《数据库概论》实验一：用SQL进行数据操作 实验报告

姓名 学号 联系方式

实验环境

操作系统：Ubuntu 18.04

数据库管理系统：Mysql 8.0

软件：JetBrains DataGrip

实验过程

1. 使用SQL语句建立基本表

CREATE DATABASE IF NOT EXISTS HW1;  
USE HW1;  
  
CREATE TABLE Customers (  
 cid CHAR(4) NOT NULL,  
 cname CHAR(20) NOT NULL,  
 city CHAR(20) NOT NULL,  
 discnt REAL,  
 PRIMARY KEY (cid)  
);  
  
CREATE TABLE Agents (  
 aid CHAR(3) NOT NULL,  
 aname CHAR(20) NOT NULL,  
 city CHAR(20) NOT NULL,  
 perc SMALLINT,  
 PRIMARY KEY (aid)  
);  
  
CREATE TABLE Products (  
 pid CHAR(3) NOT NULL,  
 pname CHAR(20) NOT NULL,  
 city CHAR(20),  
 quantity INT NOT NULL,  
 price REAL NOT NULL,  
 PRIMARY KEY (pid)  
);  
  
CREATE TABLE Orders (  
 ordno INT NOT NULL,  
 orddate DATE NOT NULL,  
 cid CHAR(4) NOT NULL,  
 aid CHAR(3) NOT NULL,  
 pid CHAR(3) NOT NULL,  
 qty INT,  
 dols REAL,  
 PRIMARY KEY (ordno)  
);

1. 使用SQL插入数据
   1. 向Customers表插入数据

INSERT INTO Customers (cid, cname, city, discnt)  
VALUES  
('c001', 'Tiptop', 'Duluth', 10.00),  
('c002', 'Basics', 'Dallas', 12.00),  
('c003', 'Allied', 'Dallas', 8.00),  
('c004', 'ACME', 'Duluth', 8.00),  
('c006', 'ACME', 'Kyoto', 0.00);

* 1. 向Agents表插入数据

INSERT INTO Agents (aid, aname, city, perc)  
VALUES  
('a01', 'Smith', 'New York', 6),  
('a02', 'Jones', 'Newark', 6),  
('a03', 'Brown', 'Tokyo', 7),  
('a04', 'Gray', 'New York', 6),  
('a05', 'Otasi', 'Duluth', 5),  
('a06', 'Smith', 'Dallas', 5);

* 1. 向Product插入数据

INSERT INTO Products (pid, pname, city, quantity, price)  
VALUES  
('p01', 'comb', 'Dallas', 111400, 0.50),  
('p02', 'brush', 'Newark', 203000, 0.50),  
('p03', 'razor', 'Duluth', 150600, 1.00),  
('p04', 'pen', 'Duluth', 125300, 1.00),  
('p05', 'pencil', 'Dallas', 221400, 1.00),  
('p06', 'folder', 'Dallas', 123100, 2.00),  
('p07', 'case', 'Newark', 100500, 1.00);

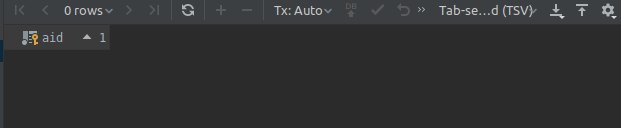
* 1. 向Orders插入数据

INSERT INTO Orders (ordno, orddate, cid, aid, pid, qty, dols)  
VALUES  
(1011, '2016-01-08', 'c001', 'a01', 'p01', 1000, 450.00),  
(1012, '2016-01-12', 'c001', 'a01', 'p01', 1000, 450.00),  
(1019, '2016-02-24', 'c001', 'a02', 'p02', 400, 180.00),  
(1017, '2016-02-10', 'c001', 'a06', 'p03', 600, 540.00),  
(1018, '2016-02-16', 'c001', 'a03', 'p04', 600, 540.00),  
(1023, '2016-03-12', 'c001', 'a04', 'p05', 500, 450.00),  
(1022, '2016-03-08', 'c001', 'a05', 'p06', 400, 720.00),  
(1025, '2016-04-07', 'c001', 'a05', 'p07', 800, 720.00),  
(1013, '2016-01-13', 'c002', 'a03', 'p03', 1000, 880.00),  
(1026, '2016-05-20', 'c002', 'a05', 'p03', 800, 704.00),  
(1015, '2016-01-23', 'c003', 'a03', 'p05', 1200, 1104.00),  
(1014, '2016-01-18', 'c003', 'a03', 'p05', 1200, 1104.00),  
(1021, '2016-02-28', 'c004', 'a06', 'p01', 1000, 460.00),  
(1016, '2016-01-25', 'c006', 'a01', 'p01', 1000, 500.00),  
(1020, '2016-02-05', 'c006', 'a03', 'p07', 600, 600.00),  
(1024, '2016-03-12', 'c006', 'a06', 'p01', 800, 400.00);

1. 使用SQL语言写出下列查询，并将查询结果保留在实验报告中
   1. 检索没有为居住在Duluth的任何客户订购过任何商品的经销商的编号

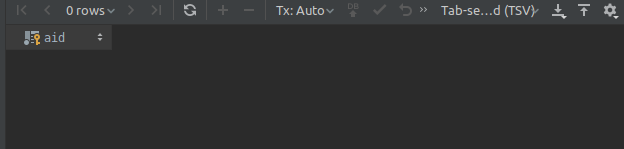
SELECT Agents.aid  
FROM Agents  
WHERE Agents.aid NOT IN (  
 SELECT DISTINCT Agents.aid  
 FROM Agents, Orders, Customers  
 WHERE Orders.aid=Agents.aid AND Orders.cid=Customers.cid AND Customers.city='Duluth'  
 );

查询为空



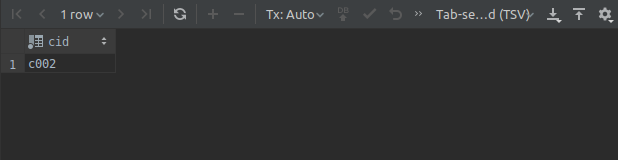
* 1. 检索为居住在Duluth和Kyoto的所有客户订购过同一种商品的经销商的编号

SELECT Agents.aid  
FROM Agents  
WHERE *EXISTS*(  
 SELECT Products.pid  
 FROM Products  
 WHERE NOT *EXISTS* (  
 SELECT \*  
 FROM Customers  
 WHERE Customers.city IN ('Duluth', 'Kyoto') AND NOT *EXISTS* (  
 SELECT \*  
 FROM Orders  
 WHERE Orders.aid=Agents.aid AND Orders.pid=Products.pid AND Orders.cid=Customers.cid  
 )  
 )  
 );



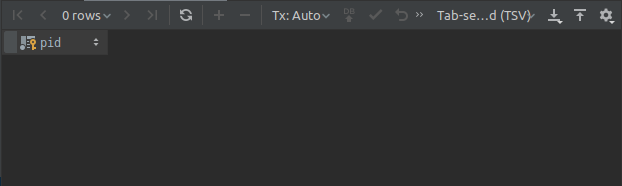
* 1. 检索仅通过a03和a05两个经销商订购过商品的客户编号

SELECT DISTINCT o.cid  
FROM Orders o  
WHERE o.aid='a03'  
 AND o.cid IN (  
 SELECT d.cid  
 FROM Orders d  
 WHERE d.aid='a05'  
 )  
 AND NOT *EXISTS* (  
 SELECT \*  
 FROM Orders r  
 WHERE r.cid=o.cid AND r.aid NOT IN ('a03', 'a05')  
 );



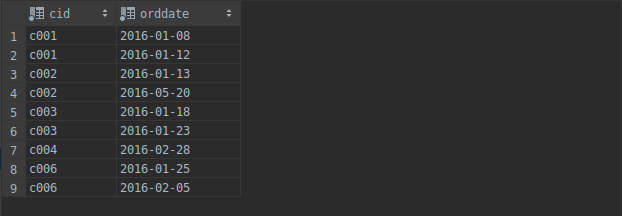
* 1. 在所有有客户的城市中都被销售过的商品的编号

SELECT Products.pid  
FROM Products  
WHERE NOT *EXISTS* (  
 SELECT Customers.city  
 FROM Customers  
 WHERE NOT *EXISTS* (  
 SELECT \*  
 FROM Orders, Customers c  
 WHERE Orders.cid=c.cid AND Orders.pid=Products.pid AND c.city=Customers.city  
 )  
 );



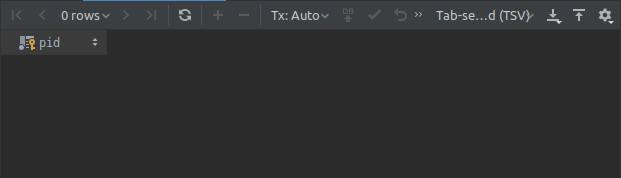
* 1. 返回每一个客户的编号及其最后两份订单的订购日期（按照订单编号的大小区分订单的先后）

SELECT Orders.cid, Orders.orddate  
FROM Orders  
WHERE 2 > (  
 SELECT *COUNT*(\*)  
 FROM Orders o  
 WHERE o.cid=Orders.cid AND o.orddate < Orders.orddate  
 )  
ORDER BY Orders.cid, Orders.ordno ASC;



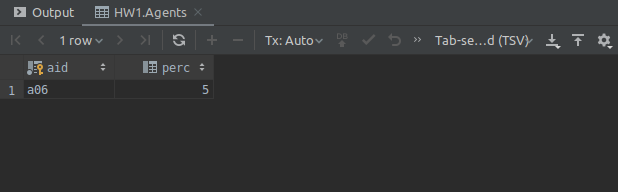
* 1. 检索居住在Dallas的所有客户都订购过的商品编号

SELECT Products.pid  
FROM Products  
WHERE NOT *EXISTS* (  
 SELECT \*  
 FROM Customers  
 WHERE Customers.city='Dallas' AND NOT *EXISTS* (  
 SELECT \*  
 FROM Orders  
 WHERE Orders.pid=Products.pid AND Orders.cid=Customers.cid  
 )  
 );



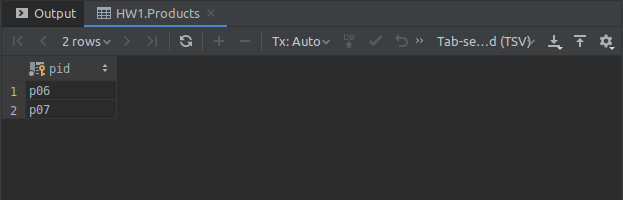
* 1. 检索为居住在Duluth的所有客户订购过商品的经销商的编号及其佣金百分比，并按照佣金百分比的降序输出查询结果

SELECT Agents.aid, Agents.perc  
FROM Agents  
WHERE NOT *EXISTS* (  
 SELECT \*  
 FROM Customers  
 WHERE Customers.city='Duluth' AND NOT *EXISTS*(  
 SELECT \*  
 FROM Orders  
 WHERE Orders.aid=Agents.aid AND Orders.cid=Customers.cid  
 )  
 )  
ORDER BY Agents.perc DESC ;



* 1. 检索符合下述条件的商品的编号：至少有一个客户通过与该客户位于同一个城市的经销商订购过该商品

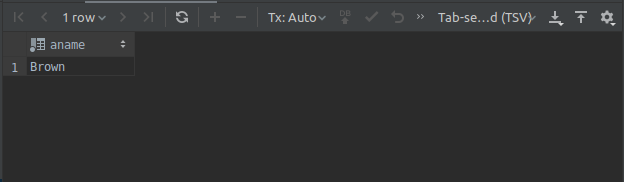
SELECT Products.pid  
FROM Products  
WHERE *EXISTS*(  
 SELECT \*  
 FROM Agents a, Customers c, Orders o  
 WHERE o.aid=a.aid AND o.cid=c.cid AND c.city=a.city AND o.pid=Products.pid  
 );



* 1. 检索享有最高销售提成比例的经销商（请分别写出使用统计函数和不使用统计函数的两种不同表示方法）

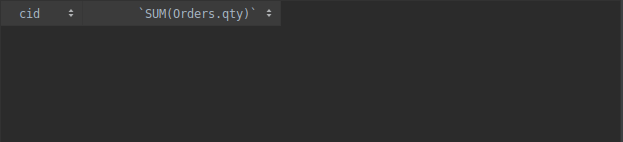
SELECT a.aname  
FROM Agents a  
WHERE a.perc >= ALL(  
 SELECT Agents.perc  
 FROM Agents  
);

SELECT a.aname  
FROM Agents a  
WHERE a.perc = ALL(  
 SELECT *MAX*(Agents.perc)  
 FROM Agents  
);



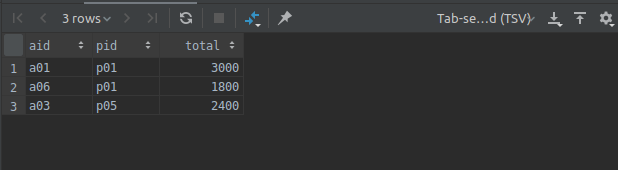
* 1. 检索仅仅通过a04号经销商订购过商品的客户编号，并给出每个客户的订购总金额

SELECT Orders.cid, *SUM*(Orders.qty)  
FROM Orders  
WHERE Orders.aid='a04' AND Orders.cid NOT IN (  
 SELECT o.cid  
 FROM Orders o  
 WHERE o.aid<>'a04'  
 )  
GROUP BY Orders.cid



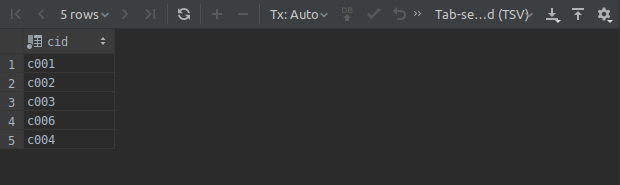
* 1. 检索每个经销商销售每一种产品的总数量，结果只需要返回每一种商品中，销售总数量排名前三且数量统计超过1000的

SELECT t.aid, t.pid, t.total  
FROM  
 (SELECT Orders.aid, Orders.pid, *SUM*(Orders.qty) AS total  
 FROM Orders  
 GROUP BY Orders.aid, Orders.pid) t  
WHERE t.total>1000 AND 3>(  
 SELECT *COUNT*(\*)  
 FROM  
 (SELECT Orders.aid, Orders.pid, *SUM*(Orders.qty) AS total  
 FROM Orders  
 GROUP BY Orders.aid, Orders.pid) t1  
 WHERE t1.pid=t.pid AND t1.total>t.total  
 )  
ORDER BY t.pid ASC;



* 1. 检索符合下述要求的客户的编号：在该客户订购过的所有商品中，每一种商品的平均每笔订单的订购数量均达到或超过300

SELECT DISTINCT Orders.cid  
FROM Orders  
WHERE NOT *EXISTS* (  
 SELECT o.pid  
 FROM Orders o  
 WHERE o.cid=Orders.cid  
 GROUP BY o.pid  
 HAVING *AVG*(o.qty<300)  
 );



1. 删除基本表

DROP TABLE Agents, Customers, Orders, Products;

实验中遇到的困难及解决办法

[详细说明你认为本次实验中比较困难的地方，也可以对实验设计提出建议]

参考文献及致谢

[如果你参考了任何书籍、网页，或与他人进行了讨论，请在此注明]