الفصل الثانى

كتابة جمل PL / SQL

- فى نهاية كل بلوك فى لغة PL / SQL يجب وضع علامة (/) "salash" لتنفيذ ما سبقها من كتل (البلوكات) PL / SQL .

صيغ الملاحظات وجعل الكود لا يأخذ به من قبل المترجم :

1- لجعل سطر ما تعليق (ملاحظة) نضع فى أول السطر علامتين (- -) "dash" أو ناقص أو كتابة فى أول السطر كلمة (rem) .

2- لجعل أكثر من سطر تعليق أو ملاحظة نستخدم علامات فى البداية (*/) وعلامة (/*) فى نهاية التعليق او الملاحظة .

مثال :

```
DECLARE
...
  v_sal NUMBER (9,2);
BEGIN
  /* Compute the annual salary based on the
     monthly salary input from the user */
  v_sal := :g_monthly_sal * 12;
END;     -- This is the end of the block
```

وتستخدم هذه الملاحظات لعمل ملاحظات تعيين المبرمج على فهم البرنامج وكيفية عمله وذلك من الأمور المعروفة .

- الدول العاملة داخل لغة PL / SQL

SQL Functions in PL/SQL

- Available in procedural statements:
  - Single-row number
  - Single-row character
  - Data type conversion
  - Date
  - Timestamp
  - GREATEST and LEAST
  - Miscellaneous functions

} Same as in SQL

- Not available in procedural statements:
  - DECODE
  - Group functions

- كل الدوال " single row function" فى لغة SQL أيضا هنا ما عدا دالة " decode"
وبعض الدوال الأخرى
- مثال:

SQL Functions in PL/SQL: Examples

- Build the mailing list for a company.

```
v_mailing_address := v_name||CHR(10)||
                     v_address||CHR(10)||v_state||
                     CHR(10)||v_zip;
```

- Convert the employee name to lowercase.

```
v_ename       := LOWER(v_ename);
```

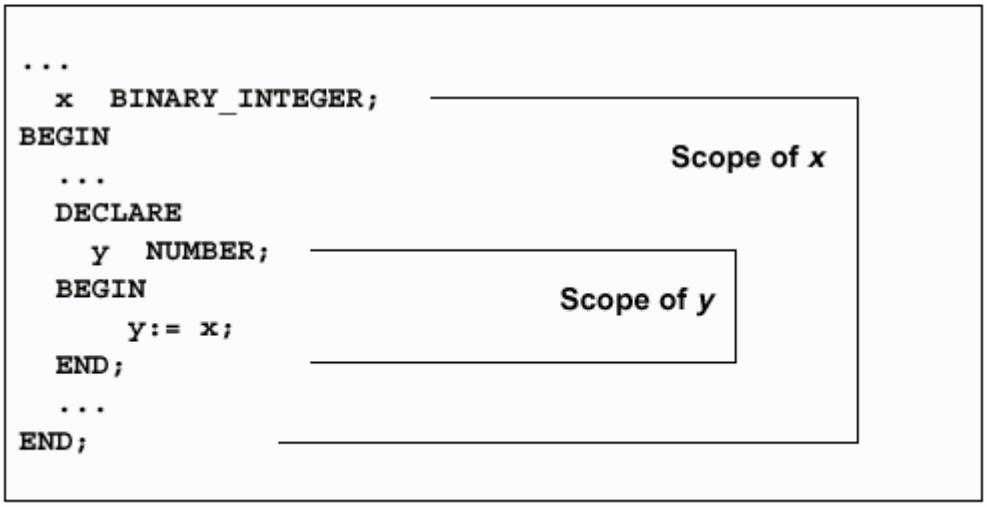
الكتل والبلوكات المتداخلة نطاق عمل المتغيرات :

- يمكن داخل PL / SQL تداخل أكثر من بلوك (كتلة) مع بعضهم البعض .

- نطاق عمل المتغيرات داخل البلوكات كالأتى :

Nested Blocks and Variable Scope

Example:



فى الشكل المتغير (x) يتم التعامل معه فى البلوك الخارجى والداخلى أما المتغير (y) نطاق عمله فقط فى البلوك الداخلى .

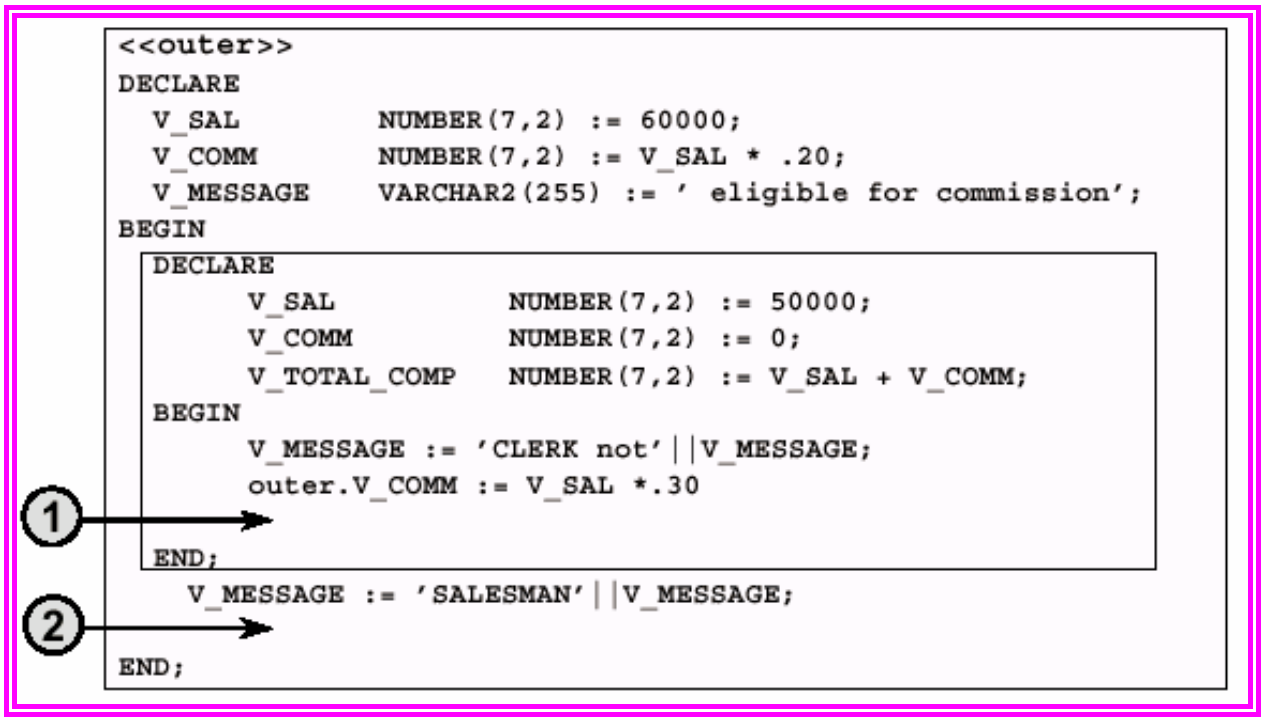
- **ومن ذلك يتضح أن – البلوك يرى البلوكات التى تحتويه فقط .**
- **البلوكات التى داخل هذا البلوك الخارجى لها لا يستطيع التعامل معها هذا البلوك الخارجى .**

وضع تسمية البلوكات

```
<<outer>>
   DECLARE
      birthdate DATE;
   BEGIN
         DECLARE
             birthdate DATE;
         BEGIN
            ...
            outer.birthdate :=
                          TO_DATE('03-AUG-1976',
                                  'DD-MON-YYYY');
         END;
    ....
    END;
```

يمكن تسمية البلوك وأستخدام هذه التسمية اذا حدث أن المتغيرات كانت بنفس الأسماء داخل بلوكات متعددة – وكذلك لأعطاء مزيد من التوضيح للكود لمعرفة كل بلوك بأسم معين يساعد على الفهم والأستدلال .

مثال :




```
<<outer>>
DECLARE
  V_SAL         NUMBER(7,2) := 60000;
  V_COMM        NUMBER(7,2) := V_SAL * .20;
  V_MESSAGE     VARCHAR2(255) := ' eligible for commission';
BEGIN
  DECLARE
        V_SAL           NUMBER(7,2) := 50000;
        V_COMM          NUMBER(7,2) := 0;
        V_TOTAL_COMP    NUMBER(7,2) := V_SAL + V_COMM;
  BEGIN
        V_MESSAGE := 'CLERK not'||V_MESSAGE;
        outer.V_COMM := V_SAL *.30
  END;
    V_MESSAGE := 'SALESMAN'||V_MESSAGE;
END;
```

فى الموضوع 1- يتم أدخال قيمة المعادلة فى v_com الخارجى

فى الموضع 2- يتم وضع قيمة جديدة v_message وهذا المتغير فى البلوك (outer) العمليات .

- **وضع العمليات الحساسية والمنطقية فى PL / SQL :**

مثل SQL تماما بنفس الفكر والترتيب .

أمثلة :

1- لعمل عداد رقمى ; v:=v+1

2 - لعمل مؤشر منطقى ; v_flag := (v1 = v2)

فى المثال 2 : معناه اذا كانت V1 يساوى V2 فأن المتغير v_flag قيمة true لأنه من نوع Boolean واذا لم يتساوى v1 ، يساوى v2 فان المتغير v_flag يأخذ False لأنه من نوع v_flag ويأخذ v_flag واذا كان أحدهما null يأخذ القيمة null .