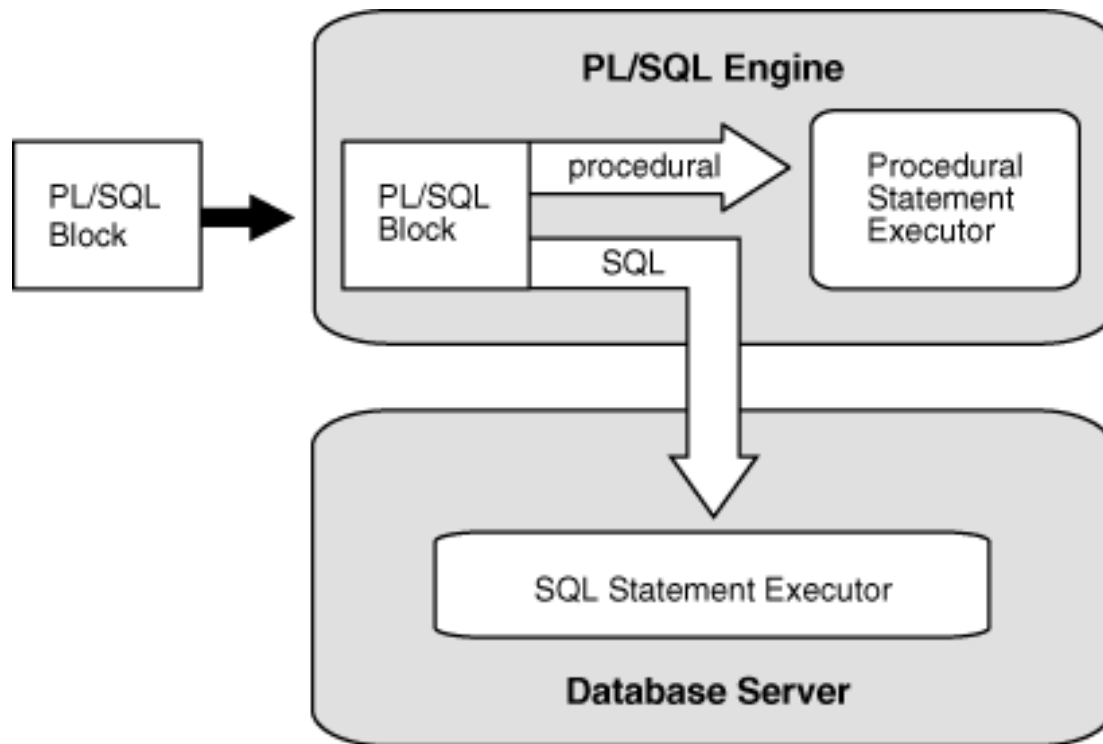




TECNICAS DE ALMACENAMIENTO DE DATOS

Programacion PL/SQL

- SQL (Lenguaje de consulta)
- PL/SQL (Lenguaje de programación)



Objetos de Base de Datos

DIRECTORY

FUNCTION

INDEX

PACKAGE

PACKAGE BODY

PROCEDURE

SEQUENCE

SYNONYM

TABLE

TABLE PARTITION

TRIGGER

TYPE

VIEW

INDEX PARTITION

INDEXTYPE

JOB

JOB CLASS

LIBRARY

LOB

LOB PARTITION

OPERATOR

TABLE SUBPARTITION

TYPE BODY

UNDEFINED

WINDOW

XML SCHEMA

CLUSTER

CONSUMER GROUP

CONTEXT

DESTINATION

EDITION

EVALUATION CONTEXT

PROGRAM

QUEUE

RESOURCE PLAN

RULE

RULE SET

SCHEDULE

SCHEDULER GROUP

```
SELECT DISTINCT OBJECT_TYPE FROM ALL_OBJECTS;
```

Programacion PL/SQL

```
--BLOQUE ANONIMO  
BEGIN
```

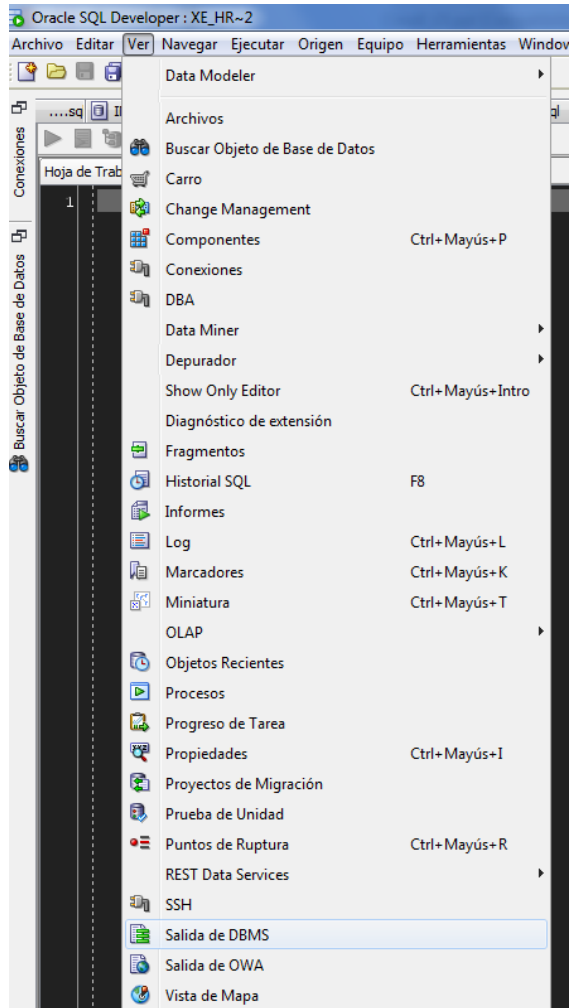
```
    DBMS_OUTPUT.PUT_LINE('¡Hola Mundo!');  
    DBMS_OUTPUT.PUT_LINE('FECHA: ' || SYSDATE);
```

```
END;
```

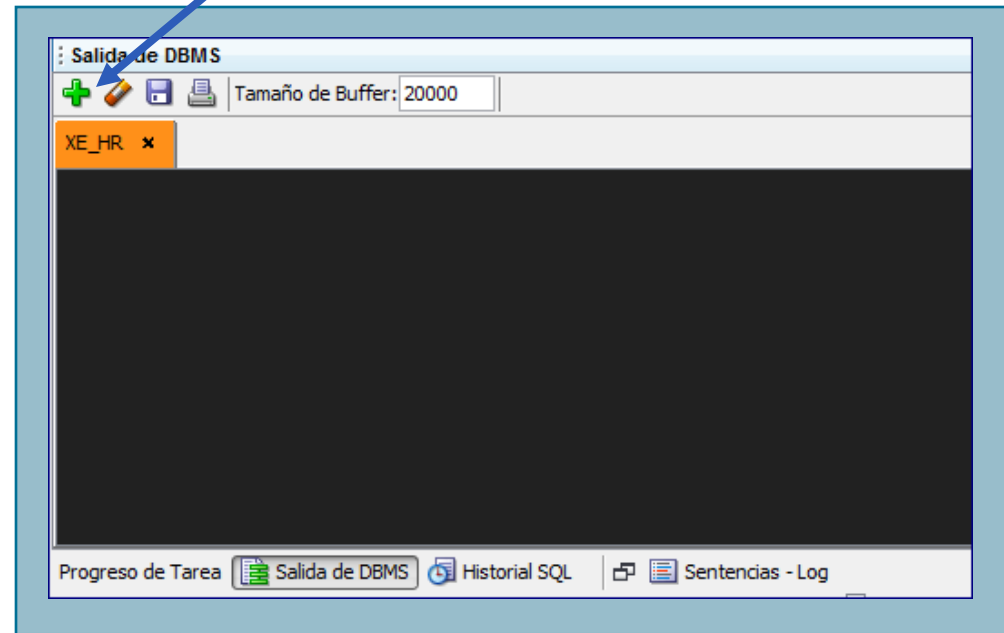
RESULTADO

```
¡Hola Mundo!  
FECHA: 19/05/2016
```

Activar Salida DBMS



Seleccionar la conexión (HR)



Salida de DBMS (DBMS Output)

Operadores en PL/SQL

TIPO DE OPERADOR	OPERADORES
Operador de asignación	<code>:=</code> (dos puntos + igual)
Operadores aritméticos	<code>+</code> (suma) <code>-</code> (resta) <code>*</code> (multiplicación) <code>/</code> (división) <code>**</code> (exponente)
Operadores relacionales (comparación)	<code>=</code> (igual a) <code><></code> (distinto de) <code><</code> (menor que) <code>></code> (mayor que) <code>>=</code> (mayor o igual a) <code><=</code> (menor o igual a)
Operadores lógicos	<code>AND</code> (y lógico) <code>NOT</code> (negación) <code>OR</code> (o lógico)
Operador de concatenación	<code> </code>

Variables

- El nombre de la variable no debería ser el mismo que el de una columna de una tabla utilizada en el bloque
- Por defecto, todas las variables se inicializan a NULL

--SINTAXIS DECLARACION

```
VARIABLE_NAME DATATYPE [ := VALUE ] ;
```

--EJEMPLOS

```
VFECHA DATE ;
```

```
VFECHA DATE := SYSDATE ;
```

DATATYPES

TIPO	EJEMPLO
NUMBER [(PRECISION, ESCALA)]	IMPORTE NUMBER (16, 2)
CHAR [(LONGITUD_MAXIMA)]	NOMBRE CHAR (30)
VARCHAR2 (LONGITUD_MAXIMA)	NOMBRE VARCHAR2 (30)
DATE	FECHA_DE_INGRESO DATE
BOOLEAN	HABILITADO BOOLEAN
BLOB	FOTO_CV BLOB
CLOB	PAPER CLOB

Variables %TYPE

--SINTAXIS

```
VARIABLE_NAME TABLA.CAMPO%TYPE;
```

--EJEMPLOS

```
VFECHA EMPLOYEES.HIRE_DATE%TYPE;  
VEMPLOYEE_ID EMPLOYEES.EMPLOYEE_ID%TYPE;
```

Variables - EJEMPLO

--MANEJO DE VARIABLES / REASIGNACION

DECLARE

V_MENSAJE VARCHAR2(100) := '¡Hola Mundo!';

BEGIN

DBMS_OUTPUT.PUT_LINE(v_mensaje);

V_MENSAJE := '¡Chau!';

DBMS_OUTPUT.PUT_LINE(v_mensaje);

END;

RESULTADO

¡Hola Mundo!
¡Chau!

Estructuras de control

--SINTAXIS

IF (expresion) THEN

-- Instrucciones

ELSIF (expresion) THEN

-- Instrucciones

ELSE

-- Instrucciones

END IF;

Estructuras de control

--EJEMPLO

DECLARE

VID_EMPLEADO EMPLOYEES.EMPLOYEE_ID%TYPE;

BEGIN

IF (VID_EMPLEADO IS NULL) THEN

DBMS_OUTPUT.PUT_LINE('EMPLOYEE_ID NULO');

ELSIF (VID_EMPLEADO = 99) THEN

DBMS_OUTPUT.PUT_LINE('NO NULO - VALOR: ' || VID_EMPLEADO);

ELSE

DBMS_OUTPUT.PUT_LINE('EMPLOYEE_ID: ' || VID_EMPLEADO);

END IF;

END;

Estructuras de control

--EJEMPLO

DECLARE

VID_EMPLEADO EMPLOYEES.EMPLOYEE_ID%TYPE;

BEGIN

IF (VID_EMPLEADO IS NULL) THEN

DBMS_OUTPUT.PUT_LINE('EMPLOYEE_ID NULO');

ELSIF (VID_EMPLEADO = 99) THEN

DBMS_OUTPUT.PUT_LINE('NO NULO - VALOR: ' || VID_EMPLEADO);

ELSE

DBMS_OUTPUT.PUT_LINE('EMPLOYEE_ID: ' || VID_EMPLEADO);

END IF;

END;

RESULTADO

EMPLOYEE_ID NULO

Bucles

En PL/SQL tenemos a nuestra disposición los siguientes iteradores o bucles:

- LOOP
- WHILE
- FOR

--EJEMPLO LOOP

LOOP

-- Instrucciones

IF (expresion) THEN

-- Instrucciones

EXIT;

END IF;

END LOOP;

--EJEMPLO WHILE

WHILE (expresion)

LOOP

-- Instrucciones

END LOOP;

--EJEMPLO FOR

FOR contador IN inicio..final

LOOP

-- Instrucciones

END LOOP;

Bucles - EJEMPLOS

--EJEMPLO LOOP

DECLARE

VCONTADOR NUMBER := 0;

BEGIN

DBMS_OUTPUT.PUT_LINE('-- INICIO LOOP --');

LOOP

DBMS_OUTPUT.PUT_LINE('CONTADOR: ' || VCONTADOR);

IF (VCONTADOR > 5) THEN

DBMS_OUTPUT.PUT_LINE('-- FIN LOOP --');

EXIT;

END IF;

VCONTADOR := VCONTADOR + 1;

END LOOP;

END;

RESULTADO

```
-- INICIO LOOP --  
CONTADOR: 0  
CONTADOR: 1  
CONTADOR: 2  
CONTADOR: 3  
CONTADOR: 4  
CONTADOR: 5  
CONTADOR: 6  
-- FIN LOOP --
```

Bucles - EJEMPLOS

--EJEMPLO WHILE

DECLARE

VCONTADOR NUMBER := 0;

BEGIN

WHILE (VCONTADOR < 10)
LOOP

DBMS_OUTPUT.PUT_LINE('VALOR CONTADOR: ' || VCONTADOR);
VCONTADOR := VCONTADOR + 1;

END LOOP;

END;
/

RESULTADO

VALOR CONTADOR: 0
VALOR CONTADOR: 1
VALOR CONTADOR: 2
VALOR CONTADOR: 3
VALOR CONTADOR: 4
VALOR CONTADOR: 5
VALOR CONTADOR: 6
VALOR CONTADOR: 7
VALOR CONTADOR: 8
VALOR CONTADOR: 9

Bucles - EJEMPLOS

--EJEMPLO FOR
BEGIN

```
FOR i IN 1 .. 10  
LOOP
```

```
    DBMS_OUTPUT.PUT_LINE ('ITERACION: ' || i);
```

```
END LOOP;
```

```
END;  
/
```

RESULTADO

```
ITERACION: 1  
ITERACION: 2  
ITERACION: 3  
ITERACION: 4  
ITERACION: 5  
ITERACION: 6  
ITERACION: 7  
ITERACION: 8  
ITERACION: 9  
ITERACION: 10
```

SELECT INTO

--EJEMPLO SELECT INTO

DECLARE

```
VID_EMPLEADO EMPLOYEES.EMPLOYEE_ID%TYPE := 100;
VFIRST_NAME   EMPLOYEES.FIRST_NAME%TYPE;
VLAST_NAME    EMPLOYEES.LAST_NAME%TYPE;
```

BEGIN

```
SELECT FIRST_NAME, LAST_NAME
      INTO VFIRST_NAME, VLAST_NAME
      FROM EMPLOYEES
      WHERE EMPLOYEE_ID = VID_EMPLEADO;
```

```
DBMS_OUTPUT.PUT_LINE('ID_EMPLEADO: ' || VID_EMPLEADO);
DBMS_OUTPUT.PUT_LINE('FIRST_NAME: ' || VFIRST_NAME);
DBMS_OUTPUT.PUT_LINE('LAST_NAME: ' || VLAST_NAME);
```

```
END;
/
```

RESULTADO

```
ID_EMPLEADO: 100
FIRST_NAME: Steven
LAST_NAME: King
```

EXCEPCIONES

--SINTAXIS PARA CAPTURAR EXCEPCIONES

DECLARE

-- Declaraciones

BEGIN

-- Ejecucion

EXCEPTION

WHEN NO_DATA_FOUND THEN

-- Se ejecuta cuando ocurre excepcion de tipo NO_DATA_FOUND

WHEN ZERO_DIVIDE THEN

-- Se ejecuta cuando ocurre una excepcion de tipo ZERO_DIVIDE

WHEN OTHERS THEN

-- Se ejecuta cuando ocurre una excepcion de un tipo

-- no tratado en los bloques anteriores

END;

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VFIRST_NAME EMPLOYEES.FIRST_NAME%TYPE;

VLAST_NAME EMPLOYEES.LAST_NAME%TYPE;

BEGIN

SELECT FIRST_NAME, LAST_NAME
INTO VFIRST_NAME, VLAST_NAME
FROM EMPLOYEES
WHERE EMPLOYEE_ID IN (100,101);

EXCEPTION WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VFIRST_NAME EMPLOYEES.FIRST_NAME%TYPE;

VLAST_NAME EMPLOYEES.LAST_NAME%TYPE;

BEGIN

```
SELECT FIRST_NAME, LAST_NAME
      INTO VFIRST_NAME, VLAST_NAME
      FROM EMPLOYEES
      WHERE EMPLOYEE_ID IN (100,101);
```

EXCEPTION WHEN **OTHERS** THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

RESULTADO

ERROR: ORA-01422: exact fetch returns more than requested number of rows

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VFIRST_NAME EMPLOYEES.FIRST_NAME%TYPE;

VLAST_NAME EMPLOYEES.LAST_NAME%TYPE;

BEGIN

SELECT FIRST_NAME, LAST_NAME
INTO VFIRST_NAME, VLAST_NAME
FROM EMPLOYEES
WHERE EMPLOYEE_ID = 1;

EXCEPTION WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VFIRST_NAME EMPLOYEES.FIRST_NAME%TYPE;

VLAST_NAME EMPLOYEES.LAST_NAME%TYPE;

BEGIN

SELECT FIRST_NAME, LAST_NAME
INTO VFIRST_NAME, VLAST_NAME
FROM EMPLOYEES
WHERE EMPLOYEE_ID = 1;

EXCEPTION WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

RESULTADO

ERROR: ORA-01403: no data found

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VNUM NUMBER;

BEGIN

SELECT 1/0
INTO VNUM
FROM DUAL;

EXCEPTION WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

EXCEPCIONES

--EJEMPLO EXCEPCIONES

DECLARE

VNUM NUMBER;

BEGIN

SELECT 1/0
INTO VNUM
FROM DUAL;

EXCEPTION WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('ERROR: ' || SQLERRM);

END;

RESULTADO

ERROR: ORA-01476: divisor is equal to zero

EXCEPCIONES

Tipos de excepciones (mas frecuentes)

- NO_DATA_FOUND
- TOO_MANY_ROWS
- ZERO_DIVIDE
- DUP_VAL_ON_INDEX
- CURSOR_ALREADY_OPEN
- INVALID_CURSOR
- INVALID_NUMBER

RAISE_APPLICATION_ERROR

- Permite enviar un mensaje de error personalizado al producirse una excepción PL/SQL

--SINTAXIS

RAISE_APPLICATION_ERROR(<error_num>,<mensaje>);

<error_num> es un entero negativo comprendido entre -20001 y -20999
<mensaje> es la descripción del error

RAISE_APPLICATION_ERROR

--EJEMPLO EXCEPCIONES - RAISE

DECLARE

VNUM NUMBER;

BEGIN

SELECT 1/0
INTO VNUM
FROM DUAL;

EXCEPTION WHEN OTHERS THEN

/* LOG ERROR */

RAISE_APPLICATION_ERROR(-20001, 'No se puede dividir por cero');

END;

Bloques no anónimos

