

BAB 4 : Interaksi PL/SQL dengan Oracle Server

Tujuan Pembelajaran :

- Membuat SELECT Statement dalam PL/SQL
- Membuat DML (Data Manipulation Language) Statement dalam PL/SQL

4.1. SQL Statement dalam PL/SQL

Pada saat suatu informasi perlu diekstrak dari suatu database, maka digunakan SQL Statement. SELECT Statement dapat digunakan untuk mendapatkan baris-baris pada table. Sedangkan perintah DML (Data Manipulation Language) dapat digunakan untuk memodifikasi data yang ada pada database.

Prinsip yang berlaku pada PL/SQL dalam hubungannya dengan kontrol transaksi dan DML Statement adalah :

- Keyword END menyatakan akhir dari PL/SQL blok, bukan akhir dari transaksi. Jadi satu blok bisa terdiri dari beberapa transaksi
- PL/SQL tidak secara langsung berhubungan dengan DDL (Data Definition Language), jadi dalam PL/SQL tidak ada perintah seperti CREATE TABLE, ALTER TABLE atau DROP TABLE
- PL/SQL tidak mendukung DCL (Data Control Language), jadi dalam PL/SQL tidak ada perintah seperti GRANT atau REVOKE.

4.2. SELECT Statement dalam PL/SQL

Untuk mendapatkan atau *me-retrieve* data dari database, digunakan SELECT Statement.

Bentuk umum (sintak) dari SELECT Statement :

SELECT	<i>select_list</i>
INTO	<i>{variable_name[, variable_name] ...</i> <i>/ record_name}</i>
FROM	<i>table</i>
WHERE	<i>condition</i>];

select_list : daftar kolom (sedikitnya satu) termasuk di dalamnya ekspresi SQL, fungsi baris, fungsi group.

variable_name: scalar variable yang menangani nilai yang dipanggil (*retrieved*)

record_name : PL/SQL record yang menangani nilai yang dipanggil (*retrieved*)

table : nama table database

condition : terdiri dari nama kolom, ekspresi, konstanta dan operator perbandingan termasuk variable dan konstanta PL/SQL.

Klausa INTO dalam SELECT Statement harus ada, dan digunakan untuk menentukan nama variable yang menangani nilai yang didapat dari klausa SELECT.

4.3. Memanggil Data pada PL/SQL

Berikut ini akan dicontohkan SELECT Statement untuk proses *retrieving data* dalam PL/SQL :

DECLARE	
	v_deptno NUMBER(4);
	v_location_id NUMBER(4);
BEGIN	
	SELECT department_id, location_id
	INTO v_deptno, v_location_id
	FROM departments
	WHERE department_name='Sales';

END;	

Contoh kedua berikut menggunakan deklarasi dengan atribut %TYPE :

```
DECLARE
    v_hire_date    employee.hire_date%TYPE;
    v_salary       employee.salary%TYPE;
BEGIN
    SELECT      hire_date, salary
    INTO        v_hire_date,v_salary
    FROM        employees
    WHERE       employee_id=100;
    ....
END;
/
```

Contoh ketiga berikut menggunakan fungsi group yaitu SUM untuk mendapatkan jumlah total *salary* dari semua *employee* :

```
SET SERVEROUTPUT ON
DECLARE
    v_sum_sal      NUMBER(10,2);
    v_deptno       NUMBER NOT NULL :=60;
BEGIN
    SELECT  SUM(salary)
    INTO    v_sum_sal
    FROM    employees
    WHERE   department_id=v_deptno;
    DBMS_OUTPUT.PUT_LINE('The sum salary is ' || TO_CHAR(v_sum_sal));
END;
/
```

4.4. Manipulasi Data dengan PL/SQL

Untuk memanipulasi data yang ada dalam database, digunakan perintah DML (Data Manipulation Language) :

- INSERT : menambahkan baris baru
- UPDATE : memodifikasi baris yang sudah ada
- DELETE : menghapus baris yang tidak diinginkan
- MERGE : memilih data dari satu table kemudian mengubah atau menyisipkan data tersebut ke table yang lain.

4.5. Menambah Data

Perintah INSERT digunakan untuk menyisipkan data baru.

Berikut ini contoh penggunaan perintah INSERT :

```
BEGIN
    INSERT INTO employees
    (employee_id, first_name, last_name, email, hire_date, job_id, salary)
    VALUES
    (employee_seq.nextval,'Ruth','Cores','RCORES',sysdate, 'AD_AST',4000);
END;
/
```

4.6. Mengubah Data

Perintah UPDATE digunakan untuk memodifikasi baris yang sudah ada.

Berikut ini contoh penggunaan perintah UPDATE :

```
DECLARE
  V_sal_increase   employees.salary%TYPE := 800;
BEGIN
  UPDATE employees
  SET      salary=salary + v_sal_increase;
  WHERE   job_id='ST_CLERK';
END;
/
```

4.7. Menghapus Data

Perintah DELETE digunakan untuk menghapus baris yang tidak diinginkan.

Berikut ini contoh penggunaan perintah DELETE :

```
DECLARE
  V_deptno         employees.department_id%TYPE := 10;
BEGIN
  DELETE FROM      employees
  WHERE            department_id = v_deptno;
END;
/
```

4.8. Menggabung Baris Data

Perintah MERGE digunakan untuk memilih data dari satu table kemudian mengubah atau menyisipkan data tersebut ke table yang lain

Berikut ini contoh penggunaan perintah MERGE :

```
DECLARE
  v_empno   employees.employee_id%TYPE := 100;
BEGIN
  MERGE INTO copy_emp c
        USING employees e
        ON (e.employee_id = v_empno)
  WHEN MATCHED THEN

        UPDATE SET
          c.first_name = e.first_name,
          c.last_name = e.last_name,
          c.email = e.email,
          .....
  WHEN NOT MATCHED THEN
    INSERT VALUES(e.employee_id, e.first_name, e.last_name,
                  ....., e.department_id);
END
```

4.9. Aturan Penamaan

Berikut ini aturan penamaan dalam PL/SQL :

Identifier	Aturan Penamaan	Contoh
Variable	v_name	v_sal
Konstanta	c_name	c_increase_sal
Cursor	name_cursor	emp_cursor
Exception	e_name	e_too_many
Table type	name_table.type	emp_table_type
Table	name_table	employees
Record type	name_record.type	emp_record_type
Record	name_record	customer_record
Variabel substitusi dalam SQL*Plus	p_name	p_sal
Host atau bind variable	g_name	g_year_sal

4.10. SQL Cursor

Pada saat suatu SQL Statement diberikan oleh user, Oracle server akan membuka suatu area pada memory untuk menyediakan proses *parse* dan *execute* dari SQL Statement tersebut. Area pada memory tadi disebut dengan *cursor*.

Ada dua macam cursor, yaitu : implicit cursor dan explicit cursor

4.11. Atribut dari SQL Cursor

Dengan menggunakan attribut dari SQL Cursor, kita dapat menguji hasil dari SQL Statement.

Berikut ini atribut dari SQL Cursor :

Nama Atribut	Kegunaan
SQL%ROWCOUNT	Jumlah baris data yang dihasilkan (nilai integer)
SQL%FOUND	Atribut boolean yang bernilai TRUE jika ada satu baris data atau lebih.yang dihasilkan
SQL%NOTFOUND	Atribut boolean yang bernilai TRUE jika tidak ada satu baris data pun.yang dihasilkan
SQL%ISOPEN	Selalu bernilai false karena PL/SQL akan menutup implicit cursor begitu selesai dieksekusi.

Berikut ini contoh dari penggunaan Cursor :

```
VARIABLE rows_deleted VARCHAR2(30)
DECLARE
    v_employee_id    employees.employee_id%TYPE := 176;
BEGIN
    DELETE FROM    employees
    WHERE    employee_id=v_employee_id;
    :rows_deleted := (SQL%ROWCOUNT || 'row deleted');
END;
/
PRINT rows_deleted
```

4.12. Ringkasan

- Telah dipelajari penggunaan SQL Statement dalam PL/SQL melalui perintah SELECT INSERT, UPDATE, DELETE dan MERGE
- Terdapat dua tipe cursor : implicit dan explicit

- Untuk menguji hasil dari dari DML Statement, dapat digunakan atribut cursor :
 - SQL%ROWCOUNT
 - SQL%FOUND
 - SQL%NOTFOUND
 - SQL%ISOPEN

4.13. Latihan Soal

1. Buat program untuk menampilkan nomer maksimum dari department_id (gunakan fungsi MAX) , setelah itu simpan hasilnya dalam bind variable *g_max_deptno* , dan tampilkan hasilnya.
2. Modifikasi program soal nomer 1 dengan menyisipkan informasi department baru ke dalam table DEPARTMENTS.
 - a. Gunakan perintah DEFINE untuk nama department. Beri nama department yang baru ini *Education*.
 - b. Buat program untuk menyisipkan data baru dengan nilai yang telah dibuat sebelumnya dimana nomer department sama dengan *g_max_deptno* ditambah 10 (jadi 270 + 10 = 280), nama department sama dengan soal 2a, dan isi kolom *location_id* dengan nilai NULL.
 - c. Tampilkan data department yang baru dibuat.
3. Modifikasi data yang dibuat pada soal nomer 2b, untuk data dengan *department_id*=280 (nilai 280 dari soal 2b), set *location_id*=1700.
4. Buat program untuk menghapus data department yang dibuat pada soal nomer 2 (*department_id*=280). Kemudian tampilkan hasil dari proses penghapusan data dengan menggunakan atribut cursor SQL%ROWCOUNT. Periksa hasil penghapusan dengan memberikan SQL Statement.