

## ◆ records (record & % rowtype) ◆

در ادامه مباحث مطرح شده با موضوع PL/SQL در این بخش به معرفی نوع های داده ای ترکیبی (composit data type)ها میپردازم

و در این بخش به معرفی record و در بخش بعد به معرفی collection خواهیم پرداخت

❖ نوع داده ای ترکیبی چیست و چه کاربردی دارد ؟

❖ میتوان گفت که نوع داده ای ترکیبی از کنار هم قرار دادن یک یا چند نوع داده ای شمارش پذیر (scalar data type) مانند date ، number ، varchar2 ایجاد می شود که به منظور تعریف یک ساختار جهت نگهداری داده های مرتبط با هم تعریف میشود ، در واقع انواع داده ای ترکیبی شبیه به ساختار های داده ای زبان های برنامه نویسی مانند class یا structure میباشد

❖ چند نوع داده ای ترکیبی در PL/SQL وجود دارد ؟

❖ دو نوع داده ای ترکیبی در PL/SQL وجود دارد که شامل

- 1. record**
  - user-defined record
  - % rowtype
- 2. collection**
  - index by table
  - nested table
  - varray

❖ در چه مواقعی باید از record و در چه مواقعی باید از collection استفاده کنیم؟

✚ هرگاه نیاز داشته باشیم تا از نوع داده ای ترکیبی برای نگهداری یک سطر داده استفاده کنیم میتوانیم از record استفاده کنیم

✚ هرگاه نیاز داشته باشیم تا از نوع داده ای ترکیبی برای نگهداری بیش از یک رکورد استفاده کنیم میتوانیم از collection استفاده کنیم

❖ در چه مواقعی باید از user\_defined و چه مواقعی باید از %rowtype استفاده کنیم؟

✚ مواقعی که می خواهیم نوع داده ای ترکیبی داشته باشیم که فیلدهای آن در یک جدول یا view وجود ندارد از user\_defined استفاده میکنیم

✚ مواقعی که می خواهیم نوع داده ای ترکیبی داشته باشیم که فیلدهای آن در یک جدول یا view وجود دارد از %rowtype استفاده میکنیم

❖ فیلد در نوع داده ای ترکیبی چیست؟

✚ به هر کدام از اقلام داده ای (ستون ها) در نوع های داده ای ترکیبی یک فیلد (field) گفته میشود

❖ چگونه و در کدام قسمت میتوانیم یک user-defined رکورد تعریف کنیم؟

✚ برای تعریف user\_defined رکورد باید دو گام ذیل را طی کنیم

**گام A:** تعریف نوع داده ای با استفاده از کلمه type

**گام B:** تعریف متغیری از نوع داده ای تعریف شده گام A

نحوه تعریف user-defined رکورد به شکل زیر میباشد

**declare**

**A type <<type\_name>> is record (field\_name1 type , field\_name2 type , field\_name3 type , .... );**

**B variable\_name <<type\_name>>;**

به مثال ذیل دقت نمایید 📌📌📌📌📌

**declare**

```
A type stu_info is record (ST_NUM number , f1_name employees.first_name %  
type, birth_date date );  
B v_stu_info stu_info;  
begin  
v_stu_info.stu_num := 123;  
end;
```

👉 همانطور که مشاهده میکنید گام های **A** و **B** در بخش **declare** تعریف شده اند  
👉 میتوان با استفاده از نقطه (.) جهت مراجعه به فیلدهای رکورد تعریف شده به منظور خواندن و نوشتن دسترسی پیدا کرد

چگونه و در کدام قسمت میتوانیم یک رکورد از نوع **%rowtype** تعریف کنیم؟

✚ برای تعریف رکورد از نوع **%rowtype** کافی است در قسمت **declare** شناسه (identifier) ی از نوع جدول یا **view** به همراه عبارت **%rowtype** تعریف کنیم

به مثال ذیل دقت نمایید 📌📌📌📌📌

**declare**

```
emp_rec employees % rowtype;  
begin  
emp_rec.salary := 2300;  
emp_rec.first_name := 'Ali';  
.....  
end;
```

👉 همانطور که مشاهده میکنید گام **A** برای **%rowtype** استفاده نمیشود و فقط گام **B** در بخش **declare** تعریف شده است

👉 همچنان میتوان با استفاده از نقطه (.) جهت مراجعه به فیلدهای رکورد تعریف شده به منظور خواندن و نوشتن دسترسی پیدا کرد

👉 کاربرد %rowtype در مواقعی است که میخواهیم تمامی ستون های یک سطر از view یا جدول را در متغیری قرار دهیم ، به مثال ذیل دقت نمایید 📄📄📄

```
declare  
emp_rec employees%rowtype;  
begin  
select * into emp_rec from employees emp where emp.employee_id = 100;  
dbms_output.put_line(emp_rec.first_name);  
end;
```

👉 همچنین میتوان از قابلیت های insert یا update سطر های جدول استفاده کرد ، به مثال ذیل دقت نمایید 📄📄📄📄

```
declare  
emp_rec employees%rowtype;  
begin  
  
select 1500,  
first_name,  
last_name,  
'Ali_ebrahimifard@yahoo.com' email,  
phone_number,  
hire_date,  
job_id,  
salary,  
commission_pct,  
manager_id,  
department_id  
into emp_rec  
from employees emp  
where emp.employee_id = 100;  
  
insert into employees values emp_rec;  
end;
```

در مثال بالا مقادیر ستون های کارمند ۱۰۰ در emp\_rec قرار داده شده و پس از آن در جدول employees درج میشود

```
declare
emp_info_rec emp_info%rowtype;
begin
select * into emp_info_rec from emp_info e where e.employee_id = 100;
emp_info_rec.salary := 2300;
update emp_info e set row = emp_info_rec where e.employee_id = 101;

end;
```

در مثال بالا مقادیر ستون های کارمند ۱۰۰ در emp\_info\_rec قرار داده شده و تمام مقادیر ستون های کارمند ۱۰۱ با مقدار فوق جایگزین میشود

\*\*\* در بحث بعدی راجع به collection و روش تعریف آن توضیحاتی را ارائه خواهم داد \*\*\*