Title

P13. Access record fields in the PL/SQL block.

Description

A record is a collection of fields that might have different data types.

We use the <record_name>.<field_name> syntax to access the fields of a record.

This syntax is valid either in the PL or SQL context.

For example, we can select columns from a table into a record have the same structure as the selected columns. We can assign a value to a record field in a PL statement.

Examples

Example 1: Select columns from a table into a record

Let select the employee_id, first_name, and last_name columns from the employees table into a record and print the three fields.

To complete the task, we need the following steps: 1. Define the record type containing the fields employee_id, first_name, and last_name. 2. Declare a record variable of the record type. 3. Select the columns of a row into the record variable 4. Print the fields of the record variable.

```
set serveroutput on
declare
    -- #1 Define the record type
    type emp_rec is record (
        employee_id employees.employee_id%type,
        first_name employees.first_name%type,
        last_name employees.last_name%type
    );
    -- #2 Declare a record variable
    emp emp_rec;
begin
    -- #3 Select the columns into the record variable
    select employee_id, first_name, last_name
    -- #4 Column values are assigned to the record fields in the
       corresponding positions
    into emp.employee_id, emp.first_name, emp.last_name
    from employees
    where employee_id = 100;

```

```
    -- #5 Print the fields of the record variable
    dbms_output.put_line('Employee ID: ' || emp.employee_id);
    dbms_output.put_line('First Name: ' || emp.first_name);
    dbms_output.put_line('Last Name: ' || emp.last_name);
end;
/
```

When your record type has the same structure as the selected columns, you can assign all the columns to the record variable in one statement.

For example, the select statement in above code can be rewritten more concisely as follows:

```
declare
    ...
begin
    select employee_id, first_name, last_name
    into emp
    from employees
    where employee_id = 100;
    ...
end;
/
```

Example 2: Assign a value to a record field

Let say you want to assign 100, 'John', and 'Doe' to the `employee_id`, `first_name`, and `last_name` fields of a record variable in the PL/SQL block, respectively. This is might be case when you want to initialize a record variable and use it later in the block.

You have two ways to assign values to the record fields: 1. Use the Record Type definition as a constructor to create a record variable. 2. Assign a value to each field of the record variable.

Let demo the first way:

```
set serveroutput on
declare
    -- #1 Define the record type
    type emp_rec is record (
        employee_id employees.employee_id%type,
        first_name employees.first_name%type,
        last_name employees.last_name%type
    );
    -- #2 Declare a record variable and initialize it using the Record
       Type definition as a constructor
    emp emp_rec := emp_rec(100, 'John', 'Doe');
begin
    dbms_output.put_line('Employee ID: ' || emp.employee_id);
    dbms_output.put_line('First Name: ' || emp.first_name);
```

```
    dbms_output.put_line('Last Name: ' || emp.last_name);
end;
/
```

When you want to update a field of the record variable, you can assign a new value to the field as follows:

```
set serveroutput on
declare
    -- #1 Define the record type
    type emp_rec is record (
        employee_id employees.employee_id%type,
        first_name employees.first_name%type,
        last_name employees.last_name%type
    );
    -- #2 Declare a record variable and initialize it using the Record
       Type definition as a constructor
    emp emp_rec := emp_rec(100, 'John', 'Doe');
begin
    -- #3 Assign a new value to the first_name field
    emp.first_name := 'Jane';
    dbms_output.put_line('Employee ID: ' || emp.employee_id);
    dbms_output.put_line('First Name: ' || emp.first_name);
    dbms_output.put_line('Last Name: ' || emp.last_name);
end;
/
```

In the above code, the `first_name` field of the `emp` record variable is updated to 'Jane'.