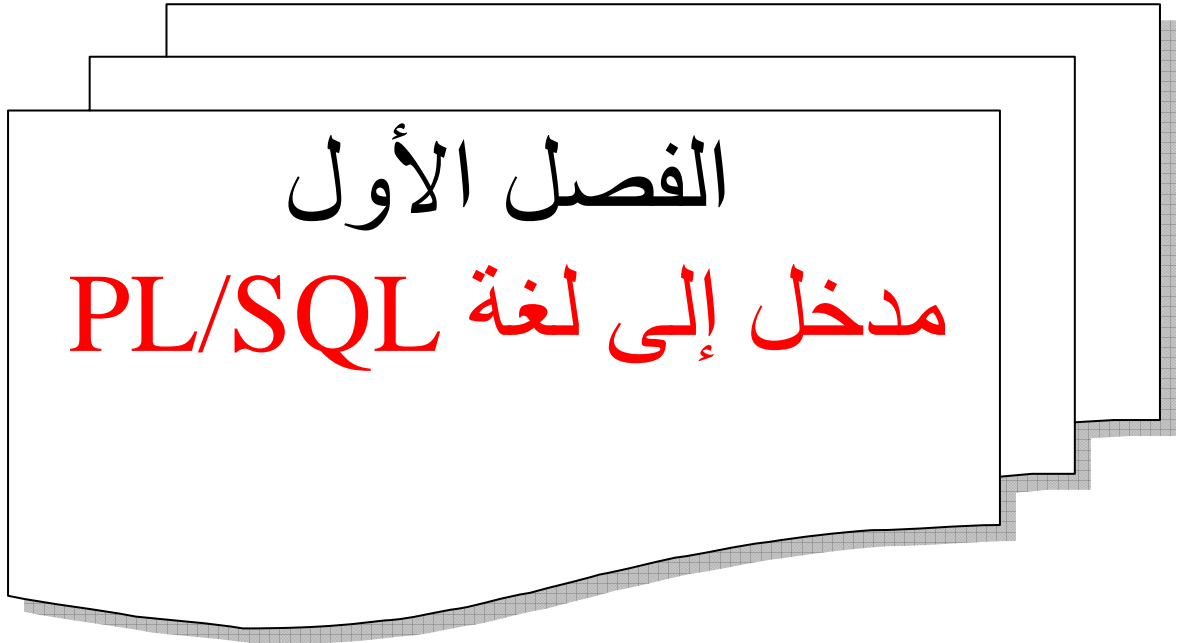


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

محتويات الكتاب :

الموضوع	م
الفصل الأول : مدخل إلى لغة PL/SQL	
• مقدمة	1
• مميزات اللغة	2
• ماذا أحتاج لتعلم اللغة	3
• كتابة البرنامج الأول	4
• طرق تنفيذ البرنامج	5
• مكونات كتلة الترميز (أجزاء البرنامج)	6
• الكلمة المفتاحية (GOTO) القفز الغير المشروط	7
• التعليقات	8
• تمارين على الفصل الأول	9
الفصل الثاني : المتغيرات والثوابت	
• تعريف المتغيرات	1
• أهمية المتغيرات	2
• استخدامات المتغيرات	3
• شروط تسمية المتغيرات	4
• أنواع المتغيرات في اللغة	5
• كيفية تعريف المتغيرات في اللغة	
✓ البيانات الرقمية	
✓ البيانات النصية	
✓ البيانات الأخرى	6
✓ القيود	
✓ الأداة %TYPE	
✓ الأداة %ROWTYPE	
• أمثلة على الفصل الثاني	7
• تمارين على الفصل الثاني	8



المقدمة :

تعتبر اللغة PL/SQL هي لغة البرمجة في نظام أوراكل ، وهي أداة برمجية قوية ، وكلمة PL/SQL اختصاراً لـ (Procedure Language/Structure Query Language)، وتستخدم هذه اللغة لتجهيز نظام Oracle عن طريقة معالجة التسجيلات ، وتستخدم أيضاً في أدوات التطوير المنتجة من قبل شركة Oracle ، وهي تعتبر أساس برنامج بناء النماذج Oracle Forms و Oracle Reports . وتستخدم هذه اللغة أيضاً لتعريف نموذج ما ، والقيام ببعض الحسابات الخاصة في تقرير ما ، ومن أجل التسجيلات .

مزايا لغة PL/SQL :

1- **التكامل** : وهذه اللغة دور أساسي بين أجزاء وأدوات أوراكل حيث يكتب بها أكواد (Forms) ويتم بها برمجة أجزاء وأدوات الأوراكل .

2- **تحسين الأداء** : حيث يمكن لـ PL/SQL تحسين أداء التطبيقات وذلك من خلال :

أ- تجميع جمل SQL معاً في بلوك واحد (كتلة واحدة) وإرسالهما إلى خادم (Data Base) لتنفيذها دفعة واحدة مما يؤدي إلى ارتفاع مستوى الأداء عامة .

ب- يمكن لـ PL/SQL العمل داخل أي جزء من أجزاء وأدوات Oracle وبذلك يضيف قوة المعالجة الإجرائية إلى هذه الأدوات (Oracle Forms) ، (Oracle Reports) ، ، مما يؤدي إلى تحسين مستوى الأداء .

3- **تطوير البرنامج Modularized** وذلك بـ :

أ- تجميع منطقي للبيانات داخل كتل (Blocks) البرنامج .

ب- الكتل المتداخلة (Nested Blocks) تتيح العديد من المزايا .

ت- إتاحة تقسيم المشاكل المعقدة إلى مجموعة أبسط من المشاكل يمكن حلها ببساطة .

ث- الاستفادة من خبرات وأكواد سابقة بجمعها في شكل مكتبات (Libraries) يمكن الاستفادة منها بين أدوات Oracle المختلفة .

ج- يمكن تنفيذ كود PL/SQL من أي أداة من أدوات Oracle المختلفة .

ح- يمكن تعريف المتغيرات والمتحولات (Variables) التي تستقبل العديد من أنواع البيانات المختلفة مثل الأرقام والنصوص والصور والفيديو والبيانات المركبة الخ .

خ- وتحتوي أيضاً على المميزات الأخرى مثل أوامر التكرار والتحكم في سير البرنامج ومعالجة الأخطاء والإستثناءات وووووو....الخ .

ماذا أحتاج لتعلم هذه اللغة :

لا أحتاج إلى شيء إضافي ما دام عندي برنامج SQL*Plus منزل على جهازي من قبل، فبرنامج SQL *Plus يفي بالغرض لتعلم وكتابة هذه اللغة ، الذي أحتاج فقط إلى المعرفة والقدرة على كتابة أوامر SQL وإني اجتزت هذه المادة بشكل جيد .

كتابة البرنامج الأول بلغة PL/SQL :

يمكن كتابة برامج PL/SQL وتنفيذها من محث SQL ضمن برنامج SQL *Plus . ولكن، وبسبب درجة تعقيد وطول البرامج، لا تعتبر آلية مفضلة للإستخدام . إذ يفضل تطوير البرنامج في أي محرر نصوص مثل المفكرة Notepad أو بانى الإجراءات في نظام Oracle . وبعد ذلك يمكن استدعاء ملفات البرامج ضمن SQL *Plus .

هيا نكتب برنامجنا الأول بلغة PL/SQL ، فلنفتح برنامج SQL *Plus وكتابة البرنامج التالي ، وهذا البرنامج لطباعة رسالة (Hello) :

القائمة 1-1

```
SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line ('Hello');
  3  end;
  4  /
Hello
```

PL/SQL تم بنجاح إجراء

فالبرنامج السابق يتألف من كتلة ترميز بلغة PL/SQL تحتوي على تعليمة واحدة . تُظهر الرسالة

(Hello) .

بعض الخصائص الهامة للبرنامج السابق :

- يبدأ بكلمة (Begin) وينتهي بكلمة (End) .
- أمر الطباعة في لغة PL/SQL هر (DBMS_OUTPUT.PUT_LINE) .
- تتكون كتل ترميز PL/SQL من تعليمات ، وكل تعليمة تنتهي بفاصلة منقوطة .
- يتم وضع (/) في نهاية كتلة ترميز PL/SQL لتنفيذ تعليمات كتلة الترميز .
- تعتبر الكلمة المفتاحية (END) هي الكلمة المفتاحية الوحيدة في كتلة ترميز PL/SQL التي تنتهي بفاصلة منقوطة .

ملحوظة مهمة :

لا حظ أن أول سطر في البرنامج السابق كلمة (SET SERVEROUTPUT ON) ، هي تعليمة SET التابعة لبرنامج *Plus SQL المسؤولة بإظهار الرسائل المرسله من قبل الإجراءات الموجودة في الحزمة البرمجية DBMS_OUTPUT ، يجب وضعها مرة واحدة فقط ضمن جلسة العمل ، فإذا لم تكن هناك استخدام للحزمة البرمجية DBMS_OUTPUT فلا داعي إلى استخدام هذه التعليمة .

أعتقد إلى هنا ما توصلنا إلى حاجة مرتبة مفهومة .. فأنا لست مطالباً بأي حاجة إلى هنا إلا جملة SET_SERVEROUTPUT ON : يجب استخدامها مرة واحدة فقط ضمن جلسة العمل إن كان هناك استخدام جملة DBMS_OUTPUT.PUT_LINE المسؤولة لتنفيذ أمر ما .

طرق تنفيذ برنامج PL/SQL :

- 1- إذا كانت الكتلة البرمجية لترميز PL/SQL مكتوب داخل برنامج *Plus SQL فيمكن تنفيذها بوضع (/) بعد نهاية كتلة الترميز مباشرة .
- 2- وإذا كانت الكتلة البرمجية لترميز PL/SQL في ملف خارجي فيمكن تنفيذها من محث SQL *Plus باستخدام الكلمة المفتاحية (START) أو الرمز (@) . وهذا النوع من الكتل البرمجية تسمى (إجراءات كتل مجهولة) . والشكل العام لاستخدام هذه الكلمة المفتاحية :

الامتداد. اسم الملف | مسار الملف START

```
SQL> START C:\1.SQL
Hello
PL/SQL تم بنجاح إجراء
SQL>
```

وبعد الانتهاء من تنفيذ كتلة ترميز PL/SQL سيظهر نظام Oracle الرسالة التالية:

PL/SQL تم نجاح إجراء

أو

PL/SQL procedure successfully completed

والتي تخبرنا بأنه تم تنفيذ البرنامج بنجاح ، أما إذا حدث أي خطأ فإن نظام Oracle سيصدر رسالة خطأ .

من هنا نبدأ الجد-----علينا الصبر والمثابرة

مكونات كتلة الترميز (أجزاء البرنامج) :

لكتابة أي برنامج بلغة PL/SQL يجب علينا أن نعرف أن ترميز كتلة PL/SQL يتكون من أربعة

مقاطع وهي بالترتيب :

1- الترويسة : وهو مقطع اختياري في كتلة الترميز . ويستخدم لتحديد نوع كتلة الترميز واسمها . وأنواع

كتل الترميز هي : anonymous procedure أي إجرائية مجهولة الاسم ، و named procedure

أي إجرائية لها اسم ، و function أي تابع . وتستخدم الترويسة مع النوعين الأخيرين فقط .

2- التصريح : وهو أيضاً مقطع اختياري في كتلة الترميز . ويحتوي على أسماء الأغراض المحلية التي سيتم

استخدامها في كتلة الترميز . وتتضمن المتغيرات وتعريف المؤشرات والاستثناءات ، ويبدأ هذا المقطع

بالكلمة الافتتاحية (DECLARE) .

3- التنفيذ : وهو المقطع الإجمالي الوحيد في كتلة الترميز . ويحتوي كل التعليمات التي سيتم تنفيذها ،

والتي تتألف من تعليمات DML ، إجرائيات (كتل ترميز PL/SQL) ، توابع (كتل ترميز PL/SQL) تعيد

قيمة ما، وبرامج جزئية مسبقة البناء . ويبدأ هذا المقطع بالكلمة الافتتاحية (BEGIN) .

4- الاستثناءات : وهو مقطع اختياري . ويستخدم لالتقاط ومعالجة أي خطأ يحدث أثناء التعليمات الموجودة

في المقطع التنفيذي . ويبدأ هذا المقطع بالكلمة المفتاحية (EXCEPTION) .

ملحوظة :

* يلي ذلك كله الكلمة المفتاحية (END) لإنهاء كتلة الترميز ، وهي الكلمة المفتاحية الوحيدة التي

تنتهي بفاصلة منقوطة .

* ولتنفيذ كتلة الترميز يتم وضع الرمز (/) slash بعد نهاية كتلة الترميز .

والشكل العام لكثلة الترميز :

DECLARE

(جزء التصريح والتعريف) هنا يتم وضع المتغيرات وتعريف المؤشرات والإستثناءات

BEGIN

(جزء التنفيذ) هنا يتم وضع التعليمات التي سيتم تنفيذها

EXCEPTION

(جزء الاستثناءات) هنا يتم وضع الاستثناءات

END; (نهاية الكثلة البرمجية) هذه الكلمة يجب وضعها في نهاية كثلة الترميز لإنهاء الكثلة

/ (تنفيذ الكثلة البرمجية) وهذا الرمز يتم وضعه لتنفيذ كثلة الترميز

سوف نأخذ المثال التالي :

كثلة برمجية يسترجع اسم الموظف الذي يحمل الرقم (7782) من جدول الموظفين (EMP) ، ثم طباعته، وإذا كان هناك حدوث أي خطأ على التعليمات التنفيذية سيظهر رسالة (ERROR OCCURED) .

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  X VARCHAR2(30);      تعريف متغير من النوع الحرفي طوله 30
  3  BEGIN                حرفاً
  4  SELECT ENAME
  5  INTO X                هنا جملة الاستعلام عن اسم الموظف الذي يحمل الرقم 7782
  6  FROM EMP              وطباعته
  7  WHERE EMPNO = 7782;
  8  DBMS_OUTPUT.PUT_LINE (X);
  9  EXCEPTION
  10 WHEN OTHERS THEN
  11 DBMS_OUTPUT.PUT_LINE ('ERROR OCCURED');  تعليمات الإستثناء
  12 END;
  13 /
CLARK هذا اسم الموظف المسترجع من جملة الاستعلام
PL/SQL تم بنجاح إجراء
SQL>
```

الكلمة المفتاحية (GOTO) القفز الغير المشروط :

تستخدم الكلمة المفتاحية (GOTO) لإجراء قفز غير مشروط والانتقال من مقطع من كتلة الترميز

إلى مقطع آخر. ولاستخدام هذه الكلمة المفتاحية هناك قواعد وتعليمات يجب إتباعها وهي بالترتيب :

- 1- تعريف العناوين : وذلك بوضع إشارتي (<<) قبل اسم العنوان وإشارتي (>>) بعد اسم العنوان .
- 2- يتم استخدام الكلمة المفتاحية (GOTO) ثم وضع اسم العنوان المراد القفز إلى هناك بعد (GOTO) .

ملحوظة :

س/ ما المقصود بتعريف العناوين الذي أشرنا إليه في الفقرة رقم واحد ؟

ج/ تعرف العناوين بأنها وسائل تستخدم لوضع علامات لمقاطع كتلة الترميز .

سنأخذ المثال التالي :

كتلة برمجية تستخدم الكلمة المفتاحية (GOTO) لقفز غير مشروط من مقطع إلى مقطع آخر ، تحتوي ثلاث

تعليمات تنفيذية ، يتم تنفيذ التعليمة الوسطى ثم الأولى ثم التعليمة الأخيرة :

```
SQL> Begin <<b_lable>>
2 goto middle;
3
4 <<top>>
5 DBMS_OUTPUT.PUT_LINE ('Top Statement');
6 goto bottom;
7
8 <<middle>>
9 DBMS_OUTPUT.PUT_LINE ('Middle Statement');
10 goto top;
11
12 <<bottom>>
13 DBMS_OUTPUT.PUT_LINE ('Bottom Statement');
14
15 END;
16 /
Middle Statement
Top Statement
Bottom Statement
```

تم بنجاح إجراء

SQL> |

الشرح :

عند بداية تنفيذ البرنامج يحصل الجملة التالية (goto middle) فيذهب إلى المقطع (middle) وينفذ التعليمات ثم يحصل الجملة التالية (goto top) فيذهب إلى المقطع (top) وينفذ التعليمات ثم يحصل الجملة التالية (goto bottom) فيذهب إلى المقطع (bottom) ثم يحصل كلمة (end) فيتهي البرنامج .

ملحوظة :

يوصي العديد من الخبراء في هذا المجال بعدم استخدام التعليمة (GOTO) ، فهي تجعل فقدان

السيطرة على التطبيقات أمراً سهلاً . وهي للاستخدام في بعض الحالات النادرة التي يمكن أن تجعل البرنامج

أسهل وأبسط .

التعليقات :

كأي لغات البرمجة الأخرى يمكن إدخال التعليقات ضمن كتلة الترميز ، ويوضع المبرمج مثل هذه التعليقات لتسهيل أمره وقت مراجعته وتطويره للبرنامج بعد فترة زمنية ، ولا تؤثر هذه التعليقات في حجم البرنامج أبداً . ويوجد أداتان لكتابة هذه التعليقات :

- 1- لكتابة تعليق سطرًا واحدًا : يتم وضع إشارتي ناقص (--) في بداية السطر الذي نرغب بوضعه كتعليق .
 - 2- ولكتابة تعليق عدة أسطر : يتم وضع الرمز (/*) في بداية التعليق ، ووضع الرمز (* /) في نهاية التعليق .
-

تمارين على الفصل الأول

السؤال الأول :

ضع علامة صح () أما العبارة الصحيحة وعلامة () خطأ أمام العبارة الخاطئة :

- 1- لغة PL/SQL هي أساس البرمجة في نظام أوراكل . ()
- 2- من مزايا لغة PL/SQL التكامل فقط . ()
- 3- لا يمكن تنفيذ أكواد PL/SQL من أي أداة من أدوات أوراكل المختلفة . ()
- 4- يمكن كتابة أكواد PL/SQL في محرر نصوص ثم استدعاؤه من محث SQL *Plus . ()
- 5- الكلمة المفتاحية الوحيدة التي تنتهي بفاصلة منقوطة هي (End) . ()
- 6- ولتنفيذ كتلة الترميز يتم وضع الرمز (*) star بعد نهاية كتلة الترميز . ()
- 7- المقطع الإجباري الوحيد هو المقطع التنفيذي (Begin) . ()
- 8- الكلمة (GOTO) تشير إلى القفز المشروط . ()

السؤال الثاني :

أجب عما يأتي :

1- إذا كان ملف كتلة الترميز مكتوب في ملف خارجي . فكيف يتم استدعاؤه من محث SQL *Plus ؟

2- ما هو الرمز المستخدم في كتابة تعليق لسطر واحد فقط ؟ وما هو الرمز المستخدم في كتابة عدة أسطر ؟

3- أكتب كتلة برمجية يتم فيه طباعة النص التالي (Welcome To PL/SQL) ؟

الفصل الثاني

المتغيرات والثوابت

المتغيرات والثوابت :

تعريف المتغيرات :

إن المتغير هو موقع تخزين - من أجل حفظ قيمة - يمكن أن يتم إسناد قيمة إليه ، وسمي متغير لأنه يمكن أن تتغير قيمته أثناء تنفيذ البرنامج .

أهمية المتغيرات :

- 1- حجز مكان في الذاكرة للمتغير .
- 2- إعلام المترجم بنوع البيانات التي يمكن أن تخزن في المتغير .

استخدامات المتغيرات :

- 1- تخزين مؤقت للبيانات .
- 2- التعامل مع قيم مخزنة .
- 3- إعادة استخدام البيانات نتيجة تغيرات داخل وأثناء البرنامج .
- 4- الإختصار وسهولة التعديل والصيانة وذلك من استخدام (%type) و (%rowtype) سيأتي شرحها لاحقاً ، و من خلالهما يمكن تعريف متغير حسب نوع عمود أو صف في قاعدة البيانات مما يتيح قدر كبير من المرونة دون التقيد بنوع معين من البيانات .

شروط تسمية المتغيرات :

- 1- اسم المتغير يجب أن يبدأ بحرف .
- 2- يمكن أن يحتوي اسم المتغير على حروف وأعداد أو شرطة سفلية أو الحرف \$.
- 3- لا يحتوي اسم المتغير على رموز خاصة .
- 4- لا يحتوي اسم المتغير على مسافات .
- 5- لا يكون اسم المتغير ضمن الأسماء المحجوزة في اللغة مثل Declare, for, if ... الخ.
- 6- لغة PL/SQL لا تفرق بين الحروف الكبيرة والصغيرة فمثلاً x لا يختلف عن X .
- 7- يفضل أن يكون اسم المتغير ذو معنى .

أنواع المتغيرات في لغة PL/SQL :

تنقسم المتغيرات إلى قسمين أساسيين :

1- متغيرات PL/SQL :

وتحتوي على عدة أنواع منها :

أ- المفردة Scalar.

ب- المركبة (المعقدة) Composite .

ت- المشار بها (عناوين) Reference .

ث- ذات الأحجام الكبيرة LOB(Large Object) .

2- متغيرات ليست PL/SQL :

مثل متغيرات : Bind , host .

فما علينا من هذه الأنواع في هذا المنهج إلا الفقرة الأولى (المفردة) .

كيفية تعريف المتغيرات في لغة PL/SQL :

يتم تعريف المتغيرات في لغة PL/SQL في جزء التصريح والشكل العام للتعريف هو :

Variable_name [CONSTANT] datatype (precision);

اسم المتغير [قيد] نوع البيانات (الدقة أو الطول) ;

ملحوظة مهمة :

1- يجب أن ينتهي التعريف بفاصلة منقوطة .

2- اسم المتغير ونوعه إجباريان عند تعريف أي متغير .

مثل ما قلنا أن من أنواع المتغيرات في هذه اللغة (المفردة) فما معنى المفردة ؟

المفردة المقصود منها هنا : أن لا يحتوي قيمة المتغير سوى قيمة واحدة فقط . مثل :

1- البيانات الرقمية : Number .

2- البيانات النصية : Char , Varchar2 .

3- البيانات البولية : Boolean .

4- البيانات الزمنية (التاريخ) : Date .

5- الاستثناءات : Exception . وغيرها .

سنأخذ كل أنواع البيانات السابقة وكيفية التعريف بالتفصيل .

تعريف المتغيرات الرقمية :

يمكن أن تتضمن البيانات الرقمية وسيطين اثنين : الدقة (الطول الكامل للقيمة) و القيمة العشرية (عدد الأرقام الممكن وضعها إلى يسار أو يمين الفاصلة العشرية .
البيانات الرقمية تنقسم إلى قسمين :

1- البيانات الرقمية الصحيحة ويمكن تعريفه بالشكل التالي :

Salary integer(3);

2- البيانات الرقمية ذات الفاصلة العشرية ويمكن تعريفه بالشكل التالي:

age number(3);

فهذا المتغير قيمته عددا دون تحديد القيمة العشرية ويمكن أن تحتوي هذه القيمة ثلاث أرقام عشرية وثلاث أرقام عشرية إلى اليسار ، بحيث لا يتجاوز الطول الأعظمي أكثر من ثلاثة مواضع .

summary number(3,2);

فهذا المتغير طوله 3 منها عددان إلى اليمين بعد الفاصلة العشرية

ملحوظة :

* يتم إعطاء قيمة افتراضية للدقة مقدارها 38 .

تعريف المتغيرات النصية :

يمكن تعريف البيانات النصية بشكل عادي ، ويتكون البيانات النصية من نمطين :

1- CHAR -2 VARCHAR2 . والشكل العام لتعريف البيانات النصية :

Job char(3);

Name varchar2(30);

X char();

ملحوظة :

- 1- تحتوي التعاريف من النوع char فراغات في المواضع غير المشغولة .
- 2- لا يمكن اعتبار الفراغات والقيم الفارغة نفس الشيء وحتى لا يمكن إجراء أي مقارنة بينهما .
- 3- يعتبر الطول الافتراضي عندما لا يتم تحديد الطول لنمط البيانات char هو 1 ، والطول الأعظمي هو 32767 .
- 4- يجب تحديد الطول عند تعريف متغير لنمط البيانات varchar2 ، والطول الأعظمي هو 32767 .

5- تسمح اللغة بتعريف متغيرات من الأنماط char و varchar2 بأطوال أعظمية أكبر .

تعريف المتغيرات من أنواع أخرى :

1- Boolean : يستخدم لتسجيل حالة ما ، ويمكن أن يأخذ القيمة True أو False أو Null .

مثل: Yes Boolean ;

2- Date : يستخدم لتسجيل قيم التاريخ .

مثل: s_day date ;

3- Exception : يستخدم لتعريف استثناء مخصص أو مقبض للخطأ .

مثل: e_error exception;

تعريف القيود :

يمكن وضع القيود على المتغيرات معرفة في كتلة الترميز ، ويُعرف القيد بأنه شرط يتم وضعه على المتغير. ويوجد نوعان شائعان من القيود وهما :

1- CONSTANT : وهذا القيد يتأكد من أن القيمة لم تتغير بعد نسب قيمة أولية للمتغير . فإذا

حاولت تعليمة ما تغيير القيمة ، سيحدث خطأ .

2- NOT NULL : هذا القيد يتأكد من أن المتغير يحتوي على دائماً على قيمة ، فإذا حاولت

تعليمة ما نسب قيمة فارغة إلى المتغير ، سيحدث خطأ .

نسب قيم للمتغيرات :

توجد طريقتان لنسب القيم للمتغيرات في لغة PL/SQL :

1- معامل النسب (:=) مثل :

Salary number := 15;

2- الكلمة المفتاحية (INTO) وتستخدم في تعليمة (SELECT) و (FETCH) ، وسيأتي شرحه

بالتفصيل لاحقاً .

تعريف متغيرات من نوع تسجيلية PL/SQL ومتغيرات المجموعات :

ميزة أضيفت في لغة PL/SQL وهي تعريف متغيرات من نوع تسجيلية ومتغيرات المجموعات بمعنى : يمكن تعريف متغير على أساس تعريف عمود أو صف بأكمله في جدول بقاعدة البيانات أو متغير سبق تعريفه ، فالمتغير الجديد يأخذ نفس نوع بيانات عمود جدول في قاعدة البيانات ، وبذلك يتيح قدر كبير من المرونة في تعريف المتغيرات ، وأيضاً يوفر الكثير من الوقت للمبرمج ، وأيضاً اختصار البرنامج .

الفوائد الهامة من هذه الميزة :

- 1- يستطيع المطور أن يعرف بشكل آلي متغير بنفس مواصفات بيانات عمود جدول أو متغير من نوع مؤشر ، وبدون معرفة المواصفات لبيانات العمود أو المؤشر .
- 2- يستطيع المطور إعداد المتغيرات لمؤشر ما أو تسجيلية جدول بتعليمة واحدة . وستمتلك المتغيرات نفس المواصفات للجدول أو متغيرات المؤشر .

*** الخاصية أو الأداة (%TYPE) :**

تعتبر الأداة (%TYPE) أول أداة لتعريف متغير مجموعة ، والتي تسمح بتعريف متغير بنفس مواصفات عمود جدول بقاعدة البيانات ، والشكل العام للتعريف بهذه الأداة :

VARIABLE_NAME TABLE_CURSOR_NAME.COLUMN_NAME%TYPE ;

اسم المتغير اسم الجدول أو المؤشر %TYPE اسم العمود

وفي ما يلي نأخذ مثالاً على ذلك :

كتلة برمجية لعرض اسم الموظف وراتبه من جدول الموظفين (EMP) ، بحيث يكون رقم الموظف

يساوي 7782 .

```
SQL> Declare
2  name emp.ename%type;
3  salary emp.sal%type;
4  Begin
5  select ename,sal
6  into name,salary
7  from emp
8  where empno = 7782;
9  DBMS_OUTPUT.PUT_LINE ('Name:' || name || 'Sal Is:' || salary);
10 End;
11 /
Name:CLARKSal Is:2450
PL/SQL تم بنجاح إجراء
SQL> |
```

دعنا الآن نشرح المثال السابق بالتفصيل :

السطر الأول : كلمة مفتاحية (Declare) تحدد جزء التصريح والتعريف .

السطر الثاني : تعريف متغير واسمه (name) ونوع بياناته نفس نوع بيانات عمود (ename) من جدول (emp).

السطر الثالث : تعريف متغير واسمه (salary) ونوع بياناته نفس نوع بيانات عمود (sal) من جدول (emp).

السطر الرابع : كلمة مفتاحية (Begin) تحدد جزء التنفيذ .

السطر الخامس : جملة استعلام (select) يسترجع الاسم والراتب (ename) و (sal) .

السطر السادس : كلمة مفتاحية (into) لإسناد قيم للمتغيرات التي تم تعريفها في جزء التصريح (ename) و

(salary) .

السطر السابع : أشرنا إلى استرجاع السجل يكون من جدول الموظفين (emp) .

السطر الثامن : جملة الشرط لاسترجاع السجل بحيث يكون رقم الموظف (empno) يساوي (7782) .

السطر التاسع : جملة الطباعة بحيث يطبع اسم الموظف وراتبه المسترجع من جملة الإستعلام السابقة .

السطر العاشر : كلمة مفتاحية (End) لإنهاء البرنامج أو كتلة الترميز وتنتهي بفاصلة منقوطة .

السطر الحادي عشر : يدل الرمز (/) على تنفيذ البرنامج أو كتلة الترميز السابقة .

ملحوظة :

1. السطر التاسع : نرى الرمز (||) ، ويستخدم لطباعة قيمة أكثر من متغير أو جملة في نفس جملة الطباعة .
2. لطباعة نص يجب وضعه بين علامتي تنصيص مفردة .

*** الخاصية أو الأداة (%ROWTYPE) :**

الأداة () ، وهي ثاني أداة لتعريف متغير مجموعة ، وتستخدم لتأسيس مصفوفة من المتغيرات مبنية على الأعمدة الموجودة في مؤشر ما أو جدول ما . والشكل العام للتعريف بهذه الأداة :

ARRAY_NAME TABLE/CURSOR_NAME%ROWTYPE ;

اسم المتغير ROWTYPE% اسم الجدول أو المؤشر ;

وفي ما يلي نأخذ مثالاً على ذلك :

كتلة برمجية لعرض اسم الموظف وراتبه ووظيفته من جدول الموظفين (EMP) ، بحيث يكون رقم

الموظف يساوي 7782 . (باستخدام الأداة (%ROWTYPE) :

```
SQL> Declare
  2  x emp%rowtype;
  3  Begin
  4  select *
  5  into x
  6  from emp
  7  where empno = 7782;
  8  DBMS_OUTPUT.PUT_LINE ('Name: ' || x.ename || ' Sal Is:' || x.sal || ' Job Is:' || x.job);
  9  End;
 10 /
Name: CLARK Sal Is:2450 Job Is:MANAGER
تم بنجاح إجراء PL/SQL
SQL>
```

شرح البرنامج السابق :

هو نفس البرنامج الذي كتبنا في الأداة (%TYPE) ولكن يختلف هنا في بعض الفقرات عن البرنامج

السابق من حيث الكتابة :

1- تم تعريف متغير (x) تأخذ نفس نوع أعمدة جدول الموظفين (emp) .

2- جملة الطباعة في السطر الثامن يختلف عن السابق فعند طباعة قيمة متغير يجب وضع اسم المتغير ثم

نقطة ثم اسم العمود في الجدول كالشكل التالي مثلاً :

DBMS_OUTPUT.PUT_LINE ('Name: ' || x.ename);

ملحوظة :

كل ما سبق تسمى المؤشرات الضمنية ، وتسترجع سجلاً واحداً فقط ، وإذا استرجع أكثر من سجل سيحدث خطأ .

أمثلة الفصل الثاني :

1- كتلة برمجية تطبع نص وقيمة المتغير (x) = 1500 :

```
SQL> Declare
 2  x number(4) := 1500;
 3  Begin
 4  DBMS_OUTPUT.PUT_LINE ('x= ' || x);
 5  End;
 6  /
x= 1500
تم بنجاح إجراء
SQL>
```

2- كتلة برمجية تسترجع اسم الموظف وراتبه واسم القسم الذي يشتغل فيه وطباعته ، بحيث يكون رقم

الموظف يساوي (7782) : باستخدام الأداة (%TYPE) :

```
SQL> Declare
 2  name emp.ename%type;
 3  salary emp.sal%type;
 4  de_name dept.dname%type;
 5  Begin
 6  select ename,sal,dname
 7  into name,salary,de_name
 8  from emp,dept
 9  where emp.empno = 7782
10  and emp.deptno = dept.deptno;
11  DBMS_OUTPUT.PUT_LINE ('Name=' || name || ' Sal Is:' || salary || ' dname' || de_name);
12  End;
13  /
Name=CLARK Sal Is:2450 dnameACCOUNTING
تم بنجاح إجراء
SQL> |
```

3- كتلة برمجية يتم فيها طباعة اسم الموظف ووظيفته الذي يحمل الرقم (7782) . ولكن باستخدام الأداة

(%ROWTYPE) :

```
SQL> Declare
 2  x emp%rowtype;
 3  Begin
 4  select *
 5  into x
 6  from emp
 7  where emp.empno = 7782 ;
 8  DBMS_OUTPUT.PUT_LINE ('Name=' || x.ename || ' Job Is:' || x.job);
 9  End;
10  /
Name=CLARK Job Is:MANAGER
تم بنجاح إجراء
SQL> |
```

تمارين الفصل الثاني :

السؤال الأول :

أجب عما يأتي :

1- ماذا تعني كلمة المتغير ؟

2- ما هي أهمية المتغيرات ؟

أ -

ب -

3- أذكر شروط تسمية المتغيرات ؟

أ -

ب -

ج -

د -

هـ -

و -

4- ما هي أنواع المتغيرات في لغة PL/SQL ؟

أ -

ب -

السؤال الثاني :

املاً الفراغات التالية :

1. من أنماط البيانات النصية char و
2. يمكن تعريف المتغيرات في جزء
3. تستخدم الأداة (%ROWTYPE) لـ
4. القيمة الافتراضية من نمط البيانات (CHAR) عندما لا يتم تحديد الطول هو
5. الطول الكامل للقيمة ، و عدد الأرقام الممكن وضعها إلى يمين أو يسار الفاصلة العشرية .
6. يستخدم للبيانات البولية أو المنطقية .
7. DATE يستخدم لـ

السؤال الثالث :

- 1- اكتب كتلة برمجية لاستعراض تاريخ التعيين للموظف الذي اسمه (KING) ؟
- 2- اكتب كتلة برمجية لحساب عدد الموظفين في جدول الموظفين (EMP) ؟
- 3- اكتب كتلة برمجية تقوم فيه تعريف متغيرين من النوع الرقمي وتنسب قيمة للأول (50) والثاني (30) ، ثم تقوم بجمع العددين وطباعة الناتج ؟