

Al hablar sobre SQL en PL/SQL debemos tener en cuenta que sólo se permiten instrucciones DML (Lenguaje de Manipulación de Datos) como las que ya conocemos (Insert, Update, Delete...) y de control de transacciones como Commit o Rollback

No podremos utilizar instrucciones de definición de datos como crear tablas (Es mentira, con sql dinámico sí se puede)

Otro dato importante es que utiliza un acoplamiento temprano, es decir, comprueba los permisos y la existencia de los objetos cuando el programa está compilando

Ejemplo 1:

```
ejemplo1.sql
1  BEGIN
2  create table zapatilla (
3      nombre char(10),
4      numero number
5  );
6  insert into zapatilla values('Hola',1);
7  END
```

Da error porque a la hora de compilar la tabla zapatilla no existe

DML → En el caso de los selects debemos tener en cuenta que no puede devolver más de una fila y que selecciona datos y los guarda en variables

Ejemplo del libro de un select into(pasarlo para que funcione al tema 3) •

```
ejemplo2.sql
1  DECLARE
2      v_StudentRecord students%ROWTYPE;
3      v_Department classes.department%TYPE;
4      v_Course classes.course%TYPE;
5  BEGIN
6      SELECT * INTO v_StudentRecord
7      FROM students
8      WHERE id = 10000;
9
10     SELECT department, course INTO v_Department, v_Course
11     FROM classes
12     WHERE room_id = 20003;
13 END;
```

Ejemplo del libro de un insert (hacer que funcione en tema 3) •

```

ejemplo2_2.sql
1 DECLARE
2   v_StudentID students.id%TYPE;
3 BEGIN
4   SELECT student_sequence.NEXTVAL INTO v_StudentID
5   FROM dual;
6
7   INSERT INTO students (id, first_name, last_name)
8   VALUES (v_StudentID, 'Timothy', 'Taller');
9
10  INSERT INTO students (id, first_name, last_name)
11  VALUES (student_sequence.NEXTVAL, 'Patrick', 'Poll');
12 END;

```

(opcional poner el delete y el update pero es lo mismo) •

Posible error común es el uso del where, este, primero busca la columna y luego la variable, si estas se llaman igual por ejemplo al hacer un delete, borraría todo.

CAMBIARLO TAMBIÉN •

```

ejemplo3(errcomun).sql
1 DECLARE
2   Department CHAR(3);
3 BEGIN
4   Department := 'CS';
5
6   DELETE FROM classes
7   WHERE department = Department;
8 END;
9

```

Si uno fuera tipo char y el otro tipo varchar2 no borraría nada porque son diferentes tipos, la columna y la variable. Lo suyo sería que en la variable se utilizase %type para que sea del mismo tipo que la columna

Referente al control de transacciones

EJEMPLO DEL LIBRO (CAMBIAR) •

```

ejemplo4CT.sql
1 BEGIN
2   INSERT INTO temp_table(char_col) VALUES('UNO');
3   SAVEPOINT A;
4
5   INSERT INTO temp_table(char_col) VALUES('DOS');
6   SAVEPOINT B;
7
8   INSERT INTO temp_table(char_col) VALUES('TRES');
9   SAVEPOINT C;
10
11  -- OTRAS INSTRUCCIONES
12
13  COMMIT;
14 END;
15

```

Savepoint como su nombre indica es como un punto de guardado, te permite hacer rollback a ese punto, sin tener que hacer rollback de toda la transacción

En este caso si utilizásemos rollback to B; Se desharía el tercer insert into pero se mantendrían los dos primeros insert, es decir el 1 y el 2

- **CAMBIAR TODOS LOS EJEMPLOS Y MIRAR SI HAY ALGUNA COSA MÁS IMPORTANTE A DESTACAR**