

EL LENGUAJE PL/SQL

- Un lenguaje de propósito general orientado a la inclusión de sentencias SQL
- Estructura del código PL/SQL. Bloques:

```
DECLARE
    <variable,tipo>
BEGIN
    sentencias PL/SQL
END;
```

- Los ficheros terminan con el caracter /
- Uso de SHOW ERRORS.

- Un ejemplo trivial:

```
DECLARE
    parametro NUMBER(3):=123;
BEGIN
    IF parametro=123 THEN DBMS_OUTPUT.PUT('El valor es el ');
                        DBMS_OUTPUT.PUT_LINE(parametro);
                        ELSE DBMS_OUTPUT.PUT_LINE('Hay un error.');
```

END IF;

END;

- Uso de SET SERVEROUTPUT ON.

- Sentencia DECLARE:

```
DECLARE
```

```
    numero NUMBER(3);
```

```
    fecha DATE;
```

```
    cadena VARCHAR2(3);
```

```
    columna_codigo EMPLEADOS.CODIGO_EMPLEADO%TYPE;
```

```
    registro_empleados EMPLEADOS%ROWTYPE;
```

```
BEGIN
```

```
    sentencias PL/SQL
```

```
END;
```

- SELECT ...INTO ...FROM ...WHERE ...;

DECLARE

codigo EMPLEADOS.CODIGO_EMPLEADO%TYPE;

nombre EMPLEADOS.NOMBRE%TYPE;

primer_apellido EMPLEADOS.APELLIDO1%TYPE;

segundo_apellido EMPLEADOS.APELLIDO2%TYPE;

BEGIN

codigo:=1;

SELECT nombre, apellido1, apellido2

INTO nombre, primer_apellido, segundo_apellido

FROM EMPLEADOS

WHERE codigo_empleado=codigo;

DBMS_OUTPUT.PUT_LINE('La tupla se ha leído');

INSERT INTO LISTADO

VALUES (codigo,nombre, primer_apellido, segundo_apellido);

DBMS_OUTPUT.PUT_LINE('La tupla se ha insertado');

END;

SENTENCIAS DE CONTROL

- Sentencia de Selección

```
IF condicion1 THEN ...  
    ELSIF condicion 2 THEN ...  
    ELSE condicion 3 ...  
  
END IF;
```

- Sentencia LOOP:

```
LOOP  
    ...  
    EXIT etiqueta WHEN condicion ;  
    ...  
END LOOP;
```

EJEMPLO ITERACIÓN

DECLARE

```
codigo EMPLEADOS.CODIGO_EMPLEADO%TYPE;  
nombre EMPLEADOS.NOMBRE%TYPE;  
primer_apellido EMPLEADOS.APELLIDO1%TYPE;  
segundo_apellido EMPLEADOS.APELLIDO2%TYPE;  
maximo EMPLEADOS.CODIGO_EMPLEADO%TYPE;
```

BEGIN

```
codigo:=1;  
SELECT Max(Codigo_Empleado) INTO maximo FROM EMPLEADOS;  
FOR indice IN 1..maximo LOOP  
    SELECT codigo_empleado, nombre, apellido1, apellido2  
        INTO codigo, nombre, primer_apellido, segundo_apellido  
        FROM EMPLEADOS WHERE codigo_empleado=indice;  
    INSERT INTO LISTADO  
        VALUES (codigo, nombre, primer_apellido, segundo_apellido);  
    DBMS_OUTPUT.PUT_LINE('La tupla se ha insertado');  
END LOOP;
```

END;

- Mecanismo de SQL para el acceso a base de datos. Uso en SELECT.
- Declaración del cursor:

```
DECLARE
    ...
    CURSOR nombre IS sentencia_select;
    ...
BEGIN
    ...
END;
```

- Operaciones:

```
OPEN nombre_cursor
FETCH nombre_cursor INTO lista_variables
CLOSE nombre_cursor
```

ATRIBUTOS DE LOS CURSORES

%NOTFOUND	Devuelve TRUE cuando falla el FETCH
%FOUND	Contrario
%ROWCOUNT	Devuelve el número de filas tratadas ya con el FETCH
%ISOPEN	Devuelve TRUE cuando el cursor está abierto

PROCESAMIENTO DE LOS CURSORES

DECLARE

```
codigo EMPLEADOS.CODIGO_EMPLEADO%TYPE;  
nombre EMPLEADOS.NOMBRE%TYPE;  
primer_apellido EMPLEADOS.APELLIDO1%TYPE;  
segundo_apellido EMPLEADOS.APELLIDO2%TYPE;  
CURSOR busqueda IS  
    SELECT codigo_empleado,nombre, apellido1, apellido2 FROM EMPLEADOS;
```

BEGIN

```
OPEN busqueda;
```

```
LOOP
```

```
    FETCH busqueda INTO codigo, nombre, primer_apellido, segundo_apellido;  
    EXIT WHEN busqueda%NOTFOUND;  
    INSERT INTO LISTADO VALUES (codigo,nombre, primer_apellido, segundo_apellido);  
    DBMS_OUTPUT.PUT_LINE('La tupla se ha insertado');
```

```
END LOOP;
```

```
CLOSE busqueda;
```

```
COMMIT;
```

```
END;
```

CURSORES PARA ACTUALIZACIÓN

- Los cursores que han de actualizar la tabla que recorren han de declararse de otra forma:

```
DECLARE
```

```
    CURSOR nombre_cursor IS expresión_select
```

```
    FOR UPDATE OF COLUMNS (columna_1,...,columna_n);
```

- Actualización de la tupla actual:

```
UPDATE tabla SET columna_1=expresión_1,...,columna_n=expresión_n
```

```
    WHERE CURRENT OF nombre_cursor;
```

DECLARE

```
CURSOR busqueda_salario IS
    SELECT codigo_empleado, salario FROM EMPLEADOS
    FOR UPDATE OF salario;
codigo empleados.codigo_empleado%TYPE;
salario empleados.salario_empleado%TYPE;
```

BEGIN

```
OPEN busqueda_salario;
LOOP
    FETCH busqueda_salario INTO codigo, salario;
    EXIT WHEN busqueda_salario%NOTFOUND;
    IF salario < 100 THEN
        UPDATE empleados SET salario=salario*0,2 ;
                                WHERE CURRENT OF busqueda_salario ;
    END IF;
END LOOP;
CLOSE busqueda_salario ;
COMMIT;
```

END;