

# Gestión de bases de datos

Registros y tablas

# Registros. Sintaxis

Un registro es un grupo de elementos relacionados, almacenados en campos, cada uno de los cuales tiene su propio nombre y se tratan como una sola unidad lógica.

Sintaxis:     TYPE nombre\_tipo IS RECORD(  
                campo1 tipo1 [NOT NULL] [:= expr1],  
                campo2 tipo2 [NOT NULL] [:= expr2],  
                ....  
                campoN tipoN [NOT NULL] [:= exprN]);

# Registros. Sintaxis

Ejemplo. Tipo registro direccion

```
TYPE direccion IS RECORD (  
    calle VARCHAR2(50),  
    numero INTEGER(4),  
    piso INTEGER(4),  
    puerta VARCHAR2(2),  
    provincia VARCHAR2(20),  
    pais VARCHAR2(20) := 'España' );
```

**Los campos de un registro que no sean inicializados, se inicializarán a NULL.**

# Registros. Declaración

Declaración de una variable:

`v_ejemplo t_tipoRegistro;`

*Ej: `mi_direccion direccion;`*

Para acceder a los campos de un registro se utiliza el punto (.)

`v_ejemplo.campo;`

*Ej: `mi_direccion.provincia := 'Sevilla';`*

**Para rellenar los valores de los registros se indica el nombre de la variable de registro seguida de un punto y el nombre del campo a rellenar**

### Ejercicio 1

Declarar un tipo registro Tpersona con los siguientes campos: un código de tipo numérico, un nombre de tipo cadena de 100 caracteres y la edad integer. Asignarle valor a una variable de tipo Tpersona e imprimirlo por pantalla.

# Anidación.

Un registro puede estar anidado. Ejemplo:

```
DECLARE
    TYPE PAIS IS RECORD (
        CO_PAIS NUMBER,
        DESCRIPCION VARCHAR2(50),
        CONTINENTE VARCHAR2(20)
    );
    TYPE MONEDA IS RECORD (
        DESCRIPCION VARCHAR2(50),
        PAIS_MONEDA PAIS /* Un campo de un registro puede ser de un tipo Registro. */
    );

    miPais PAIS;
    miMoneda MONEDA;

BEGIN
END;
/
```

### Anidación.

Para poder asignar un registro a otro, ambos deben ser del **mismo tipo**, no basta que tengan el mismo número de campos. Ejemplo:

```
DECLARE
```

```
    miPAIS PAIS;
```

```
    otroPAIS PAIS;
```

```
BEGIN
```

```
    miPAIS.CO_PAIS := 27;
```

```
    miPAIS.DESCRIPCION := 'ITALIA';
```

```
    miPAIS.CONTINENTE := 'EUROPA';
```

```
    otroPAIS := miPAIS;
```

```
END;
```

```
/
```

# Atributo %TYPE.

Se pueden declarar campos del mismo tipo de un campo de una tabla con %TYPE. Ejemplo:

```
DECLARE
    TYPE TDEPT IS RECORD (
        DEPTNO NUMBER(2) NOT NULL := 99,
        DNAME DEPT.DNAME%TYPE
    );
    dept_rec TDEPT;
BEGIN
    dept_rec.DNAME := 'PURCHASING';
END;
/
```



### Atributo %TYPE.

Ejemplo: recuperar los datos de un empleado con EMPNO igual a 7900 de la tabla EMP.

```
DECLARE
    TYPE TEMPLEADO IS RECORD (
        CODIGO EMP.EMPNO%TYPE,
        NOMBRE EMP.ENAME%TYPE,
        PUESTO EMP.JOB%TYPE
    );
    V_VAR1 TEMPLEADO;
BEGIN
    SELECT EMPNO, ENAME, JOB INTO V_VAR1 FROM EMP
    WHERE EMPNO= 7900;
    DBMS_OUTPUT.PUT_LINE(V_VAR1.CODIGO || '|' ||
    V_VAR1.NOMBRE || '|' || V_VAR1.PUESTO);
END;
/
```

# Atributo %ROWTYPE.

También se pueden crear registros que mantengan los mismos campos y tipos de datos que una tabla: %ROWTYPE. Ejemplo:

```
DECLARE
```

```
    mi_variable NOMBRE_TABLA%ROWTYPE;
```

# Atributo %ROWTYPE.

Ejemplo: recuperar los datos del empleado con EMPNO igual a 7934 de la tabla EMP.

```
DECLARE
    v_emp EMP%ROWTYPE;
BEGIN
    SELECT * into v_emp FROM EMP WHERE EMPNO
= 7934;
END;
/
```

**Se crea el registro de forma dinámica y se asignan los valores a los campos a través de una select de una tabla**

# Ejercicio 2

2.1. Crea un registro `tpersona` igual que el del ejemplo de teoría. Después, crea otro registro alumno que tenga como campos `nombre` `varchar 100`, y `profesor` `tpersona`.

2.2. Crea una variable `alumno1` del tipo `alumno`, dale valores a todos sus campos y muéstralos por la salida.

2.3. Crea una variable `alumno2` del tipo `alumno`, dale valores solo a `nombre` y `código de profesor`, y saca los datos por la salida.

### Ejercicio 3

Crea un registro `tipodept` con la misma estructura que la tabla `dept` (y mismos tipos de datos).

A continuación, crea una variable fila del tipo `tipodept`. Se quiere guardar en fila todos los datos del departamento cuyo `dptno` es 10.

Muestra por la salida el valor de cada campo guardado en fila.

### Ejercicio 4

Empleando el registro `tipdept` anterior, se quieren guardar solo los datos `deptno` y `dname` de la tabla `dept` para el caso del departamento cuyo `id` es 20. Para ello crea una variable `fila2` del tipo `tipodept`, guarda esos datos y muéstralo por la salida.

### Ejercicio 5

Se quieren guardar todos los datos del departamento cuyo deptno es 30 en una variable del tipo %rowtype llamada filacompleta. Muestra por la salida el valor de los distintos campos de filacompleta.

### Ejercicio 6

Se quieren guardar los datos deptno y loc del departamento cuyo deptno es 40 en una variable del tipo %rowtype llamada filacompleta2. Muestra por la salida el valor de los dos campos de filacompleta2.



# Tablas. Sintaxis

Las tablas son tipos de datos que nos permiten almacenar valores del mismo tipo de dato.

## Sintaxis:

```
TYPE nombre_tipo IS TABLE OF <TIPO_DATOS>  
INDEX BY BINARY_INTEGER
```

Ejemplo:

```
TYPE TNUM IS TABLE OF NUMBER  
INDEX BY BINARY_INTEGER;
```

**INDEX BY BINARY\_INTEGER** es el índice que permite acceder a los elementos de la tabla.

# Tablas. Declaración

Declaración de una variable:

```
v_ejemplo t_tipoTabla;
```

Ejemplo:

```
v_lista TNUM;
```

Para acceder a las posiciones de una tabla se utilizan los paréntesis:

```
v_ejemplo(posicion);
```

Ejemplo:

```
v_lista(1):= 2;
```

### Ejercicio 7

Declarar una tabla de números y asignarle con un bucle for los números del 1 al 10. Además de asignar el valor dentro del bucle, mostrar el valor de la tabla por pantalla en cada iteración.

# Tablas de registros

Se pueden declarar elementos de una tabla de tipo registro:

```
DECLARE
    TYPE PAIS IS RECORD (
        CO_PAIS NUMBER,
        DESCRIPCION VARCHAR2(50),
        CONTINENTE VARCHAR2(20));
    TYPE PAISES IS TABLE OF PAIS INDEX BY BINARY_INTEGER;
    tPAISES PAISES;
BEGIN
    tPAISES(1).CO_PAIS := 27;
    tPAISES(1).DESCRIPCION := 'ITALIA';
    tPAISES(1).CONTINENTE := 'EUROPA';
END;
/
```

### Ejercicio 8

Declarar una tabla de 'personas'. Donde 'personas' es un tipo registro que almacena nombre, apellido1 y apellido2.

Asignarle valores para una persona e imprimirlos por pantalla.

# Funciones de tablas

Cuando trabajamos con tablas de PL/SQL podemos utilizar las siguientes funciones:

**nombredetabla.funcion**

- ☐ FIRST. Devuelve el menor índice de la tabla.  
NULL si está vacía.
- ☐ LAST. Devuelve el mayor índice de la tabla.  
NULL si está vacía.
- ☐ EXISTS(i). Utilizada para saber si en un cierto índice hay almacenado un valor. Devolverá TRUE si en el índice i hay un valor. Se emplea en estructuras de control.

# Funciones de tablas

- ☐ COUNT. Devuelve el número de elementos de la tabla PL/SQL.
- ☐ PRIOR (n). Devuelve el número del índice anterior a n en la tabla.
- ☐ NEXT(n). Devuelve el número del índice posterior a n en la tabla.

# Funciones de tablas

- ☐ DELETE. Borra todos los elementos de la tabla PL/SQL.
- ☐ DELETE(n). Borra el correspondiente al índice n.
- ☐ DELETE(m,n). Borra los elementos entre m y n.



# Ejercicio 9

Sobre el ejercicio 7:

1. Utilizar la función COUNT para devolver el número de elementos.
2. Recorrerlo con FIRST Y LAST.
3. Eliminar el último de la tabla y devolver el número total de elementos.
4. Preguntar si existe valor en la posición 10.

### Ejercicio 10

Se quiere mostrar por pantalla los datos de ciertos empleados (tabla emp). Se pide:

- ☐ Se pedirá al usuario por la entrada de plsql que introduzca dos valores integer que coincidan con dos empnos de la tabla emp, y se guardarán en una tabla de integers.
- ☐ Se debe crear una tabla que contendrá como valores todos los campos de las filas de emp cuyo empno coincida con los de la tabla anterior. Puedes usar registros o %rowtype.
- ☐ Recorre la tabla anterior mostrando por la salida los valores de cada columna de los dos registros.

NOTA: debes usar las funciones FIRST, LAST, COUNT, etc. para los índices de tus bucles

Usa por ejemplo empno 7839 y 7698.



# CEU

*Centro de Estudios  
Profesionales*

---

*Fundación San Pablo Andalucía*