ใบงานที่ 2 การสร้างและการใช้งาน Parameters

วัตถุประสงค์

- **เพื่อให้นิสิตสามารถสร้าง Parameters ในรูปแบบต่างๆ ได้**
- **เพื่อให้นิสิตสามารถสร้างและเรียกใช้ Stored Procedure ในรูปแบบการรับค่า Parameters เข้าได้**
- **เพื่อให้นิสิตสามารถสร้างและเรียกใช้ Stored Procedure ในรูปแบบการส่งค่า Parameters กลับได้**

<u>กำหนดให้</u>

A. กำหนดให้นิสิตสร้างตารางฐานข้อมูล “buyer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้ พร้อมทำการรันคำสั่งโดยแสดงผลลัพธ์ลงในตารางด้านล่าง

- SQL statement 1

```
CREATE TABLE IF NOT EXISTS `buyer` (
  `BUYER_ID` varchar(4) NOT NULL,
  `NAME` varchar(50) NOT NULL,
  `EMAIL` varchar(50) NOT NULL,
  `COUNTRY_CODE` varchar(2) NOT NULL,
  `BUDGET` decimal(18,2) NOT NULL,
  `USED` decimal(18,2) NOT NULL,
  PRIMARY KEY (`BUYER_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

B. กำหนดให้นิสิตเพิ่มข้อมูลลงในตารางฐานข้อมูล “buyer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้พร้อมทำการรันคำสั่งโดยแสดงผลลัพธ์ลงในตารางด้านล่าง

- SQL statement 2

```
INSERT INTO `buyer` (`BUYER_ID`, `NAME`, `EMAIL`, `COUNTRY_CODE`, `BUDGET`, `USED`) VALUES
('A001', 'Win Kai', 'win.weerachai@buu.com', 'TH', 8000000.00, 600000.00),
('A002', 'John  Zion', 'john.smith@ buu.com', 'TH', 7000000.00, 800000.00),
('A003', 'Jame Luca', 'jame.born@ buu.com', 'US', 6000000.00, 600000.00),
('A004', 'Chalee Maeve', 'chalee.angel@ buu.com', 'US', 5000000.00, 100000.00);
```

1. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

   1. จงสร้าง SP getBuyer
   2. หากมีการสร้าง SP getBuyer ก่อนนี้ให้ทำการลบออก
   3. กำหนดให้ SP getBuyer แสดงข้อมูลทุกคอลัมน์ในตาราง Buyer โดยให้ผู้ใช้สามารถกำหนดข้อมูลในคอลัมน์ COUNTRY_CODE เองได้

- จงระบุโค้ด SP getBuyer ในตาราง

Item	โค้ด
1	

```
DROP PROCEDURE IF EXISTS getBuyer;
DELIMITER $$
CREATE PROCEDURE getBuyer(IN pCountryCode VARCHAR(2))
BEGIN
    SELECT * FROM buyer WHERE buyer.COUNTRY_CODE = pCountryCode;
END $$
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		

- จงระบุคำสั่งเรียกใช้ SP getBuyer ในตารางในเงื่อนไขแสดงข้อมูล COUNTRY_CODE = ‘TH’ เท่านั้น

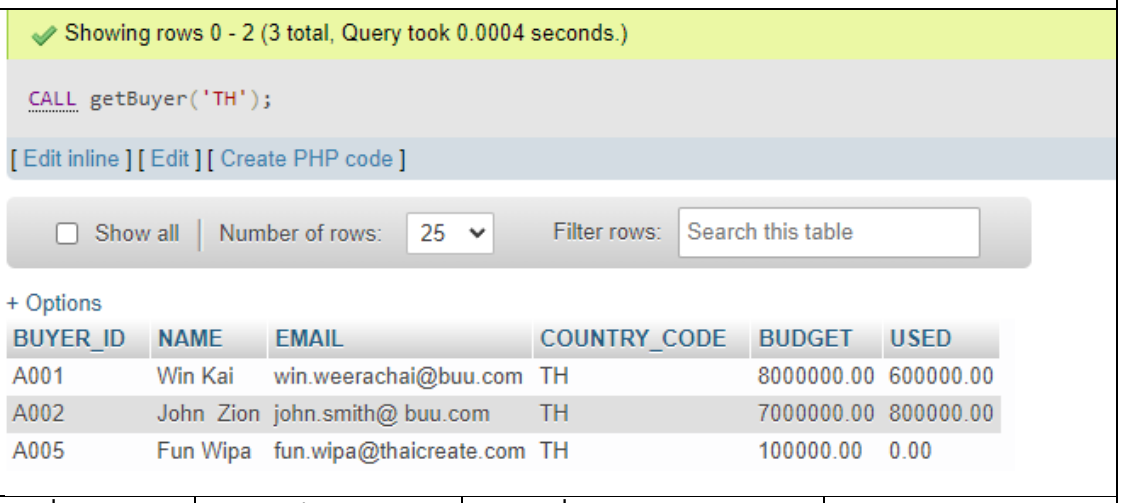
Item	คำสั่ง
2	

```
CALL getBuyer('TH');
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		

- จงระบุภาพหน้าจอผลลัพธ์การเรียก SP getBuyer ใช้จาก Item 2

Item	ภาพหน้าจอผลลัพธ์
3	



การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	1		

2. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyer
2. หากมีการสร้าง SP getBuyer ก่อนนี้ให้ทำการลบออก
3. กำหนดให้ SP getBuyer แสดงข้อมูลทุกคอลัมน์ในตาราง Buyer โดยให้ผู้ใช้สามารถกำหนดข้อมูลในคอลัมน์ COUNTRY_CODE เองได้ และสามารถกำหนดข้อมูลตัวเลขที่มีค่ามากกว่าหรือเท่ากับข้อมูลในคอลัมน์ USED ได้

- จงระบุโค้ด SP getBuyer

Item	โค้ด
1	

```
DROP PROCEDURE IF EXISTS getBuyer;
DELIMITER $$
CREATE PROCEDURE getBuyer(IN pCountryCode VARCHAR(2),IN pUsed
DECIMAL(10,2))
BEGIN
SELECT * FROM `buyer` WHERE COUNTRY_CODE = pCountryCode AND USED >= pUsed;
END $$
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	0.5		

- จงระบุคำสั่งเรียกใช้ SP getBuyer กรณีต้องการแสดง COUNTRY_CODE = ‘TH’ และ USED = ‘300000.00’

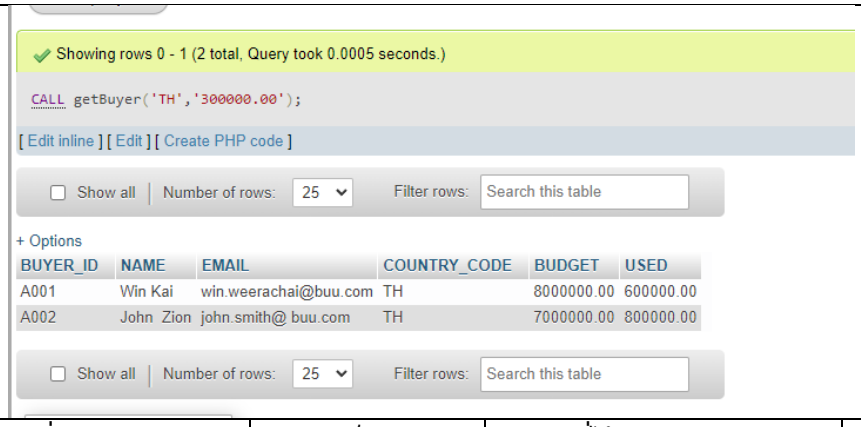
Item	คำสั่ง
2	

```
CALL getBuyer('TH','300000.00');
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	0.5		

- จงระบุภาพหน้าจอผลลัพธ์การเรียกใช้ SP getBuyer จาก Item 2

Item	ภาพหน้าจอผลลัพธ์
3	



การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	1		

3. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyer
2. หากมีการสร้าง SP getBuyer ก่อนนี้ให้ทำการลบออก
3. กำหนดให้ SP getBuyer แสดงตัวเลขจำนวน row ในตาราง Buyer โดยให้ผู้ใช้สามารถกำหนดข้อมูลในคอลัมน์ COUNTRY_CODE เองได้
4. กำหนดให้สร้าง SP getBuyerCount เพื่อทำหน้าที่นับจำนวน row ในตาราง Buyer พร้อมส่งค่าผลลัพธ์ตัวเลขจำนวน row กลับมาแสดงที่ SP getBuyer

- จงระบุโค้ด SP getBuyerCount

Item	โค้ด
1	

```
DROP PROCEDURE IF EXISTS getBuyerCount;
DELIMITER $$
CREATE PROCEDURE getBuyerCount(IN pCountryCode VARCHAR(2), OUT pBuyerCount
INT)
BEGIN
# Count Customer by CountryCode
SELECT COUNT(*) INTO pBuyerCount FROM `buyer` WHERE COUNTRY_CODE =
pCountryCode;
END $$
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	0.5		

- จงระบุโค้ด SP getBuyer

Item	โค้ด
2	

```
DROP PROCEDURE IF EXISTS getBuyer;
DELIMITER //
CREATE PROCEDURE getBuyer(IN pCountryCode VARCHAR(2))
BEGIN
# Declare Variable
DECLARE pBuyerCount INT DEFAULT 0;
# Call stored procedure getCustomerCount
CALL getBuyerCount(pCountryCode,pBuyerCount);
# Print
SELECT CONCAT('Buyer Count = ',pBuyerCount);
END //
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	0.5		

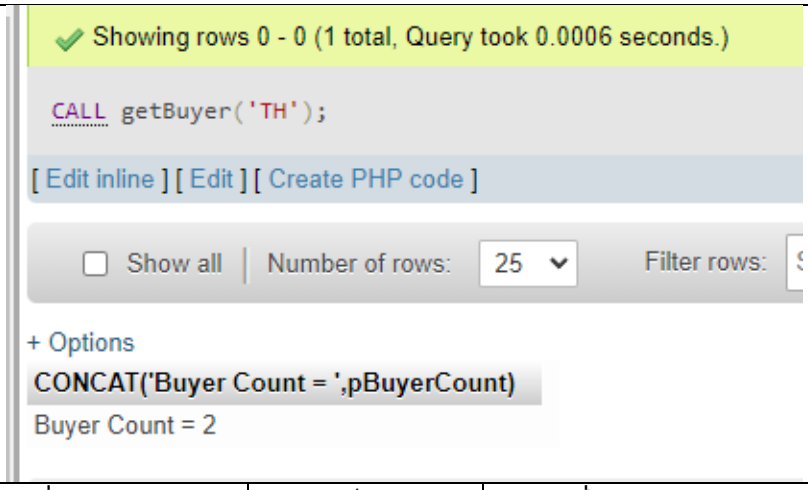
- จงระบุคำสั่งเรียกใช้ SP getBuyer กรณีต้องการแสดง COUNTRY_CODE = ‘TH’

Item	คำสั่ง
3	

```
CALL getBuyer('TH');
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	0.5		

- จงระบุภาพหน้าจอผลลัพธ์การสร้างสำเร็จลงในตารางด้านล่าง

Item	ภาพหน้าจอผลลัพธ์
4	

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	0.5		

4. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyer
2. หากมีการสร้าง SP getBuyer ก่อนนี้ให้ทำการลบออก
3. กำหนดให้ SP getBuyer แสดงตัวเลขจำนวน row ในตาราง Buyer โดยให้ผู้ใช้สามารถกำหนดข้อมูลในคอลัมน์ COUNTRY_CODE เองได้
4. กำหนดให้ส่งค่าตัวแปรกลับไปยังตัวแปรที่เรียกใช้งาน SP getBuyer พร้อมแสดงผลหน้าจอ

- จงระบุโค้ด SP getBuyer

Item	โค้ด
1	

```
DROP PROCEDURE IF EXISTS getBuyer;
DELIMITER $$
CREATE PROCEDURE getBuyer(IN pCountryCode VARCHAR(2), OUT pRowFound INT)
BEGIN
# Return SELECT
SELECT * FROM `buyer` WHERE COUNTRY_CODE = pCountryCode;
# Return pRowFound
SET pRowFound = FOUND_ROWS();
END $$
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	0.5		

- จงระบุคำสั่งเรียกใช้ SP getBuyer กรณีต้องการแสดง COUNTRY_CODE = ‘TH’

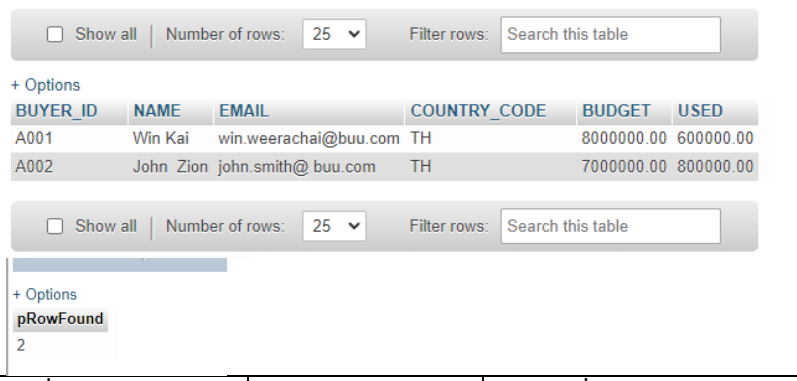
Item	คำสั่ง
2	

```
SET @p0='TH';
CALL `getBuyer`(@p0, @p1);
SELECT @p1 AS `pRowFound`;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	0.5		

- จงระบุภาพหน้าจอผลลัพธ์การเรียกใช้ SP getBuyer

Item	ภาพหน้าจอผลลัพธ์
3	



การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	1		

5. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP insertBuyer
2. หากมีการสร้าง SP insertBuyer ก่อนนี้ให้ทำการลบออก
3. กำหนดให้ SP insertBuyer มีหน้าที่ในการเพิ่มข้อมูลลงในตาราง Buyer
4. หากเพิ่มข้อมูลได้สำเร็จให้แสดงข้อความ ‘Insert Data Successfully’ พร้อมแสดงตัวเลขจำนวนของชุดข้อมูลที่เพิ่มเข้าไปใหม่

- จงระบุโค้ด SP insertBuyer

Item	โค้ด
1	

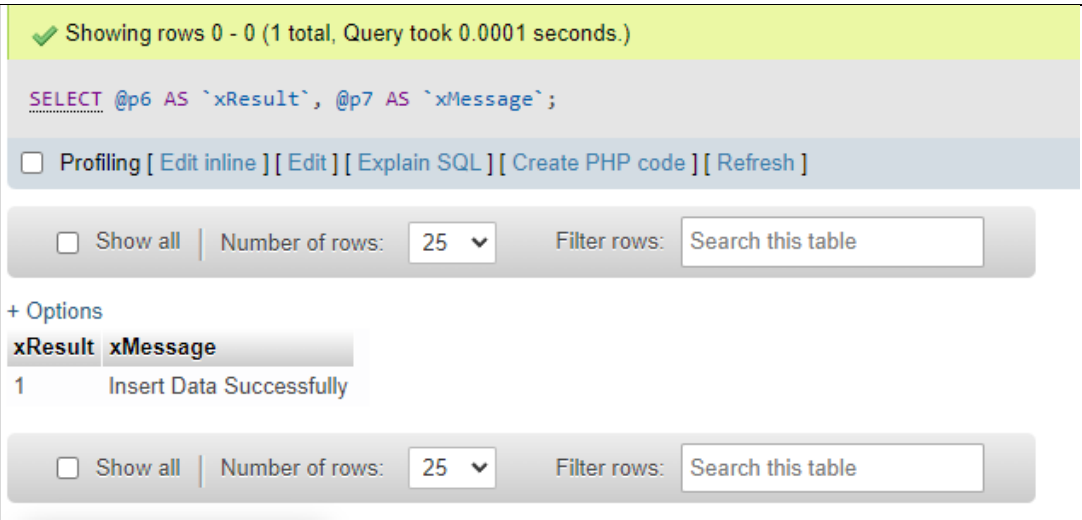
```
DROP PROCEDURE IF EXISTS insertBuyer;

DELIMITER $$

CREATE PROCEDURE insertBuyer(IN pBuyerID VARCHAR(4)
, IN pName VARCHAR(150), IN pEmail VARCHAR(150), IN pCountryCode
VARCHAR(2), IN pBudget DECIMAL(18, 2), IN pUsed DECIMAL(18
, 2), OUT pResult INT, OUT pMessage VARCHAR(500))
BEGIN
    # Declare Variable
DECLARE
    errCode CHAR(5) DEFAULT '00000';
DECLARE
    errMsg TEXT;
DECLARE
    effRows INT;
    # Declare Handler Exception
DECLARE
    CONTINUE
    HANDLER FOR SQLEXCEPTION BEGIN GET DIAGNOSTICS CONDITION 1 errCode =
RETURNED_SQLSTATE, errMsg = MESSAGE_TEXT;
END;
    # Statement
    INSERT INTO
        `buyer` (
            BUYER_ID, NAME, EMAIL, COUNTRY_CODE, BUDGET, USED
        )
    VALUES (
            pBuyerID, pName, pEmail, pCountryCode, pBudget, pUsed
        );
    # Set Result
    IF errCode = '00000' THEN GET DIAGNOSTICS effRows = ROW_COUNT;
    SET pResult = effRows;
    SET pMessage = 'Insert Data Successfully';
    ELSE SET pResult = 0;
    SET
        pMessage = CONCAT(
            'Error, Code = ', errCode, ', Message = ', errMsg
        );
END
    IF;
END $$
DELIMITER ;
```

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
5	1		

- กำหนดคำสั่งเรียกใช้ SP insertBuyer นี้ พร้อมนำผลลัพธ์ไปใส่ใน Item 2

```
SET @p0='A005'; SET @p1='Fun Wipa'; SET @p2='fun.wipa@thaicreate.com';
SET @p3='TH'; SET @p4='100000'; SET @p5='0';
CALL `insertBuyer`(@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7);
SELECT @p6 AS `xResult`, @p7 AS `xMessage`;
```

- จงระบุภาพหน้าจอผลลัพธ์การเรียกใช้ SP insertBuyer

Item	ภาพหน้าจอผลลัพธ์
2	

การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
5	1		