Interacción con ORACLE

Comparación entre SQL y PL/SQL

- Un bloque PL/SQL no es una unidad transaccional. Los COMMIT, SAVEPOINT y ROLLBACK son independientes de los bloques.
- PL/SQL no soporta sentencias DDL, tal como CREATE TABLE, ALTER TABLE o DROP TABLE.
- PL/SQL tampoco soporta sentencias DCL,tal como GRANT o REVOKE.
- Utlizando el paquete DBMS_SQL o la sentencia EXECUTE IMMEDIATE, PL/SQL puede ejecutar sentencias DDL y DCL dinámicamente.

Recuperación de Datos con PL/SQL

```
SELECT  lista_select
INTO nombre_variable[, nombre_variable, ...] |
nombre_registro
FROM  tabla
WHERE condición;
```

Ejemplo:

```
SELECT apellido, nombre
INTO v_apellido, v_nombre
FROM clientes
WHERE id_cliente = 25;
```

Recomendaciones

- Se debe especificar la misma cantidad de variables de salida en la cláusula INTO que la cantidad de columnas de la cláusula SELECT.
- Para asegurar que los tipos de datos de la cláusula INTO coinciden con los tipos de datos de las columnas, se puede usar el atributo %TYPE.
- Se debe evitar la ambigüedad en la cláusula WHERE, manteniendo una convención de nombres que distinga los nombres de columnas de la base de datos de los nombres de variables PL/SQL.

Recomendaciones (cont.)

- Se puede usar un registro PL/SQL para crear campos que se ajusten a las columnas de una tabla de la base de datos. Por ejemplo:

```
PROCEDURE dept_completo(p_id_dept IN NUMBER,
p_nombre out varchar2) IS
...
dept_reg    dept%ROWTYPE;
BEGIN
SELECT *
INTO   dept_reg  -- Registro PL/SQL
FROM dept
WHERE    id = p_id_dept;
...
p_nombre:=dept_reg.nombre;
...
END dept_completo;
```

Excepciones de SELECT

- TOO_MANY_ROWS: se produce cuando la sentencia SELECT encuentra más de una fila que satisface el criterio de la consulta. El servidor ORACLE produce un error cuyo número es –1422.
- NO_DATA_FOUND: se produce cuando la sentencia SELECT no encuentra filas que satisfagan el criterio de la consulta. El servidor ORACLE produce un error cuyo número es –1403. Si la sentencia SELECT INTO llama funciones de grupo, nunca se generará la excepción porque las funciones de grupo siempre retornan un valor o nulo.

Manipulación de Datos Utilizando PL/SQL

```
PROCEDURE orden_cliente(p_cliente_id  ordenes.cliente_id%TYPE) IS
v_fecha_orden      ordenes.fecha_orden%TYPE  := SYSDATE;
v_vendedor_id      ordenes.vendedor_id%TYPE := 11;
v_tipo_pago    ordenes.tipo_pago%TYPE     := 'EFECTIVO';
v_orden_cumplida  ordenes.orden_cumplida%TYPE:= 'N';
BEGIN
INSERT INTO ordenes (id, cliente_id, fecha_orden, fecha_envio,
            vendedor_id, total, tipo_pago, orden_cumplida)
        VALUES ( orden_id_seq.NEXTVAL, p_cliente_id,
v_fecha_orden, NULL, v_vendedor_id, 0,                    v_tipo_pago,
v_orden_cumplida);
END orden_cliente;
```

Manipulación de Datos Utilizando PL/SQL

```
PROCEDURE nva_fecha_envio(
p_orden_id   ordenes.id%TYPE,
p_fecha_envio     ordenes.fecha_envio%TYPE) IS
BEGIN
UPDATE ordenes
SET fecha_envio = p_fecha_envio
WHERE id = p_orden_id;
END nva_fecha_envio;
```

```
PROCEDURE borrar_orden (p_orden_id ordenes.id%TYPE) IS
BEGIN
DELETE FROM ordenes
WHERE id = p_orden_id;
END borrar_orden;
```