = '阿波罗')

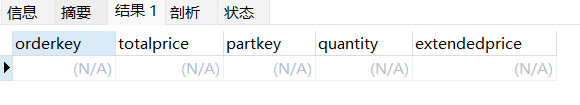
再查询‘阿波罗’的订单记录，结果如下：

SELECT O.orderkey,O.totalprice,L.partkey,L.quantity,L.extendedprice

FROM customer C,Orders O,lineitem L

WHERE C.custkey=O.custkey and O.orderkey=L.orderkey and C.name='阿波罗';

部分查询结果截屏如下：



说明：结果表明，已经找不到该顾客的订单记录，说明删除该顾客的订单记录成功。

* + - 1. **与实验结果相关的文件**

无

* + - 1. **实验总结**

在这次实验中，我深刻领悟到了数据库操作中的关键概念和技能：SQL语句对数据库的插入、修改、删除操作。

首先，通过实验，我学会了如何使用SQL语句进行单个元组的插入，以及如何进行批量数据插入。这是非常实用的技能，能够帮助我将数据有效地导入数据库中，以便进一步的分析和处理。

其次，实验还包括了修改数据和删除数据的操作。理解和掌握了UPDATE和DELETE语句的结构后，我能够轻松地对数据库中的数据进行更新和删除，确保数据的准确性和完整性。

实验的重点是插入、修改和删除数据的SQL语句，而难点则在于与嵌套SQL子查询的结合使用。这要求我们思考如何设计复杂的SQL查询，以满足特定的数据操作需求。例如，利用一个表的数据来插入、修改和删除另一个表的数据，这需要深入理解SQL语法和子查询的应用。

总的来说，这次实验让我更加熟练地运用SQL语句进行数据库操作，这对于未来在数据库管理和数据分析领域的工作将非常有帮助。我认识到了SQL的强大功能和灵活性，同时也明白了数据操作的重要性，以确保数据的质量和一致性。通过不断练习和调试，我相信我可以更好地应用这些技能，解决实际的数据管理和分析问题。