

المحاضرة الخامسة

حفظ وتخزين الدوال والإجراءات

Stored Procedures & Functions

فيما سبق كنا نتعامل مع الدوال والإجراءات داخل الوحدات النمطية الغير مسماة **anonymous block** وهي تنتهي وتمحى من الذاكرة بعد تنفيذها وإغلاق الجهاز، فإذا احتجنا إليها مرة أخرى يجب أن نكتبها من جديد. لذلك يوفر لنا الأوراكل إمكانية حفظ وتخزين الدوال والإجراءات داخل قاعدة البيانات حيث يمكننا مثلاً إنشاء وحفظ إجراء يقوم بحذف سجلات من جدول معين بمجرد تمرير رقم العامل مثلاً للإجراء. وتعتبر فكرة حفظ وتخزين الدوال والإجراءات مفيدة جداً في تطبيقات الشبكات **Clint/Server** لأنه يمكننا أن نرسل أمر واحد للـ **Server** لتنفيذ إجراء معين فيقوم الـ **Server** بتنفيذ كمية ضخمة من الأوامر المخزنة داخل الإجراء المعين بدلاً من أن نرسل كل هذه الأوامر على حدة وبذلك نكون قد خففنا الضغط على الشبكة.

إنشاء وحفظ الإجراء :-

Syntax: -

```
CREATE [ OR REPLACE ] PROCEDURE pro_name [ (argument1 [IN |OUT]
arg_datatype [,.....] ) ] IS
Local variables declaration ;
BEGIN
    statements ;
END [ pro_name ] ;
```

عند استخدام المعامل **OR REPLACE** فهذا يعني بأنه إذا كان هناك إجراء سابق مخزن بنفس الاسم فسوف يتم حذفه وحفظ الإجراء الجديد بدلاً منه لذلك يجب أن نكون حذرين عند استخدامه.

مثال: إنشاء إجراء محفوظ لحذف أي سجل من جدول my_emp :-

```
SQL> CREATE OR REPLACE PROCEDURE del_record(del_rec_no number) IS
2 BEGIN
3 delete from my_emp
4 where
5 empno = del_rec_no;
6 END;
7 /
```

Procedure created.

في المثال السابق قمنا بعمل إجراء محفوظ ليقوم بحذف أي سجل من جدول **my_emp** . وعليه يمكننا استخدام هذا الإجراء ومناداته وتمرير رقم العامل المراد حذفه كما في المثال التالي.

مثال: استخدام إجراء محفوظ:-

```
SQL> declare
2 mem_empno number;
3 begin
4 mem_empno := 9999;
5 del_record(mem_empno);
6 end;
7 /
```

PL/SQL procedure successfully completed.

نلاحظ في المثال السابق أننا قمنا باستدعاء الإجراء المحفوظ **Del_record** ومررنا له المتغير **mem_empno** وهو يمثل رقم العامل المراد حذفه من الجدول.

إنشاء وحفظ الدوال :-

Syntax: -

```
CREATE [ OR REPLACE ] FUNCTION fun_name [ (argument-1 [IN |OUT]
arg_datatype [,.....] ) ]
RETURN function_datatype IS
Local variables declaration ;
BEGIN
    statements ;
.
END [ fun_name ] ;
```

مثال: إنشاء دالة تقوم بتحويل الريالات إلى ما يعادلها بالدولار:-

```
SQL> CREATE OR REPLACE FUNCTION to_dolar(reyal number)
2 return number is
3 dolar number;
4 begin
5 dolar := reyal / 3.75;
6 return dolar;
7 end;
8 /
Function created.
```

مثال: استخدام الدوال المحفوظة :-

SQL> select to_dolar(4500) from dual;

TO_DOLAR(4500)

1200

كما لاحظنا فإننا يمكن أن نتعامل مع الدوال المحفوظة مثل تعاملنا مع دوال الأوراكل المبنية داخله.

مثال: تحسين الدالة TO_DOLAR :-

لقد قمنا بإنشاء وحفظ الدالة **To_dollar** وهي تعمل بصورة صحيحة ولكن تكمن المشكلة في أن هذه الدالة تتعامل على أن الدولار يساوي **3.75** ريال. فماذا يحدث إذا تغير سعر صرف الدولار؟. لذلك يجب علينا أن ننشئ الدالة **To_dolar** بحيث تقبل تمرير متغيرين هما المبلغ المراد تحويله وسعر صرف الدولار.

SQL> create or replace function to_dolar(reyal number,d_val number)

```
2 return number is
3 dolar number;
4 begin
5 if d_val != 0 then
6     dolar := reyal / d_val;
7 else
8     dolar := 0;
9 end if;
10 return dolar;
11 end;
12 /
```

نلاحظ في الدالة السابقة استخدمنا دالة الشرط **IF** وذلك حتى لا يعطي البرنامج خطأ إذا أدخل المستخدم سعر صرف الدولار بصفر. كما يتضح بالأمثلة التالية :-

SQL> select to_dolar(4500,3.75) from dual;

TO_DOLAR(4500,3.75)

1200

SQL> select to_dolar(4500,0) from dual;

TO_DOLAR(4500,0)

0

التعامل مع الأخطاء عند إنشاء الدوال والإجراءات :-

سوف نقوم بإنشاء الدالة **To_dolar** مرة أخرى مع وجود خطأ عمداً كالتالي :-

SQL> create or replace function to_dolar(reyal number,d_val number)

```
2 return number is
3 dolar number;
4 begin
5 if d_val != 0 then
6 dolar := reyal / d_val
7 else
8 dolar := 0;
9 end if;
10 return dolar;
11 end;
12 /
```

Warning: Function created with compilation errors.

في السطر رقم 6 نلاحظ عدم وجود الفاصلة المنقوطة ولكن الأوراكل قام بحفظ الدالة مع الاكتفاء فقط بتحذير بأن هناك أخطاء في الدالة ولم يذكر ما هي هذه الأخطاء.

معرفة الأخطاء :-

لمعرفة الأخطاء نستخدم الأمر **SHOW ERRORS** وهو أمر من أوامر ال **SQL*PLUS** ، ويقوم هذا الأمر بسرد الأخطاء الموجودة بالدالة أو الإجراء. كما سيتضح بالتالي :-

SQL> show errors

عند تنفيذ هذا الأمر سيعرض الأوراكل النتيجة التالية:-

Errors for FUNCTION TO_DOLAR:

LINE/COL ERROR

7/1 PLS-00103: Encountered the symbol "ELSE" when expecting one of the following:

. (* @ % & = - + ; < / > in mod not rem an exponent (**)
<> or != or ~= >= <= <> and or like between is null is not ||

بعد إنشاء وحفظ الدالة أو الإجراء قد نحتاج إلى استرجاعها **Source Code** لمعرفة كيف تم كتابتها أو لتعديلها، ويتم ذلك بالاستفسار من ال **Oracle Data Dictionary** وهو عبارة عن مجموعة من الجداول **Tables** يقوم الأوراكل فيها بتخزين كل ما يتعلق بقاعدة البيانات نفسها، ونسبة لأن هذه الجداول معقدة إلى حد ما لذلك يوفر لنا الأوراكل مجموعة من جداول العروض **Views** يمكننا الاستفسار منها.

USER_SOURCE

عبارة عن جدول عرض **View** يمكننا الاستعلام منه وهو يوفر أربعة حقول هي :-

NAME	وهو يحتوي على اسم أي من Procedure أو Function أو Package أو Package Body (سنتطرق لهم لاحقاً).
TYPE	يحدد نوع الاسم أي هل هو Procedure أو Function أو Package أو Package Body .
TEXT	ويحتوي على النص أو الشفرة Source code .
LINE	ويحتوي على أرقام أسطر ال Source code .

مثال: إظهار الإجراء DEL_RECORD :-

```
SQL> select text from user_source
2 where
3 name = 'DEL_RECORD';
```

عند تنفيذ الجملة أعلاه سوف تظهر النتيجة بالشكل التالي :-

TEXT

```
-----
PROCEDURE del_record(del_rec_no number) IS
BEGIN
delete from my_emp
where
empno = del_rec_no;
END;
6 rows selected.
```