# TIPOS REGISTROS

* Un tipo registro es un tipo de dato complejo , el cual permite al programador agrupar en un registro distintos atributos .
* Con este tipo de datos, podremos agrupar con un único nombre, varias columnas que queramos utilizar.
* Utilizaremos la palabra Type para que el compilador cree un tipo de datos nuevo
* la creación de este nuevo tipo de datos , lo podemos hacer a nivel de toda la base de datos o a nivel de un subprograma, y solo en este nivel se podría crear.
* Para acceder a cada uno de los campos del registro utilizamos . como notación.

# Sintaxis

CREATE TYPE <type\_name\_db> IS RECORD (

<column 1> <datatype>,

);

Con esta sintaxis creamos un tipo de datos a nivel de subprograma.

Como recordamos todas las sentencias de PL/SQL terminan con ; y el bloque se ejecuta con / al final.

Ejemplo:

Vamos a ver como crear un tipo registro a nivel de subprograma, es decir, solo está present este registro en este programa, no en la base de datos.

Vamos a crear el tipo registro emp-dept y a continuación nos declaramos la variable empleado, que es una variable de este tipo creado.

Este tipo de datos lo utilizamos exactamente igual que si fuese uno básico , varchar, number, etc

Solución

## DECLARE

TYPE emp\_det IS RECORD (

## EMP\_NO NUMBER, EMP\_NAME VARCHAR2(150), MANAGER NUMBER,

SAL NUMBER

);

empleado emp\_det; BEGIN

empleado.emp\_no:= 1001; empleado.emp\_name:='PETER'; empleado.manager:= 1000;

empleado.sal:=10000; dbms\_output.put\_line('Employee Detail');

dbms\_output.put\_line ('Employee Number: '||empleado.emp\_no); dbms\_output.put\_line ('Employee Name: '||empleado.emp\_name); dbms\_output.put\_line ('Employee Salary: ' ||empleado.sal); dbms\_output.put\_line ('Employee Manager Number: '||empleado.manager); END;

/

# COMO CREAR VARIABLES REGISTROS Y CARGARLAS CON DATOS DE TABLAS

## DECLARE

TYPE emp\_det IS RECORD (

## EMP\_NO NUMBER, EMP\_NAME YARCHAR2( 150), MANAGER NUMBER,

SALARY NUMBER

);

empleado emp\_det; BEGIN

INSERT INTO emp (emp\_no, emp\_name, salary, manager) VALUES (1002,'PETER', 15000,1000);

## COMMIT;

SELECT emp\_no, emp\_name, salary, manager INTO empleado FROM emp WHERE emp\_no=1002;

dbms\_output.put\_line (‘Employee Detail’);

dbms\_output.put\_line (‘Employee Number: ‘||empleado. emp\_no); dbms\_output.put\_line (‘Employee Name: ‘||empleado. emp\_name); dbms\_output.put\_line (‘Employee Salary: '||empleado. salary); dbms\_output.put\_line (‘Employee Manager Number: '||empleado.manager); END;

/

Veamos lo que hace

1. como vemos hacemos un insert en emp con los datos de Peter
2. Hacemos una select como veremos más adelante que se hacen en PL/SQL, es decir, recupera un dato que se guarda en la variable empleado
3. A continuación, visualizamos los datos del empleado

**Imagina que el registro es un "bloque" que guarda diferentes valores:**

Por ejemplo, supongamos que queremos almacenar información sobre un empleado, como su número de empleado, su nombre, su salario y el número de su jefe. En lugar de crear varias variables separadas para cada uno de estos datos, podemos usar un registro para almacenarlos juntos.

**1. Definición del tipo de registro:**

TYPE emp\_det IS RECORD (

EMP\_NO NUMBER,

EMP\_NAME VARCHAR2(150),

MANAGER NUMBER,

SALARY NUMBER

);

* **TYPE emp\_det IS RECORD**: Aquí estamos creando un **tipo de registro** llamado emp\_det. Este es como un "molde" o plantilla para crear registros que almacenarán datos de empleados.
* **Dentro del RECORD**, definimos cuatro campos (también llamados **atributos** o **componentes**) que van a formar parte del registro. Cada campo puede tener un tipo diferente:
  + **EMP\_NO NUMBER**: Un campo para almacenar el número de empleado (tipo NUMBER).
  + **EMP\_NAME VARCHAR2(150)**: Un campo para almacenar el nombre del empleado (tipo VARCHAR2, que es una cadena de texto con hasta 150 caracteres).
  + **MANAGER NUMBER**: Un campo para almacenar el número de empleado del jefe (tipo NUMBER).
  + **SALARY NUMBER**: Un campo para almacenar el salario (tipo NUMBER).

**Nota:** Cada uno de estos campos es como una propiedad de un objeto. Por ejemplo, el emp\_no es el número del empleado, el emp\_name es su nombre, y así sucesivamente.

**2. Crear una variable de tipo emp\_det:**

empleado emp\_det;

* Aquí estamos creando una **variable llamada empleado** que será de tipo emp\_det. Esta variable es un "registro" que tendrá todos esos campos que definimos antes: emp\_no, emp\_name, manager, y salary.
* **La variable empleado** actúa como una "caja" que puede contener todos esos valores. Entonces, cuando usamos esta variable, estamos accediendo a cada uno de sus campos (por ejemplo, empleado.emp\_no, empleado.emp\_name).

**3. Insertar un registro en la base de datos:**

INSERT INTO emp (emp\_no, emp\_name, salary, manager)

VALUES (1002, 'PETER', 15000, 1000);

COMMIT;

* Esto inserta un registro en la tabla emp con los valores del número de empleado (1002), el nombre del empleado (PETER), el salario (15000), y el número del jefe (1000).

**4. Recuperar los datos en un registro:**

SELECT emp\_no, emp\_name, salary, manager INTO empleado

FROM emp WHERE emp\_no = 1002;

* Aquí, usamos un **SELECT INTO** para **recuperar los datos** del empleado con emp\_no = 1002 desde la tabla emp y asignarlos a la variable empleado.
* Después de ejecutar este código, el **registro empleado** contiene todos esos datos:
  + empleado.emp\_no tiene el valor 1002.
  + empleado.emp\_name tiene el valor 'PETER'.
  + empleado.salary tiene el valor 15000.
  + empleado.manager tiene el valor 1000.

**5. Mostrar los datos almacenados en el registro:**

dbms\_output.put\_line('Employee Detail');

dbms\_output.put\_line('Employee Number: ' || empleado.emp\_no);

dbms\_output.put\_line('Employee Name: ' || empleado.emp\_name);

dbms\_output.put\_line('Employee Salary: ' || empleado.salary);

dbms\_output.put\_line('Employee Manager Number: ' || empleado.manager);

* Finalmente, con dbms\_output.put\_line, mostramos cada uno de los valores almacenados en el registro empleado. Cada campo dentro del registro (emp\_no, emp\_name, salary, manager) se accede de la siguiente manera: **empleado.emp\_no**, **empleado.emp\_name**, **empleado.salary**, etc.

**Resumen:**

* Un **registro** en PL/SQL es una estructura que agrupa varios datos bajo un mismo nombre.
* Los **campos del registro** pueden ser de diferentes tipos de datos (como NUMBER, VARCHAR2, etc.).
* Un **registro** te permite trabajar con múltiples valores de manera organizada, como si fuera un solo objeto con diferentes propiedades.
* Puedes usar registros para almacenar los resultados de consultas (SELECT INTO), para manipular los datos de manera más sencilla y estructurada.

Espero que ahora esté más claro el concepto de **registro**. ¿Tienes alguna pregunta sobre alguna parte específica del código o el concepto?