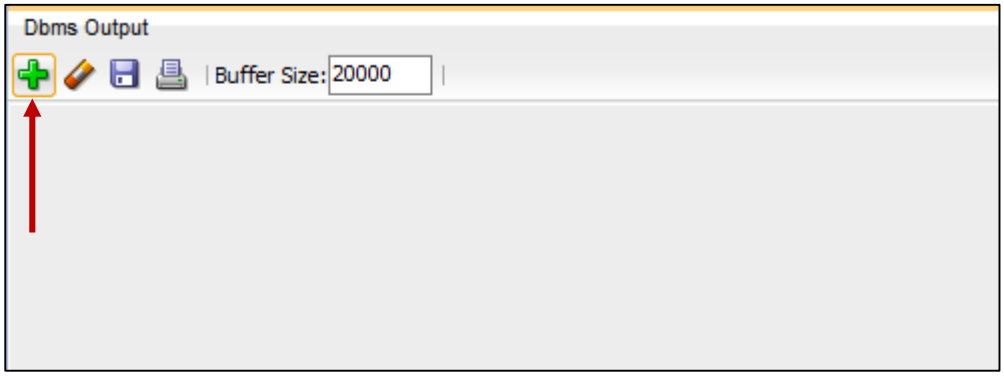
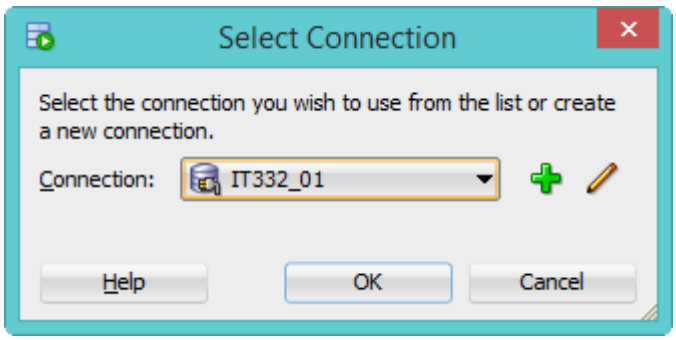
LAB 6: Intro to PL/SQL

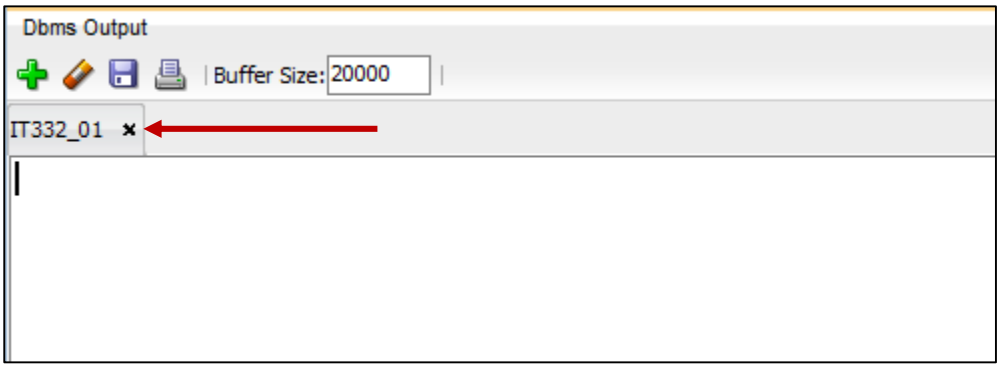
การใช้ DBMS Output

- เลือก View – > DBMS Output
- ในตำแหน่ง DBMS Output กดปุ่มบวก



- เลือก Connection เพื่อติดต่อฐานข้อมูลกดตกลง





ใช้ตารางจาก Lab1 จงอธิบายคำสั่งต่อไปนี้ว่าทำงานอะไรบ้าง พร้อมทั้ง บันทึกหน้าจอผลลัพธ์

1.
```
DECLARE
        M_PID NUMBER(3);
        M_PName  VARCHAR2(50);
        M_UnitPrice  NUMBER(5,2);
BEGIN
        SELECT ProductID, ProductName, UnitPrice
        INTO M_PID, M_PName, M_UnitPrice
        FROM L1Products WHERE ProductID = 4;
        -- FROM Products WHERE ProductID = &Pid;   //กรณีที่ต้องการให้แสดงหน้าต่างให้กรอกรหัสสินค้า
        DBMS_OUTPUT.PUT_LINE('PRODUCTID IS '|| M_PID);
        DBMS_OUTPUT.PUT_LINE('PRODUCT NAME IS ' || M_PName);
        DBMS_OUTPUT.PUT_LINE('UNITPRICE IS ' || M_UnitPrice);
EXCEPTION
        WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Invalid ProductID');
END;
```

2.
```
DECLARE
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_UnitPrice  L1Products.UnitPrice%TYPE;
BEGIN
        SELECT ProductID, ProductName, UnitPrice
        INTO M_PID, M_PName, M_UnitPrice
        FROM L1Products WHERE ProductID = 4;
```

```
        DBMS_OUTPUT.PUT_LINE('PRODUCTID IS '|| M_PID);
        DBMS_OUTPUT.PUT_LINE('PRODUCT NAME IS ' || M_PName);
        DBMS_OUTPUT.PUT_LINE('UNITPRICE IS ' || M_UnitPrice);
EXCEPTION
        WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Invalid ProductID');
END;
```

3.
```
DECLARE
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_Quantity  L1Products.Quantity%TYPE;
BEGIN
        M_PID := &Pid;
        M_Quantity := &Qty;
        INSERT INTO L1Order_Details Values(10255, M_PID, M_Quantity);
        UPDATE L1Products SET Quantity = Quantity – M_Quantity
                Where ProductID = M_PID;
        COMMIT;
        SELECT ProductID, ProductName, Quantity
        INTO M_PID, M_PName, M_Quantity
        FROM L1Products WHERE ProductID = M_PID;
        DBMS_OUTPUT.PUT_LINE('PRODUCTID IS '|| M_PID);
        DBMS_OUTPUT.PUT_LINE('PRODUCT NAME IS ' || M_PName);
        DBMS_OUTPUT.PUT_LINE('QUANTITY IS ' || M_Quantity);
END;
```

4. DECLARE

```
        ADD_RATE  NUMBER(5,2) := 1.20;
BEGIN
        DBMS_OUTPUT.PUT_LINE('1 ADD RATE IS '|| ADD_RATE);
        DECLARE
                ADD_RATE NUMBER(5,2) := 1.10;
        BEGIN
                UPDATE L1Products SET UnitPrice = UnitPrice * ADD_RATE
                WHERE ProductID = 3;
                COMMIT;
                DBMS_OUTPUT.PUT_LINE('2 ADD RATE IS '|| ADD_RATE);
        END;
        UPDATE L1Products SET UnitPrice = UnitPrice * ADD_RATE
        WHERE ProductID = 4;
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('3 ADD RATE IS '|| ADD_RATE);
END;
```

5. DECLARE

```
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_UnitPrice  L1Products.UnitPrice%TYPE;
BEGIN
        SELECT ProductID, ProductName, UnitPrice
        INTO M_PID, M_PName, M_UnitPrice
        FROM L1Products WHERE ProductID = &Pid;
        DBMS_OUTPUT.PUT_LINE(M_PName  || '  ' ||  M_UnitPrice);
```

```
        IF  M_UnitPrice  < 100 THEN
            DBMS_OUTPUT.PUT_LINE('Low price');
        ELSE
            DBMS_OUTPUT.PUT_LINE('High price');
        END IF;
    END;
```

6.
```
    DECLARE
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_UnitPrice  L1Products.UnitPrice%TYPE;
    BEGIN
        FOR I IN 1..5 LOOP
            SELECT ProductID, ProductName, UnitPrice
            INTO M_PID, M_PName, M_UnitPrice
            FROM L1Products WHERE ProductID = I;
            DBMS_OUTPUT.PUT_LINE(M_PID  || ' ' || M_PName  || ' ' ||  M_UnitPrice);
        END LOOP;
    END;
```

7.
```
    DECLARE
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_UnitPrice  L1Products.UnitPrice%TYPE;
        I NUMBER(2) := 1;
    BEGIN
        WHILE I <= 5 LOOP
            SELECT ProductID, ProductName, UnitPrice
```

```
            INTO M_PID, M_PName, M_UnitPrice
            FROM L1Products WHERE ProductID = I;
            DBMS_OUTPUT.PUT_LINE(M_PID  || ' ' || M_PName  || ' ' ||  M_UnitPrice);
            I := I + 1;
        END LOOP;
    END;
```

8.
```
    DECLARE
        M_PID L1Products.ProductID%TYPE;
        M_PName  L1Products.ProductName%TYPE;
        M_UnitPrice  L1Products.UnitPrice%TYPE;
        I NUMBER(2) := 1;
    BEGIN
        LOOP
            SELECT ProductID, ProductName, UnitPrice
            INTO M_PID, M_PName, M_UnitPrice
            FROM L1Products WHERE ProductID = I;
            DBMS_OUTPUT.PUT_LINE(M_PID  || ' ' || M_PName  || ' ' ||  M_UnitPrice);
            I := I + 1;
            EXIT WHEN I > 5;
        END LOOP;
    END;
```

9. จากตารางจาก LAB4 จงเขียนโปรแกรมคำนวณ GPA ของนักศึกษา โดยให้รับค่ารหัสนักศึกษาเข้ามา
   GPA = ผลรวมของ (หน่วยกิต x เกรด) / ผลรวมของหน่วยกิตทั้งหมด