◆ dynamic SQL

در ادامه مطالب مربوط به PL/SQL در این بخش ق صد دارم تا به معرفی مجموعه د ستوراتی که با عنوان dynamic SQLشـــناخته میشـــنوند پرداخته و به ارائه توضـــیحاتی پیرامون تعریف ، معرفی انواع گوناگون و کاربرد dynamic SQLها در برنامه های نوشته شده با زبان PL/SQL بپردازم

دستور dynamic SQL چیست ؟

- هرگاه در کد نو شته شده با زبان PL/SQL از عبارت execute immediate همراه با یک د ستور SQL ، DDL، TCLکه در یک رشته قرار داده شده است مواجه شدید از dynamic SQL استفاده شده است
- به این نکته دقت کنید که وقتی از ر شته ا ستفاده میکنیم میتوانیم در زمان اجرا عبارت های مورد نظر خودمان را به آن اضافه کرده و دستور SQL یا ساختار بلاک خود را در زمان اجرا ساخته و آن را اجرا کنیم

در چه مواقعی باید از dynamic SQL استفاده کنیم ؟

- هرگاه همه یا بخ شی از د ستور SQL تا قبل از زمان اجرا (Runtime) م شخص نی ست باید از dynamic SQLاستفاده کنیم
- در مواقعی که میخواهیم یک در ستور DDL را در برنامه ای که با زبان PL/SQL نو شته شده ا ست را اجرا کنیم باید از dynamic SQL استفاده کنیم
- اگر میخواهیم ار دستورات DCL در زبان PL/SQL استفاده کنیم باید از dynamic SQL استفاده کنیم
- اگر میخواهیم برنامه ای بنوی سم که عمومی تر با شد و مانند یک ابزار بتواند حالت های گوناگونی را پ شتیبانی کند میتوانیم از dynamic SQL استفاده کنیم

انواع dynamic SQL کدام هستند ؟

حالت ۱: در مواقعی است که میخواهیم از یک دستور DDL یا DCL در برنامه PL/SQL استفاده کنیم که به شکل ذیل استفاده میگردد 👇

execute immediate 'DDL یا DCL دستور'

به مثال ذیل دقت کنید 👇👇👇

```
begin
execute immediate '
create table EMP_DYN
(
employee_id NUMBER(6),
first_name VARCHAR2(20),
last_name VARCHAR2(25) not null,
email VARCHAR2(25) not null,
phone_number VARCHAR2(20),
hire_date DATE not null,
job_id VARCHAR2(10) not null,
salary NUMBER(8,2),
commission_pct NUMBER(2,2),
manager_id NUMBER(6),
department_id NUMBER(4) )';

end;
```

✳در مثال فوق از یک دستور DDL برای ایجاد یک جدول در یک رشته استفاده شده است✳

✳همانطور که مشاهده میکنید داخل عبارت رشته ای دستور مربوط به dynamic SQL سمی کولون (;) وجود ندارد✳

همانطور که در مثال فوق مشاهده کردید از سایر دستورات DDL یا DCL نیز میتوانیم همانند مثال فوق استفاده کنیم

حالت ۲: در مواقعی است که میخواهیم از یک دستور DML در برنامه PL/SQL استفاده کنیم که تعداد پارامترها یا تعداد ستون های دستور فوق تا زمان اجرا (Runtime) نامشخص است در این حالت باید از نگارش ذیل استفاده گردد 👇👇👇

```
execute immediate 'DML Statement ' using value_of_parameter1 ,
value_of_parameter2 , ....
```

به مثال ذیل دقت کنید 👇👇👇

```
declare
v_sql clob;
v_dep_id_value number := &p_dep_id_value;
v_dep_name_value varchar2(100) := &p_dep_name_value;
v_mananger_id number := &p_mananger_id;
v_location_id number := &p_location_id;

begin

if v_mananger_id is null and v_location_id is null then
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value;

elsif v_mananger_id is not null and v_location_id is null then
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, manager_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :manager_id)';
```

```
execute immediate v_sql
using v_dep_id_value, v_dep_name_value, v_mananger_id;

elsif v_mananger_id is null and v_location_id is not null then
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, location_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :location_id)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value, v_location_id;

else v_mananger_id is not null and v_location_id is not null then
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, manager_id, location_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :manager_id,:location_id)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value, v_mananger_id, v_location_id;
end if;

end;
```

✳در مثال فوق از یک د ستور DML برای درج رکورد جدید در جدول سازمانها ا ستفاده شده ا ست ، برنامه فوق بر اساس مقادیر پارامترهای وارد شده تصمیم گیری میکند که از کدامیک از حالت های فوق باید استفاده شود✳

✳به کاربرد دستور using در مثال فوق فوق دقت کنید که مقادیر پارامترهای وارد شده توسط کاربر را به دستور dynamic SQLفوق تزریق میکند✳

☜اگر مقدار پارامترهای p_mananger_id و p_location_id که تو سط کاربر وارد میشود null با شد از د ستور ذیل استفاده میکند که در نام ستون ها و مقادیر متناظرشان دو ستون mananger_id و location_id وجود ندارد

```
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value;
```

☜اگر مقدار پارامترها p_location_id که توسط کاربر وارد میشود null باشد از دستور ذیل استفاده میکند که در نام ستون و مقادیر متناظر location_id وجود ندارد

```
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, manager_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :manager_id)';

using v_dep_id_value, v_dep_name_value, v_mananger_id;
```

☜اگر مقدار پارامترها p_mananger_id که توسط کاربر وارد میشود null باشد از دستور ذیل استفاده میکند که در نام ستون و مقادیر متناظر p_mananger_id وجود ندارد

```
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, location_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :location_id)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value, v_location_id;
```

☜اگر مقدار پارامترها p_mananger_id و p_location_id که توسط کاربر وارد میشود null نباشد از دستور ذیل استفاده میکند

```
v_sql := 'insert into departments' || chr(10) ||
' (department_id, department_name, manager_id, location_id)' ||
chr(10) || ' values' || chr(10) ||
' (:department_id, :department_name, :manager_id,:location_id)';

execute immediate v_sql
using v_dep_id_value, v_dep_name_value, v_mananger_id, v_location_id;
end if;
```

حالت ۳: در مواقعی ا ستفاده مي شود که میخواهیم از یک د ستور select در برنامه PL/SQL ا ستفاده کنیم که نام جدول و ستون ها یا کل پرس و تا زمان اجرا مشخص نیست در این مواقع میتوان از دو نگارش ذیل استفاده کنیم 👇

```
🅐execute immediate 'Select Statment' into variable1,variable2,.. using
parameter_value1, parameter_value2 , ...

🅑open sys_refcursor for 'Select Statment';
```

به دستورات زیر که مثالی از حالت 🅐 است دقت کنید 👇👇👇

```
declare
v_dep_id number := &p_dep_id;
v_cnt pls_integer;
begin
execute immediate 'select count(*) from employees e where
e.department_id = :p_dep_id '
into v_cnt
using v_dep_id;

dbms_output.put_line(v_cnt);
end;
```

✳در مثال فوق مقدار پارامتر شناسه سازمان (p_dep_id) از کاربر پرسیده شده و تعداد کارمندانی که در سازمان وارد شده مشغول به کار هستند را در خروجی نمایش میدهد✳

✳در مثال فوق به کاربرد عبارت های using و into دقت کنید ، برای مقدار دهی خروجی پرس و جو از into و برای ارسال پارامتر به dynamic SQL از using استفاده شده است✳

به دستورات زیر که مثالی از حالت 🅱 است دقت کنید 👇👇👇

```
declare
v_table_name varchar2(100) := &p_table_name;
v_field_name varchar2(100) := &p_field_name;
v_where_caluse varchar2(100) := &p_where_caluse;
v_field_result varchar2(100);
v_ref sys_refcursor;
begin
open v_ref for 'select ' || v_field_name || ' from ' || v_table_name ||
case when v_where_caluse is not null then ' where ' || v_where_caluse else null
end;
loop
fetch v_ref
into v_field_result;
exit when v_ref%notfound;
dbms_output.put_line(v_field_result);
end loop;
close v_ref;
end;
```

✳در مثال فوق از جدولی که کاربر وارد کرده است رکوردها مورد نظر را بر اساس فیلتر وارد شده توسط کاربر استخراج کرده و ستون مورد نظر کاربر را در خروجی چاپ میکند✳

حالت ۴: در مواقعی استفاده میشود که میخواهیم کل بلاک PL/SQL را بصورت پویا نوشته و اجرا کنیم

به مثال ذیل دقت کنید 👇👇👇

```
declare
v_emp_id employees.employee_id%type := &p_emp_id;
v_block_plsql clob;
v_result number;
begin

v_block_plsql := ' declare' || chr(10) ||
' v_res number; v_emp_row employees%rowtype ;' ||
chr(10) || ' begin' || chr(10) ||
' v_emp_row := get_emp(:p_emp_id);
:v_res := v_emp_row.salary * 12 ;' || chr(10) ||
' end;';
execute immediate v_block_plsql
using v_emp_id, out v_result;
dbms_output.put_line(v_result);

end;
```

✳در مثال فوق کل بلاک در متغیری رشته ای قرار داده شده است✳

✳ شنا سه کارمند از کاربر پر سیده شده و در بلاک به تابع get_emp ار سال میشود ، خروجی تابع فوق یک سطر از اطلاعات کارمند فوق ا ست که مقدار حقوق کارمند بازگردانده شده را در ۱۲ ضرب کرده و حقوق سالیانه را بد ست می آورد ، در نهایت مقدار حقوق سالیانه در متغیر v_result بازگردانده میشود✳

✳به نوع متغیر v_result که از نوع out تعریف شده است دقت کنید✳

✳در مثال فوق از تابعی با نام get_emp استفاده شده که متن تابع فوق به صورت ذیل میباشد✳👇👇👇

```
create or replace function get_emp(p_emp_id number)
return employees%rowtype is
v_res employees%rowtype;
begin

if p_emp_id is null then
raise_application_error(-20900,'پارامتر وردي خالي است');
end if;

execute immediate 'select * from employees e
where e.employee_id = :p_emp_id '
into v_res
using p_emp_id;
return v_res;

end get_emp;
```