7. WRITING EXPLICIT CURSORS

Objektif :

Setelah mengikuti materi ini, diharapkan dapat :

1. Memiliki pemahaman perbedaan Kursor Implisit dan Kursor Explisit
2. Dapat menggunakan variabel record PL/SQL
3. Dapat menggunakan kursor FOR LOOP [perulangan]

7.1. Sekilas Tentang Kursor

Dalam SQL, terdapat 2 jenis kursor:

- Kursor Implisit : tidak perlu deklarasi dan menghasilkan hanya satu baris data.
- Kursor Eksplisit : perlu deklarasi dari user dan dapat menghasilkan nilai lebih dari satu baris data.

7.2. Kursor Implisit

Oracle Server secara implisit membuka kursor pada setiap proses SQL yang dijalankan menggunkan *script PL/SQL* untuk menampikan data. Kursor implisit tidak memerlukan deklarasi kursor dan nilai yang dihasilkan hanya satu baris data saja.

Contoh program kursor implisit:

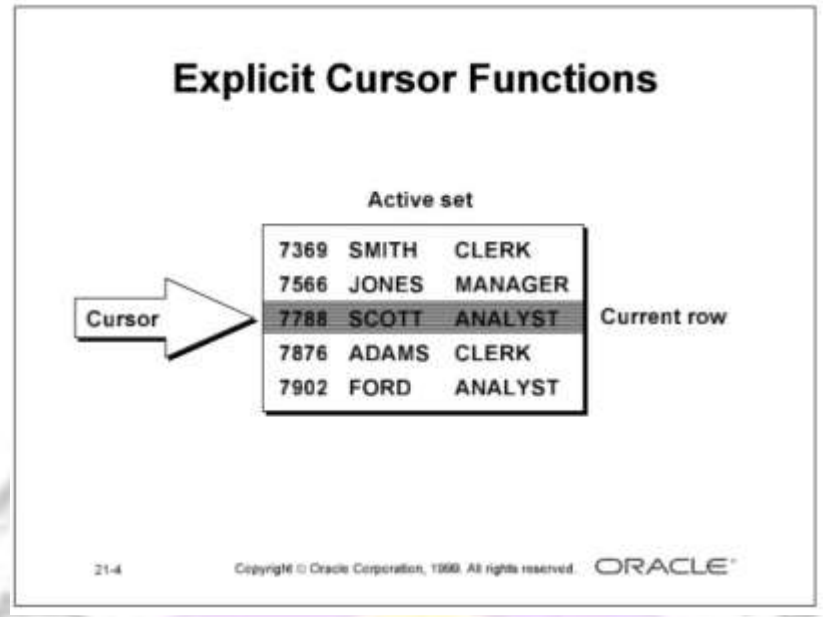
```
set serveroutput on

declare
        pegawai emp.ename%type;
begin
        select ename into pegawai from emp where empno = 7369;
        dbms_output.put_line('PEGAWAI BERNAMA '||pegawai||' BEKERJA SEBAGAI PRAMUNIAGA');
end;
/
```

Lihat Video

7.3. Kursor Eksplisit

Kursor eksplisit digunakan untuk menampilkan data yang mempunyai nilai lebih dari satu baris data. Data tersebut kemudian ditampung dalam sebuah ‘wadah’ yang disebut *active set* seperti terlihat dalam gambar di bawah.



.

Untuk membuat sebuah kursor eksplisit, perlu dilakukan :

1. **Declare**, deklarasikan terlebih dahulu variabel kursor beserta dengan nama variabel kursor (bebas).

   Bentuk umum:

```
CURSOR cursor_name IS
     select_statement;
```

   Contoh :

```
declare
        cursor cur_peg is
        select * from emp;
```

2. **Open**, menandakan bahwa kursor siap untuk dibuka dan dijalankan untuk membuat sebuah *avtive set* yang berfungsi menampung hasil query yang akan dijalankan.

   Bentuk Umum:

```
OPEN  cursor_name;
```

Contoh:

```
open cur_peg;
```

3. **Fetch**, melakukan pengambilan data sesuai dengan baris yang telah ditentukan. Apabila kondisi telah terpenuhi dan TIDAK ADA baris yang akan diproses, kursor harus ditutup.

   Bentuk Umum:

```
FETCH cursor_name INTO [variable1, variable2, ...]
                       | record_name];
```

   Contoh:

```
open cur_peg;
loop
        exit when cur_peg%notfound;
        fetch cur_peg into pegawai;
        . . . .
```

4. **Close**, untuk menutup kursor yang aktif, sehingga memungkinkan membuka kembali kursor yang lain untuk membuat sebuah *active set* yang baru.

   Bentuk Umum:

```
CLOSE    cursor_name;
```

   Contoh:

```
close cur_peg;
```

7.4. Atribut Kursor Eksplisit

- **%ISOPEN**: Mengembalikan nilai TRUE apabila kursor dalam keadaan *open*.

Contoh

```
begin
if not cur_peg%isopen then
        open cur_peg;
        loop
                fetch cur_peg into . . .
```

- **%NOTFOUND** : Memberikan nilai TRUE apabila proses *fetch* sudah tidak mengembalikan sebuah nilai. Gunakan %NOTFOUND untuk menentukan kapan waktu untuk keluar dari sebuah perulangan (loop).

Contoh

```
open cur_peg;
loop
        exit when cur_peg%notfound;
        fetch cur_peg into . . .
```

- **%FOUND** : Memberikan nilai TRUE apabila proses *fetch* mengembalikan sebuah nilai. Kebalikan dari %NOTFOUND.

Contoh

```
fetch cur_peg into pegawai;
if cur_peg%found then
. . . .
end if;
```

- **%ROWCOUNT** : Mengembalikan jumlah baris yang dihasilkan dari proses *fetch* berupa angka.

Contoh

```
dbms_output.put_line('Total pegawai: '||cur_peg%rowcount||' orang');
```

7.5. Kursor Dan Record

```
set serveroutput on

declare
        cursor cur_peg is
          select * from emp;
        pegawai cur_peg%rowtype;
begin

        open cur_peg;
        loop
                exit when cur_peg%notfound;
                fetch cur_peg into pegawai;
                if cur_peg%found then
                dbms_output.put_line('pegawai '||pegawai.ename||' pekerjaannya adalah '||pegawai.job);
                end if;
        end loop;
        dbms_output.put_line('Total pegawai: '||cur_peg%rowcount||' orang');
        close cur_peg;
end;
/
```

Lihat Video

Dalam script PL/SQL di atas, telah terlihat bahwa dapat dideklarasikan sebuah *records* untuk menggunakan struktur kolom dalam tabel. Hal lainnya adalah dapat mendeklarasikan *records* sesuai dengan kolom yang telah ditetapkan di dalam kursor eksplisit. Dalam hal ini, kursor akan melakukan *fetch* sesuai dengan query yang diminta dan dimasukkan ke dalam sebuah *record*.

7.6. Kursor For Loops

```
set serveroutput on

declare
        cursor cur_peg is
          select * from emp;
begin
        For pegawai IN cur_peg LOOP
        -- open dan fetch cursor, terjadi secara implisit disini
                if pegawai.deptno = 20 then
                dbms_output.put_line('pegawai '||pegawai.ename||' bekerja di RESEARCH DEPT.');
                end if;
        end loop; -- close cursor, terjadi secara implisit disini
end;
/
```

Lihat Video

Kursor FOR melakukan perulangan baris di dalam kursor eksplisit. Ini merupakan cara termudah karena **kursor terbuka, kursor melakukan fetch sekali setiap iterasi, dan kursor tertutup secara otomatis** ketika semua baris telah diproses. Perulangan itu sendiri berakhir ketika iterasi sudah mencapai fetch paling akhir.

PETUNJUK:

- Jangan melakukan deklarasi *record* pada FOR LOOP. Karena *record* terdeklarasi secara implisit.
- Lakukan test beberapa atribut kursor apabila dibutuhkan.

Script PL/SQL di atas adalah sebuah contoh query untuk menampilkan data pegawai yang bekerja di Departement Sales.

7.7. Kursor For Loops Menggunakan Subqueries

```
set serveroutput on

begin
        For pegawai IN (select * from emp) LOOP
        -- open dan fetch cursor, terjadi secara implisit disini
                if pegawai.deptno = 20 then
                dbms_output.put_line('pegawai '||pegawai.ename||' bekerja di RESEARCH DEPT.');
                end if;
        end loop; -- close cursor, terjadi secara implisit disini
end;
/
```

Lihat Video

Disini, TIDAK PERLU melakukan deklarasi kursor karena PL/SQL memperbolehkan untuk melakukan substitusi sebuah subqueri. Contoh di atas adalah query yang menghasilkan output sama persis dengan contoh sebelumnya, hanya berbeda penggunaan subqueri.

.