
Title

P10. Writing DML statements in PL/SQL blocks.

Description

To manipulate the data in the database tables, you can directly write the SELECT, INSERT, UPDATE, DELETE, and MERGE statements in PL/SQL blocks.

When writing DML statements, you can directly use the PL local variables in them, like the SELECT INTO statement. - The DML statements will not raise exceptions if no rows are affected, in contrast to the SELECT INTO statement with the exact fetch requirement

Note that DML statements are **massive operations**. They iterate over all rows that meet the conditions specified in the WHERE clause.

Refer to the following programming patterns for writing SELECT statements in PL/SQL blocks: - 01. Select a scalar value from a table into a pl local variable - 02. Select multiple column values from a table and put them into PL/SQL local variables

You can also directly write transaction commands in the PL/SQL blocks to control the transactions - Such as COMMIT, ROLLBACK, or SAVEPOINT commands

However, you can not write the DDL statements directly in PL/SQL blocks. - To execute the DDL statements in PL/SQL blocks, you must use the `EXECUTE IMMEDIATE` statement or the `DBMS_SQL` package.

Examples

Example 1: Use the PL local variables in the INSERT statement

Assume the table t1 has two columns: id(PK, NUMBER) and val(NUMBER)

```
1 create table t1 (id number primary key, val number);
```

Write a FOR-LOOP to insert 5 rows into the table t1. The `val` column value is a computation result based on the loop index.

```
1 declare
2     v_val number := 100;
3 begin
4     for i in 1..5 loop
5         insert into t1 (id, val) values (i, v_val * i);
```

```
6     end loop;
7 end;
8 /
```

Example 2: Use the PL local variables in the UPDATE statement

Update the `val` column value with a random number.

```
1 declare
2     v_random number;
3 begin
4     v_random := dbms_random.value(1, 100);
5     update t1 set val = v_random;
6 end;
7 /
```

where: - `dbms_random.value(1, 100)` returns a random number between 1 and 100.

Example 3: Use the PL local variables in the DELETE statement

Delete the rows from the table `t1` where the `val` column value is less than the value stored in the local variable.

```
1 declare
2     v_threshold number := 20;
3 begin
4     delete from t1 where val < v_threshold;
5 end;
6 /
```