

## المحاضرة الاولى

### مدخل الى لغة Oracle PL/SQL

تعتبر لغة ال PL / SQL وهي اختصار ل **Procedural Language** امتداد للغة SQL قامت شركة الأوراكل بتطويرها لتساعد لغة SQL للتعامل كلغة إجرائية ، وتمكننا ال PL / SQL من تعريف والتعامل مع :-

- الإجراءات Procedure
- الدوال Function
- التحكمات الشرطية IF – Condition
- التكرار Loops
- متغيرات الذاكرة Memory Variables

مستقبلاً عندما تحاول تطوير برامج باستخدام أحد منتجات الأوراكل مثل **Developer 2000** أو أي من المنتجات الأخرى مثل **Oracle Forms or Oracle Reports** يجب عليك أن تكون ملم بلغة PL / SQL لأن هذه المنتجات تحتاجها في التحكمات الداخلية للبرنامج. والبرنامج في ال PL / SQL يتكون من :-

1. إجراءات Procedures

2. دوال Functions

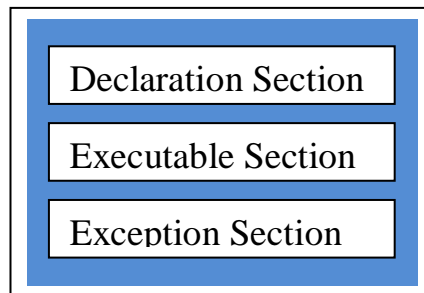
3. وحدات نمطية Blocks

توجد وحدات نمطية تسمى **Anonymous Blocks** وتعني الوحدات النمطية الغير مسماة أي أنه لا يتم إعطائها اسم ولا يتم تعريفها كالدوال والإجراءات أو الحزم التجميعية **Package** .

مكونات الوحدة النمطية PL / SQL Block :-

تتكون الوحدة النمطية PL / SQL Block من قسم التعريف **Declaration** وقسم الجزء القابل للتنفيذ أو ما يعرف بجسم الإجراء **Executable** وبداخله يوجد قسم للتعامل مع الأخطاء التي قد ترد في البرنامج **Exception** ويمكن أن نوضحها بالرسم التالي:-

PL / SQL Block



الصيغة العامة للوحدة النمطية :-

**Syntax: -**

**Declare**

*Variables declarations ;*

**BEGIN**

*Some program codes ;*

**BEGIN**

*Code in nested block ;*

**END ;**

*More programs codes ;*

**EXCEPTIONS**

*Handling errors ;*

**END ;**

<p>هام :-</p> <p>لتنفيذ الوحدة النمطية نقوم بعد سطر END الأخير بكتابة علامة الشرطة المائلة / ثم</p> <p>نضغط Enter</p>	<input checked="" type="checkbox"/>
---	-------------------------------------

قسم التعريف Declaration Section :-

قسم التعريف في الوحدة النمطية للـ **PL / SQL** قسم اختياري أي يمكن تجاهله ، لكننا نقول أنه يجب عليك تعريف كل المتغيرات والثوابت التي سوف تتعامل معها داخل جسم الوحدة النمطية. ولتعريف المتغيرات أو الثوابت يجب أن نعطيها اسم ثم نحدد نوع البيانات **Data Type** ويمكننا أيضاً أن نسند لها قيمة ابتدائية . ويوفر لنا الأوراكل أنواع بيانات إضافية تستخدم مع الـ **PL / SQL** وهي كالتالي :-

نوع البيانات	الوصف
<b>BOOLEAN</b>	منطقي ويقبل <b>True</b> أو <b>False</b> كما يمكن أن يكون <b>Null</b>
<b>BINARY_INTEGER</b>	يقبل الأرقام السالبة والموجبة في المدى بين <b>-2147483647</b> إلى <b>2147483647</b>
<b>NATURAL</b>	هو جزء من <b>Binary_integer</b> في المدى بين <b>0</b> و <b>2147483647</b>
<b>POSITIVE</b>	هو جزء من <b>Binary_integer</b> في المدى بين <b>1</b> و <b>2147483647</b>
<b>%TYPE</b>	هو نوع من التسهيل في تعريف نوع البيانات بحيث يكون شبيه لنوع بيانات حقل في جدول معين دون أن نكون نعرف ما هو نوع بيانات الحقل المعني بالضبط.
<b>%ROWTYPE</b>	يمكننا من تعريف متغير يتكون من مجموعة من المتغيرات تشكل سجلاً كاملاً في جدول معين بحيث تأخذ مجموعة المتغيرات الداخلية أسماء الحقول ونوع البيانات المعرفة في سجل الجدول المعين. مع ملاحظة أنه لا يمكن إسناد قيمة متغير من هذا النوع إلى آخر من نفس النوع إلا إذا كانا معرفين من نفس الجدول ( حتى وإن تشابهت الجداول ).

**مثال: استخدام نوع البيانات %TYPE :-**

Mem\_var-1 emp.ename%TYPE

وهذا المثال يعني أن المتغير **mem\_var-1** يساوي في نوعه نوع البيانات في حقل **ename** بجدول **emp**. ويجب أن نلاحظ أنه إذا كان في الجدول معرف على أنه **Not Null** فإننا لا نستطيع أن نسند للمتغير القيمة **Null**.

### مثال: استخدام نوع البيانات %ROWTYPE :-

mem\_var-2 emp%ROWTYPE

هذا المثال يعني أن المتغير **mem\_var-2** مساوي لسجل كامل في جدول **emp** ولحفظ قيمة داخلية في هذا المتغير ( اسم العامل مثلاً ) نكتب كالتالي :-

mem\_var-2.ename := 'RXR'

استخدام	Select INTO	مع المتغيرات :-
---------	-------------	-----------------

كما سبق فقد عرفنا أنه لتعيين قيمة لمتغير نستخدم علامة الإسناد ( := ) ، ولكن توجد طريقة أخرى وهي استخدام جملة **Select Into** وهي تعتمد على نوع المتغير، فإذا كان المتغير عادي يجب أن ترجع جملة **Select** قيمة واحدة ( عادة تستخدم مع الدوال المجمعة ). أما إذا كان المتغير من النوع **%Rowtype** فيجب أن ترجع جملة **Select** سجلاً كاملاً كما سيتضح بالأمثلة التالية.

### مثال: استخدام Select Into :-

```
SQL> declare
2 sal_avg number(8,2);
3 my_rec my_emp%rowtype;
4 begin
5 dbms_output.enable;
6 select avg(sal) into sal_avg
7 from my_emp;
8 --*****
9 select * into my_rec
10 from my_emp
11 where
12 empno = 7369;
13 --*****
14 dbms_output.put_line('The average of Salary is :- '|| sal_avg);
15 dbms_output.put_line('The employee name is :- '|| my_rec.ename);
16 end;
17 /
```

### الناتج

The average of Salary is :- 3571.88  
The employee name is :- SMITH

PL/SQL procedure successfully completed.