
2.- INTERACCIÓN CON EL SERVIDOR ORACLE

- 2.1 – Sentencias SQL*PLUS en PL/SQL.
- 2.2 – La sentencia SELECT en PL/SQL.
- 2.3 – Las sentencias DML en PL/SQL.
- 2.4 – Conversiones de nomenclatura.
- 2.5 – COMMIT y ROLLBACK.
- 2.6 – Cursores SQL.

2.1 – SENTENCIAS SQL*PLUS EN PL/SQL

Cualquier sentencia SQL, salvo excepciones, es aplicable a sentencias PL/SQL. La sentencia SELECT en PL/SQL cambia su formato incluyendo la palabra reservada INTO.

Las sentencias DML incluidas COMMIT, ROLLBACK y SAVEPOINT tienen idéntico formato que en SQL*PLUS.

No soporta sentencias DDL o lenguaje de definición de datos (CREATE, DROP etc.), ni sentencias DCL o lenguaje de control de datos (GRANT, REVOKE etc.).

Cuando iniciemos una nueva sesión conviene escribir lo siguiente:

SET SERVEROUTPUT ON: Muestra los errores de un bloque PL/SQL.

SET VERIFY OFF: Elimina las variables de sustitución.

Si no se quiere escribirlas cada vez que iniciemos una nueva sesión se pueden escribir en el fichero LOGIN.SQL.

2.2 – LA SENTENCIA SELECT EN PL/SQL

Esta sentencia sirve para recuperar datos de tablas, tal y como hacía en SQL*PLUS.

Sintaxis:

```
SELECT lista_seleccionada  
INTO {nb_variable1 [ , nb_variable2 .....] | nb_registro}  
FROM nb_tabla  
WHERE condición;
```

Ejemplo 1:

Obtener un bloque PL que calcule la suma de los salarios de la tabla EMP visualizando su resultado.

```
DECLARE
    V_TOTAL NUMBER;
BEGIN
    SELECT SUM(SAL) INTO V_TOTAL FROM EMP;
    DBMS_OUTPUT.PUT_LINE(V_TOTAL);
END;
/
```

Ejemplo 2:

Obtener un bloque PL que visualice el código de departamento y la localidad de la tabla DEPT del departamento SALES.

```
DECLARE
    V_DEPTNO NUMBER(2);
    V_LOC VARCHAR2(15);
BEGIN
    SELECT DEPTNO, LOC INTO V_DEPTNO, V_LOC
    FROM DEPT
    WHERE DNAME='SALES';
    DBMS_OUTPUT.PUT_LINE(V_DEPTNO);
    DBMS_OUTPUT.PUT_LINE(V_LOC);
END;
/
```

2.3 – LAS SENTENCIAS DML EN PL/SQL

Tienen el mismo formato que las sentencias DML de SQL*PLUS

Ejemplo 1:

Insertar en la tabla DEPT un departamento de nombre PUBLICIDAD que está en MADRID asignado directamente el código en una variable a través de la pseudocolumna NEXVAL.

```
DECLARE
    V_DEPTNO DEPT.DEPTNO%TYPE;
BEGIN
    SELECT      DEPTNO_SEQUENCE.NEXTVAL INTO V_DEPTNO
    FROM DUAL;
    INSERT INTO DEPT(DEPTNO, DNAME, LOC)
    VALUES (V_DEPTNO,'PUBLICIDAD', 'MADRID');
END;
/
```

Ejemplo 2:

Suprimir el registro insertado anteriormente:

```
DECLARE
  V_DEPTNO DEPT.DEPTNO%TYPE:= 50;
BEGIN
  DELETE FROM DEPT WHERE DEPTNO= V_DEPTNO;
END;
/
```

2.4 – CONVENCIONES DE NOMENCLATURA

Los identificadores y las columnas de la BD deben tener nombres distintos para evitar ambigüedades en la cláusula WHERE.

Los errores de sintaxis pueden surgir porque PL/SQL comprueba primero el nombre de la variable PL/SQL y luego el de la tabla.

2.5 – COMMIT Y ROLLBACK

Los comandos COMMIT y ROLLBACK se pueden introducir dentro de los bloques PL/SQL siempre y cuando se utilicen sentencias DML. Es conveniente utilizar estos comandos o bien al principio del bloque, lo cual permitirá validar los bloques ejecutados hasta ese instante o ponerlos al final para validar el bloque que se está ejecutando.

Estos comandos funcionan de forma idéntica a SQL*PLUS. Muchos programadores de PL/SQL no suelen incluirlos en los bloques, y lo ejecutan fuera del bloque, una vez terminado el trabajo.

2.6 – CURSORES SQL

Cada vez que se crea una sentencia SQL el servidor Oracle abre un área de memoria en la que se analiza y ejecuta el comando. Esta área de memoria es lo que se conoce con el nombre de cursor. Un cursor es un área de trabajo privada de SQL. Pueden ser de los siguientes tipos:

- **IMPLÍCITOS**: Son los que ya tiene el servidor Oracle. Los utiliza el servidor para analizar y ejecutar las sentencias SQL son:

Tipos	Descripción
SQL%ROWCOUNT	Cuenta el número de filas afectadas por la sentencia SQL más reciente. Sirve para que nos cuente el número de filas que hemos leído del cursor.
SQL%FOUND	Atributo booleano que da como resultado TRUE si la sentencia SQL afecta a alguna fila. Sirve para saber si el cursor tiene más datos o por el contrario ya los ha leído todos.
SQL%NOTFOUND	Igual que el anterior pero da TRUE si la sentencia SQL no afecta a ninguna fila.
SQL%ISOPEN	Siempre es FALSE porque SQL cierra los cursores implícitos inmediatamente después de ejecutarlos.

-
- **EXPLÍCITOS**: Son declarados por el programador.

Ejemplo:

Suprimir de la tabla DEPT aquellos departamentos que tengan un código mayor de 30.
Se utilizará el cursor ROWCOUNT.

```
DECLARE
  V_ELIMI NUMBER:=30;
BEGIN
  DELETE DEPT WHERE DEPTNO>V_ELIMI;
  DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT||' FILAS ELIMINADAS');
END;
/
```