

实验三 复杂查询，触发器和存储过程

班级： 07012201 学号： 1120220715 姓名： 刘秉致

一、实验目的

掌握 SQL 嵌套查询和集合查询等各种高级查询的设计方法；掌握数据库触发器的设计和使用方法。

二、实验内容

一、请用 SQL 语言实现以下查询

- (1) 查询订购了“徐州市泰力公司矿山设备四厂”制造的“活塞式减压阀”的顾客的信息。
- (2) 查询没有购买过“徐州市泰力公司矿山设备四厂”制造的“活塞式减压阀”的顾客的信息。
- (3) 查询订单平均金额超过 500 元的顾客中的中国籍顾客的信息。
- (4) 查询顾客“薛融”订购过而“宣荣揣”没有订购过的零件的信息。

二、请建立以下触发器

- (1) 在 Lineitem 表上定义一个 INSERT 触发器，当增加一项订单明细时，自动修改订单表 Orders 中的订单总金额，以保持数据的一致性。（增加的那一部分金额为：零售价×数量×折扣）
- (2) 在 Lineitem 表上定义一个 BEFORE UPDATE 触发器，当修改订单明细中的数量时，先检查零件供应表 PartSupp 中的可用数量是否足够。

三、实验步骤

一、用 SQL 语言实现以下查询

- (1) 使用 4 层嵌套查询的 SQL 查询语句

```
SELECT * FROM customer WHERE CNo IN
(SELECT CNo FROM orders WHERE ONo IN
    (SELECT ONo FROM lineitem WHERE PNo IN
```

```
(SELECT Pno FROM part WHERE Pname="活塞式减压阀"
AND PProduct="徐州市泰力公司矿山设备四厂")
);
```

- (2) 利用上一操作的查询结果，利用“全集-补集”的差操作建立如下查询语句

```
SELECT * FROM customer WHERE CNo NOT IN/*求差操作*/
(SELECT CNo FROM orders WHERE ONo IN
(SELECT ONo FROM lineitem WHERE PNo IN
(SELECT Pno FROM part WHERE Pname="活塞式减压阀"
AND PProduct="徐州市泰力公司矿山设备四厂")
);
```

- (3) 使用两个嵌套查询对属性进行筛选，其中使用 GROUP BY 子句和 AVG 集函数对平均金额进行筛选

```
SELECT * FROM customer
/*由于数据库中录入信息时国家名后的空格没有截断，只能采用字符串匹配查找*/
WHERE NNo IN (SELECT NNo FROM nation WHERE NName LIKE "中国%")
AND Cno IN(
/*按 Cno 分组并在每个组内使用集函数判断平均金额*/
SELECT CNo FROM orders GROUP BY (CNo)
HAVING AVG(OPrice)>500
);
```

- (4) 使用嵌套查询分别查询两人订购的零件的集合并用求差集合操作获得结果

```
SELECT * FROM part WHERE PNo IN
(
(SELECT PNo FROM lineitem WHERE ONo IN
(SELECT ONo FROM orders WHERE CNo IN
(SELECT CNo FROM customer WHERE CName="薛融")
```

```

    )
  )
  EXCEPT /*要求两个集合的差集*/
  (SELECT PNo FROM lineitem WHERE ONo IN
    (SELECT ONo FROM orders WHERE CNo IN
      (SELECT CNo FROM customer WHERE CName="宣荣揣")
    )
  )
);

```

二、建立以下触发器并验证

(1) 建立 INSERT 触发器语句如下

```

CREATE TRIGGER self_update AFTER INSERT
ON lineitem FOR EACH ROW
BEGIN
  /*由于插入记录内无零售价格，因此内嵌一个查询，获取零售价格*/
  UPDATE orders SET
  OPrice=OPrice+(NEW.Amount*NEW.Discount*(SELECT PCost FROM part
  WHERE PNo=NEW.PNO))
  WHERE orders.ONo=NEW.ONo;
END;

```

通过创建如下事务进行验证

```

START TRANSACTION;
/*先查询订单编号为 1 的订单的相关信息和和查询 lineitem 表中订单编号为 1 的订单明
细*/
SELECT * FROM orders WHERE ONo=1;
SELECT * FROM lineitem WHERE ONo=1;
/*在 lineitem 表中插入一条记录,订单 1 中补充订购 1 号零件，折扣为 0.9，数量 100*/
INSERT INTO lineitem VALUES(1,1,1,100,0,0.9);
/*再查询订单编号为 1 的订单的相关信息和查询 lineitem 表中订单编号为 1 的订单明
细*/
SELECT * FROM orders WHERE ONo=1;

```

```
SELECT * FROM lineitem WHERE ONo=1;
ROLLBACK;
```

(2) 建立 BEFORE UPDATE 触发器如下

```
CREATE TRIGGER enough_check BEFORE UPDATE
ON lineitem FOR EACH ROW
BEGIN
    /*先看是否增多*/
    IF(NEW.Amount>OLD.Amount) THEN
        /*再看是否足够*/
        IF(NEW.Amount>(SELECT Quantity FROM partsupp WHERE
(PNo=NEW.PNo AND SNo=NEW.SNo))) THEN
            /*mysql 中无法直接打印字符串，因此只能自行定义错误信息并返回
*/
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "NOT
ENOUGH";
        END IF;
    END IF;
END;
```

建立测试事务如下

```
/*先查询订单编号为 5426 的订单的相关信息和其中所有的零件的可用容量*/
SELECT * FROM lineitem WHERE ONo=5426;
SELECT * FROM partsupp WHERE PNo IN(SELECT PNo FROM lineitem WHERE
ONo=5426);
/*更改由供应商编号为 11642、零件编号为 2 的数量为 1000*/
UPDATE lineitem SET Amount=1000 WHERE (ONo=5426 AND Pno=2 AND
SNo=11642);
ROLLBACK;
```

四、实验结果及分析

一、 用 SQL 语言实现以下查询

(1) 结果如下

	CNo int	CName varchar	NNo int
> 1	8	阿波罗	162

共 1 条记录。

(2) 结果如下

耗时: 33ms < 1 2 3 4 ... 6614 > Total 661395				
	CNo int	CName varchar	NNo int	
> 1	1	阿宝宝	138	
> 2	2	阿爆	186	
> 3	3	阿倍仲麻吕	190	
> 4	4	阿比	222	
> 5	5	阿碧	13	
> 6	6	阿扁	43	
> 7	7	阿炳	37	
> 8	9	阿伯拉默夫	190	
> 9	10	阿博德	144	
> 10	11	阿卜杜尔	206	
> 11	12	阿卜杜热合曼	186	
> 12	13	阿不都	138	
> 13	14	阿不都哈地尔	195	
> 14	15	阿不都卡德尔	172	
> 15	16	阿不都克里木	178	
> 16	17	阿不都克力木	216	
> 17	18	阿不都麻木提	140	
> 18	19	阿不都尼衣木	87	
> 19	20	阿不都热合曼	154	

共 661395 条记录。

(3) 结果如下

耗时: 146ms < 1 2 3 > Total 294

	CNo int	CName varchar	NNo int	
> 1	51404	方铭坚	40	
> 2	65677	龚日升	40	
> 3	72536	郭世和	40	
> 4	80960	何小静	40	
> 5	81243	何吕施张	40	
> 6	89917	扈元静	40	
> 7	91887	黄何凝	40	
> 8	111438	康有为	40	
> 9	113571	蓝孟	40	
> 10	118915	李德宇	40	
> 11	120101	李继超	40	
> 12	148321	刘剥啊	40	
> 13	156407	刘宸跃	40	
> 14	158957	刘崇密	40	
> 15	177682	刘罡之	40	
> 16	214619	刘经练	40	
> 17	214940	刘井义	40	
> 18	216410	刘婧晖	40	
> 19	221107	刘可奇	40	

共 294 条记录。

(4) 结果如下

耗时: 1.6s < 1 > Total 3

	PNo int	PName varchar	PProduct varchar	PSzie varchar
> 1	4577	塑料压延机	江苏省南通阀门厂	Sy-4T1200
> 2	12450	VLV1kv铝芯聚氯乙烯绝缘聚	大连通产高压釜容器制造有	标称截面: 1.5
> 3	52770	高低密度聚乙烯薄膜吹塑机	重庆万里蓄电池股份有限公	SJ45×28□-M

共 3 条记录

二、建立以下触发器并验证

(1) 执行测试事务 1，查询订单编号为 1 的订单的相关信息如下。

	Q	* ONo int	* CNo int	* ODate varchar(10)	OPrice float
	>1	1	320728	2014/5/14	437

查询 lineitem 表中订单编号为 1 的订单明细如下。

	Q	* ONo int	* PNo int	* SNo int	* Amount int	* Refund int	* Discount float
	>1	1	42522	13503	2	0	0.272896
	>2	1	44930	18848	1	0	0.0871442

在 lineitem 表中插入一条记录, 订单 1 中补充订购 1 号零件, 折扣为 0.9, 数量 100, 再次查询。

	Q	* ONo int	* CNo int	* ODate varchar(10)	OPrice float
	>1	1	320728	2014/5/14	707

发现数据已自动更新成功, 触发器正常执行。

(2) 执行测试事务 2, 先查询订单编号为 5426 的订单的相关信息。

	Q	* ONo int	* PNo int	* SNo int	* Amount int	* Refund int	* Discount float
	>1	5426	2	11642	3	0	0.944363
	>2	5426	13865	10643	1	0	0.384125
	>3	5426	54376	28047	2	0	0.441669

查询其中所有的零件的可用容量。

Q	PNo int	SNo int	Quantity int	Cost float
>1	2	1	0	0
>2	2	803	478	4455.96
>3	2	11642	922	9970.97
>4	13865	10643	584	6204.99
>5	54376	28047	544	2931.44

更改由供应商编号为 11642、零件编号为 2 的数量为 1000, 显然数量明显超过供应上界, 发现无法更新, 并且返回自定义错误信息 “NOT ENOUGH”。触发器执行成功。

```
> Run | New tab
UPDATE lineitem SET Amount=1000 WHERE (ONo=5426 AND Pno=2 AND SNo=11642); 4ms NOT ENOUGH
```

五、实验收获与体会

通过此次实验，我加深了对于 SQL 语句有关嵌套查询部分和触发器部分内容的理解，更对 MySQL 的语法有了更深的了解。

附录：程序清单及说明

1. 1120220715_刘秉致_实验代码.sql
为本次实验的代码