

LAB 10: DATABASE TRIGGER

ใช้ตารางจาก Lab1

ตัวอย่างที่ 1 การสร้าง TRIGGER สำหรับเพิ่มข้อมูลที่ตาราง Order_details

```
CREATE OR REPLACE TRIGGER INSERT_ORDERDETAILS_TRG
AFTER INSERT ON L1ORDER_DETAILS
FOR EACH ROW
BEGIN
    UPDATE L1PRODUCTS SET QUANTITY = QUANTITY - :NEW.QUANTITY
    WHERE PRODUCTID = :NEW.PRODUCTID;
END;
```

คำสั่งที่ทำให้ trigger นี้ทำงานคือ

```
INSERT INTO L1ORDER_DETAILS VALUES(10258,4,250);
```

ผลลัพธ์คือ ตาราง L1ORDER_DETAILS จะมีข้อมูลเพิ่ม 1 แถว และ จำนวนสินค้าคงเหลือ (QUANTITY) รหัส 4 ในตาราง L1PRODUCTS จะลดลง 250 ชิ้น

จงสร้าง TRIGGER ต่อไปนี้

1. TRIGGER สำหรับ ลบข้อมูลตาราง L1Order_Details

คำสั่งที่ทำให้ TRIGGER นี้ทำงานคือ

```
DELETE FROM L1ORDER_DETAILS WHERE ORDERID = 10256 AND PRODUCTID = 5;
```

ผลลัพธ์คือ ตาราง L1ORDER_DETAILS จะถูกลบข้อมูล 1 แถว และ จำนวนสินค้าคงเหลือ (QUANTITY) รหัส 5 ในตาราง L1PRODUCTS จะเพิ่มขึ้น 105 ชิ้น (ถ้าข้อมูลเดิมในตารางเป็น 105 ชิ้น)

2. TRIGGER สำหรับ แก้ไขข้อมูลตาราง L1Order_Details เฉพาะคอลัมน์ Quantity หากมีการแก้ไข จำนวนสินค้าต้องจัดการกับจำนวนคงเหลือในคลังด้วย

คำสั่งที่ทำให้ TRIGGER นี้ทำงานคือ

```
UPDATE L1ORDER_DETAILS SET QUANTITY = 500 WHERE ORDERID = 10256 AND
PRODUCTID = 3;
```

ผลลัพธ์คือ ตาราง L1ORDER_DETAILS จะถูกแก้ไขข้อมูล 1 แถว และ จำนวนสินค้าคงเหลือ (QUANTITY) รหัส 3 ในตาราง L1PRODUCTS จะลดลง 260 ชิ้น (ถ้าข้อมูลเดิมในตารางเป็น 240 ชิ้น)

3. TRIGGER สำหรับเก็บข้อมูลย้อนหลังเมื่อมีการแก้ไขราคาสินค้า โดยโปรแกรมจะทำการบันทึกลงตาราง L1Price_History ซึ่งประกอบด้วยคอลัมน์ ดังนี้

L1Price_History

| คอลัมน์ | ชนิดข้อมูล | หมายเหตุ |
|-----------|-------------|--|
| HistID | NUMBER(5) | เป็นเลข Auto Number (Sequence) และเป็น Primary Key |
| HistDate | DATE | วันที่แก้ไขข้อมูล |
| ProductID | NUMBER(4) | รหัสสินค้าที่แก้ไข |
| UnitPrice | NUMBER(7,2) | ราคาเก่า |

คำสั่งที่ทำให้ TRIGGER นี้ทำงานคือ

```
UPDATE L1PRODUCTS SET UNITPRICE = 60.50 WHERE PRODUCTID = 3;
```

ผลลัพธ์คือ ตาราง L1PRODUCTS จะถูกแก้ไขข้อมูล 1 แถว และ ตาราง L1PRICE_HISTORY จะเพิ่มข้อมูล 1 แถว