INTRODUCTION TO PL/SQL

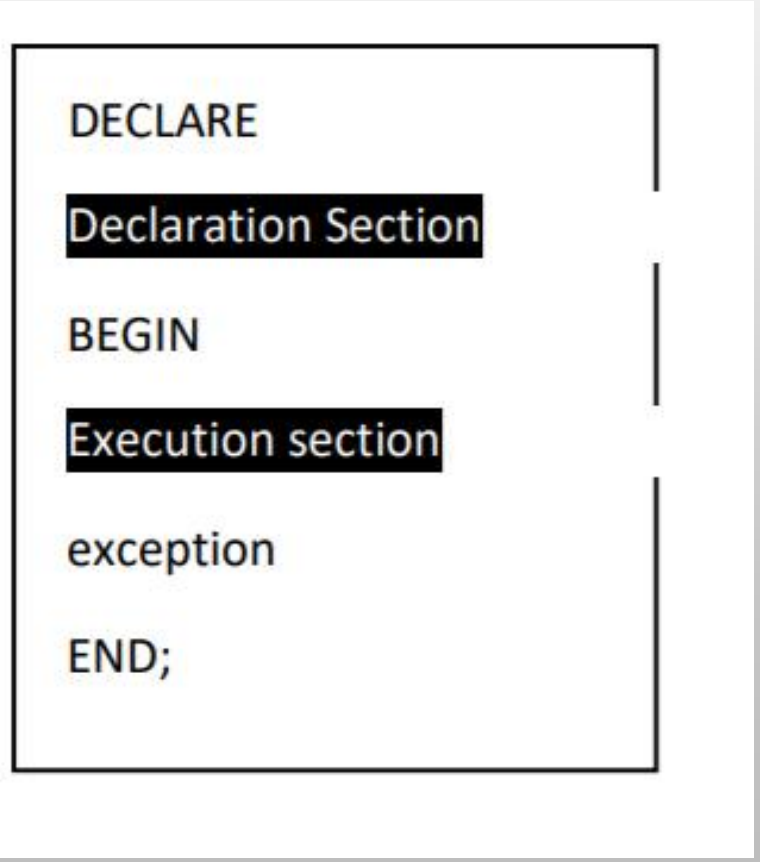
ماهي PL/SQL

- PL/SQL اختصار ل PROCEDURAL LANGUAGE EXTENSION OF SQL
  أي انها لغة إجرائية وهي امتداد ل SQL
- تم تطويرها بواسطة ORACLE في أوائل التسعينيات لتعزيز قدرات SQL

خصائص PL/SQL

- استخدام SQL داخل وحدة PL/SQL
- التعامل مع الأخطاء
- يدعم البرمجة الشيئية
- يدعم تطبيقات الويب

وحدة PL/SQL
(PL/SQL BLOCK)



الجزء التعريفي (اختياري): للمتغيرات والمؤشرات (CURSORS) و البرامج الفرعية

الجزء التنفيذي (الزامي): يأتي بين BEGIN و END وهو الجزء القابل للتنفيذ ويتكون من أوامر PL/SQL

جزء الاستثناءات (اختياري): في هذا الجزء يتم التعامل مع الأخطاء ومعالجتها

امر الطباعة في PL/SQL

```
BEGIN
DBMS_OUTPUT.PUT_LINE('WELCOME TO PL SQL COURSE');
 END;
```

كتابة الملاحظات (COMMENTS)

```
-- one-line comment
/*
multiple
lines
comments
*/
```

لكتابة ملاحظة في سطر واحد نكتب - - في بداية السطر

لكتابة ملاحظة تتكون من عدة اسطر نكتب */ في بداية الملاحظة ونكتب /* في نهاية الملاحظة

المتغيرات (VARIABLES)

الصيغة العامة لتعريف المتغيرات (GENERAL SYNTAX)

```
variable_name [CONSTANT] datatype [NOT NULL: = value];
```

يشترط للمتغير ان يبدأ بحرف، 30 حرف كحد اقصى، يمكن ان يحتوي على رقم أو _ أو $

(EXAMPLE 1) 1 مثال

```
Declare
    msg varchar2(30):='welcome to pl sql course';
Begin
    dbms_output.put_line(msg);
end;
```

(EXAMPLE 2) 2 مثال

```
Declare
    msg varchar2(30);
begin
    msg:='welcome to pl sql course';
    dbms_output.put_line(msg);
end;
```

مثال على متغير NOT NULL

```
Declare
    n number(2) not null:=5;
begin
    n:=10;
    dbms_output.put_line(n);
end;
```

ملاحظة:

عند تعريف متغير على انه NOT NULL فانه **يجب** اعطاءه قيمة ابتدائية.

مثال على متغير CONSTANT

```
declare
    n constant number(2):=5;
begin
    dbms_output.put_line(n);
end;
```

ملاحظة:

عند تعريف متغير على انه Constant فانه **يجب** اعطاءه قيمة ابتدائية ولا يمكن تغيير قيمته لاحقا

ملاحظات

نستخدم علامة =: لاسناد قيم لمتغيرات
نستخدم علامة = للمقارنة
يمكن استخدم كلمة Default لاسناد قيمة لمتغير بدلا من =:

default ≡ =:

أنواع البيانات:

نوع البيانات	الاستخدام
NUMBER(precision,scale)	رقم
VARCHAR2(max length) / CHAR(max length)	نص
Boolean	قيمة منطقية
DATE	التاريخ والوقت
LOB	صور – صوت - فيديو

العمليات الحسابية

```
DECLARE
  A NUMBER := 30;
  B NUMBER := 20;
  C NUMBER;
BEGIN
  DBMS_OUTPUT.PUT_LINE('A = '||A||' B = '||B);
  C := A + B;
  DBMS_OUTPUT.PUT_LINE('A + B = '||C);
  C := A - B;
  DBMS_OUTPUT.PUT_LINE('A - B = '||C);
  C := A * B;
  DBMS_OUTPUT.PUT_LINE('A * B = '||C);
  C := A / B;
  DBMS_OUTPUT.PUT_LINE('A / B = '||C);
END;
```

OUTPUT:

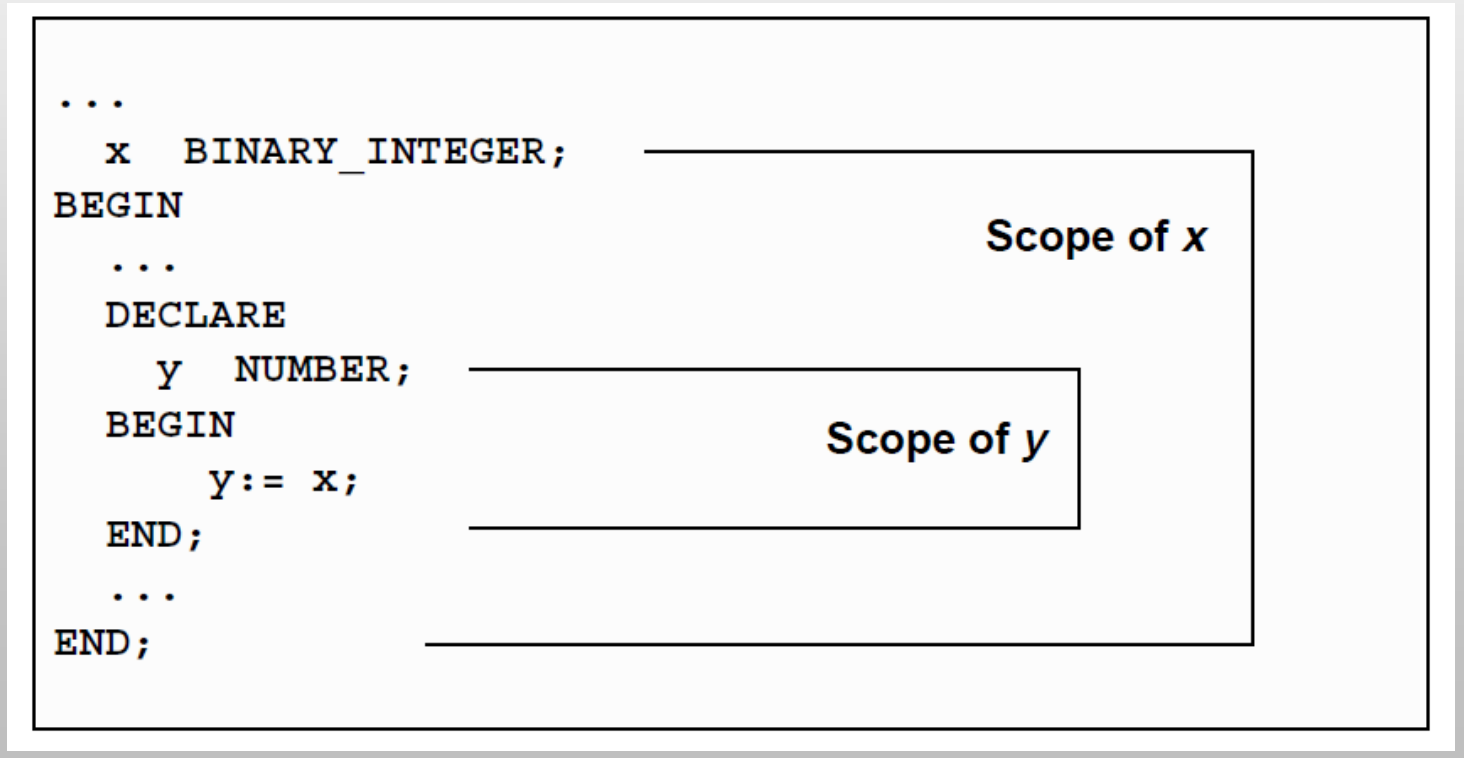
A = 30 B = 20
A + B = 50
A - B = 10
A * B = 600
A / B = 1.5

نستخدم الرمز || للربط بين النصوص والمتغيرات

NESTED BLOCK IN PL/SQL

```
Declare                                  Parent Block
...
Begin
    Declare                              Child Block
    ...
    Begin
    ...
    End;
End;
```

VARIABLE SCOPE IN PL/SQL
نطاق المتغيرات في ال PL/SQL



ويكون نطاق المتغيرات داخل الوحدات البرمجية على النحو التالي:

نطاق (y) يتم التعرف عليه في الوحدة البرمجية الداخلية فقط وعند استخدامها في الوحدة الخارجية
سيظهر خطأ،

أما المتغير (x) فيتم التعرف عليه في الوحدة الداخلية والخارجية.

VARIABLE SCOPE IN PL/SQL
نطاق المتغيرات في ال PL/SQL
EXAMPLE
مثال 1

```
DECLARE
-- Global variables
num1 number := 95;
num2 number := 85;
BEGIN
dbms_output.put_line('Outer Variable num1: ' || num1);
dbms_output.put_line('Outer Variable num2: ' || num2);
DECLARE
-- Local variables
num1 number := 195;
num2 number := 185;
BEGIN
dbms_output.put_line('Inner Variable num1: ' || num1);
dbms_output.put_line('Inner Variable num2: ' || num2);
END;
END;
```

OUTPUT :

```
Outer Variable num1: 95
Outer Variable num2: 85
Inner Variable num1: 195
Inner Variable num2: 185
```

ملاحظة:
المتغير والمعرف في الوحدة الخارجية يمكن استخدامه في الوحدة الداخلية.
أما عند محاولة استخدام متغير تم تعريفه في الوحدة الداخلية في الوحدة الخارجية سيظهر الخطأ.

VARIABLE SCOPE IN PL/SQL
نطاق المتغيرات في ال PL/SQL
EXAMPLE
مثال 2

```
DECLARE
  V_SALARY NUMBER := 3000;
BEGIN
  DECLARE
    V_SALARY NUMBER := 1500;
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Inner block: My salary is:
' || V_SALARY);
  END;
 DBMS_OUTPUT.PUT_LINE('Outer block: My salary is: '
|| V_SALARY);
END;
```

OUTPUT:

```
Inner block: My salary is: 1500
Outer block: My salary is: 3000
```

يمكن القول بان كل وحدة تتعامل مع المتغيرات الخاصة بها في حال تشابه أسماء المتغيرات

تمارين 1

اكتب برنامج يقوم بقراءة قطر دائرة وحساب مساحتها مع العلم ان:

مساحة الدائرة = $\pi * r^2$

بحيث: r هو قطر الدائرة و $\pi = 3.14$

أضف بعض الملاحظات على البرنامج السابق مثل وظيفة هذه البرنامج وأسمك ورقمك التدريبي؟

تمارين 2

اكتب برنامج يقوم بالتالي:

- تعريف 3 متغيرات لتخزين بيانات طالب (رقم الطالب، اسمه، عمره) مع اعطاءها قيم ابتدائية (100، احمد، 20)
- طباعة بيانات الطالب
- تغيير اسم الطالب الى محمد وعمره الى 19
- طباعة بيانات الطالب بعد التعديل