

LAB 9: PROCEDURE & FUNCTION

ใช้ตารางจาก Lab1

ตัวอย่างที่ 1 การสร้าง Procedure แบบไม่มีการส่ง Parameter

ต้องการสร้าง procedure สำหรับเพิ่มราคาสินค้า โดยกำหนดค่าคงที่เป็น 1.10 ของสินค้ารหัส 1 – 5 เท่านั้น

```
CREATE OR REPLACE PROCEDURE PROC_UPDATE_UNITPRICE IS
BEGIN
    FOR I IN 1..5 LOOP
        UPDATE L1PRODUCTS SET UNITPRICE = UNITPRICE * 1.10
            WHERE PRODUCTID = I;
    END LOOP;
    COMMIT;
END;
```

วิธีการเรียกใช้ คือ **EXECUTE PROC_UPDATE_UNITPRICE;**

ตัวอย่างที่ 2 การสร้าง Procedure แบบมีการส่ง Parameter

ต้องการสร้าง procedure สำหรับเพิ่มราคาสินค้า โดยรับค่าอัตรา และ รหัสสินค้าเข้ามาทีละ 1 สินค้า

```
CREATE OR REPLACE PROCEDURE PROC_UPDATE_UNITPRICE_BYRATE
(RATE NUMBER, PID NUMBER) IS
BEGIN
    UPDATE L1PRODUCTS SET UNITPRICE = UNITPRICE * RATE
        WHERE PRODUCTID = PID;
    COMMIT;
END;
```

วิธีการเรียกใช้ คือ **EXECUTE PROC_UPDATE_UNITPRICE_BYRATE (1.25, 3);**

ส่งอัตราที่ต้องการเพิ่ม และ รหัสสินค้าเข้าไป

ตัวอย่างที่ 3 การสร้าง Function

ต้องการสร้าง function สำหรับคำนวณหาอัตราภาษีที่ต้องชำระของสินค้าแต่ละรายการ โดยรับค่ารหัสสินค้าเข้ามา หากราคาต่ำกว่า 100 จะคิดภาษี 7% นอกนั้นคิด 10%

```
CREATE OR REPLACE FUNCTION FUNC_TAX_UNITPRICE (PID NUMBER)
RETURN NUMBER IS
V_PRICE NUMBER(4,2);
V_TAX NUMBER(4,2);
BEGIN
SELECT UNITPRICE INTO V_PRICE FROM L1PRODUCTS
WHERE PRODUCTID = PID;
IF V_PRICE < 100 THEN
V_TAX := 0.07;
ELSE
V_TAX := 0.10;
END IF;
RETURN V_TAX;
END;
```

การเรียกใช้ function

```
SELECT PRODUCTID, PRODUCTNAME, UNITPRICE,
FUNC_TAX_UNITPRICE (PRODUCTID) AS TAX
FROM L1PRODUCTS;
```

ตัวอย่างที่ 4 การใช้ CURSOR ใน procedure หรือ function

ต้องการสร้าง procedure ที่มีการใช้ cursor ในการทำงาน โดยรับค่าอัตราราคาที่ต้องการเพิ่ม และราคาสินค้า เข้าไปใน procedure โดยให้ cursor เลือกข้อมูลสินค้าที่มีราคาต่ำกว่า ราคาสินค้าที่ระบุ จากนั้นตรวจสอบว่า หากจำนวนสินค้าน้อยกว่า 50 ชิ้น แสดงว่าสินค้าเหลือน้อย ให้เพิ่มอัตราการเพิ่มราคาเข้าไปอีก 0.1 นอกจากนั้นให้ใช้ Rate เดิมที่รับเข้ามา

```
CREATE OR REPLACE PROCEDURE PROC_UPDATE_PRICE_CUR(RATE NUMBER, PRICE NUMBER) IS
CURSOR C_PROD(P NUMBER) IS
SELECT PRODUCTID, QUANTITY FROM L1PRODUCTS WHERE UNITPRICE < P;
-- เลือกสินค้าที่มีราคาน้อยกว่าค่าที่รับเข้ามา
V_PID L1PRODUCTS.PRODUCTID%TYPE;
V_QTY L1PRODUCTS.QUANTITY%TYPE;
BEGIN
OPEN C_PROD(PRICE);
LOOP
FETCH C_PROD INTO V_PID, V_QTY;
EXIT WHEN C_PROD%NOTFOUND;
IF V_QTY < 50 THEN -- หากจำนวนน้อยกว่า 50 ชิ้นให้เพิ่ม Rate อีก 0.1
UPDATE L1PRODUCTS SET UNITPRICE = UNITPRICE * (RATE + 0.1)
WHERE PRODUCTID = V_PID;
```

```

ELSE -- หากจำนวนมากกว่าหรือเท่ากับ 50 ให้ใช้ Rate ที่รับเข้ามา
      UPDATE L1PRODUCTS SET UNITPRICE = UNITPRICE * RATE
      WHERE PRODUCTID = V_PID;
END IF;
END LOOP;
CLOSE C_PROD;
COMMIT;
END;

```

วิธีการเรียกใช้ คือ **EXECUTE PROC_UPDATE_PRICE_CUR (1.25, 200);**

ส่งอัตราที่ต้องการเพิ่มและ ราคาสินค้าต่อหน่วย

จงสร้าง procedure หรือ function สำหรับโจทย์ต่อไปนี้

- จงเขียน function สำหรับคืนค่าข้อความ ดังนี้

หากสินค้าเหลือ < 50 ชิ้น ให้แสดงว่า order now!!

หากสินค้าเหลือ > 300 ชิ้น ให้แสดงว่า sale now!!

นอกนั้นไม่ต้องแสดงค่า

คำสั่งในการเรียกใช้

Select productid, productname, **getProductText(productid)** as txt
From Products;
- จงเขียน function คำนวณจำนวนสินค้าที่มีการสั่งซื้อทั้งหมด ให้ส่ง parameter เข้าไป

คำสั่งในการเรียกใช้

Select productid,productName, **getCountProduct(productid)** as cnt_product
From Products;
- จงเขียน procedure สำหรับเพิ่มข้อมูลสินค้าใหม่

คำสั่งในการเรียกใช้

execute **insert_product_proc(30,'Oishi',300,18.00);**
- จงเขียน procedure สำหรับเพิ่มข้อมูลลงในตารางรายละเอียดการสั่งซื้อ (Order_Details) เมื่อมีการเพิ่มลงไปจะต้องไปตัดยอดคงเหลือในตาราง products ด้วย

คำสั่งในการเรียกใช้

execute **insert_orderDetail_proc(10257, 1, 100);**

ผลที่ได้คือตาราง order_details มีการเพิ่มข้อมูล 1 แถว และจำนวนสินค้ารหัสที่ 1 คงเหลือ (quantity) ในตาราง products ลดจำนวนลง 100 ชิ้น