

SQL EN PL/SQL

- PL/SQL soporta el lenguaje de consulta SQL
 - (DQL: sentencia select)
 - DML: Insert, Update y Delete
 - Sentencias para el control de transacciones: Commit, Rollback y Savepoint
- Hay que tener en cuenta que un bloque PL/SQL no conforma una transacción por sí solo, tendremos que controlar nosotros este aspecto con las sentencias SQL correspondientes
- PL/SQL no soporta el lenguaje de definición de datos (DDI) Create table, Alter table o Drop table
- Tampoco el lenguaje de control de datos DCL . Grant y Revoke

SENTENCIA SELECT

- La sentencia Select en un bloque PL/SQL difiere levemente de una sentencia select en SQL
- Debe incluir una cláusula nueva que es la cláusula INTO para recoger los valores devueltos por la consulta
- Estos valores deben almacenarse en variables declaradas en el bloque PL/SQL

```
SELECT lista_select INTO
{ VARIABLE[,VARIABLE...] | registro}
FROM tabla
[WHERE condiciones];
```

- Es muy importante saber que una sentencia select en un bloque PL/sql debe devolver como resultado una única fila de lo contrario generaría un error.

Ejemplo: Guardar en las variables v_id y v_nombre el nombre y el código de departamento del departamento 10

```
Declare
V_id dept.deptno%TYPE;
V_nombre dept.dname%TYPE;
Begin
Select deptno,dname into v_id, v_nombre from dept
Where deptno =10;
End;
/
```

- La cláusula INTO debe contener tantas variables como columnas seleccionemos en la Select

Ejemplo2:

```
,  
variable gmax number;  
begin  
select max(sal) into :gmax from scott.emp where deptno= & g_dept;  
end;  
/  
print gmax
```

- Como vemos hemos utilizado variables de intercambio (g_dept). Estas variables son propias de SQL*PLUS y no hace falta declararlas.
- Para utilizarlas en un código PL/SQL pondremos &g_dept
- Estas variables se pierden en caso de producirse un error, por tanto conviene salvarlas

declare

var1_num1:= & g_dept;

Si fuese alfanumerica sería:

var1_car1:= '& g_dept';

- Las variables de SQL de enlace (gmax) se deben utilizar dentro del código con :gmax

SENTENCIA INSERT

- Vamos a buscar el máximo código de departamento. Lo incrementamos en 1 e insertamos el departamento EDUCACIÓN

```
declare  
v_max SCOTT.dept.deptno%TYPE;  
begin  
select max(SCOTT.DEPT.DEPTNO) into V_MAX from scott.DEPT;  
V_MAX:=V_MAX +1;  
INSERT INTO SCOTT.DEPT(DEPTNO,DNAME) VALUES (V_MAX,'EDUCACION');  
end;  
/
```

- A diferencia de la sentencia select, podemos insertar más de una fila a la vez sin que se produzca ningún error.

SENTENCIA UPDATE

- Para modificar datos en la base de datos

Ejemplo: Introducir un código de un empleado y a ese empleado incrementar su salario en un 10%

Declare

V_incremento number(2) := 10;

Begin

Update scott.emp set scott.emp.sal= scott.emp.sal + (scott.emp.sal*v_incremento/100) where scott.emp.empno= &v_emple;

End;

/

- Al igual que Insert podemos actualizar más de una fila cada vez

SENTENCIA DELETE

- La incorpora PL/SQL para sus bloques

Ejemplo: eliminar el departamento que deseemos

Begin

Delete from emp

Where deptno=& v_deptno;

End;

/

- También se puede eliminar más de una fila cada vez

sql en pl/sql