

## PERTEMUAN 13

### SEQUENCE, INDEX & SYNONYM

#### Tujuan Pembelajaran :

- Dapat membuat Sequence
- Dapat melakukan modifikasi dan menghapus Sequence
- Dapat membuat Index dan melakukan pemeliharaan pada Index
- Dapat membuat private dan public Synonim

#### TEORI DAN PERCOBAAN

##### 13.1. Definisi dan Pembuatan Sequence

Apa itu **sequence**?

- Secara otomatis mengenerate bilangan secara unik
- Object yang bisa dipakai bersama
- Biasanya digunakan untuk keperluan penyediaan PRIMARY KEY

Sequence dibuat dengan perintah **CREATE SEQUENCE.**

Berikut ini sintak umum Sequence :

```
CREATE SEQUENCE sequence  
    [INCREMENT BY n]  
    [START WITH n]  
    [{MAXVALUE n | NONMAXVALUE}  
    [{MINVALUE n | NONMINVALUE}  
    [{CYCLE | NONCYCLE}  
    [{CACHE n | NOCACHE }];
```

Percobaan 1 :

Buat sequence DEPT\_DEPTNO yang akan digunakan sebagai PRIMARY KEY dalam table DEPT. Jangan gunakan pilihan CYCLE.

```
SQL> CREATE SEQUENCE dept_deptno
  2 INCREMENT BY 1
  3 START WITH 91
  4 MAXVALUE 100
  5 NOCACHE
  6 NOCYCLE;
```

Sequence created.

### 13.2. Pemeriksa keberadaan Sequence

Untuk memeriksa keberadaan dari sequence, informasi bisa diambil dari data dictionary USER\_SEQUENCES

Percobaan 2 : Tampilkan informasi tentang keberadaan Sequence

```
SQL> SELECT sequence_name, min_value, max_value, increment_by, last_number
  2 FROM user_sequences;
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
CUSTID	1	1.0000E+27	1	109
DEPT_DEPTNO	1	100	1	91
ORDID	1	1.0000E+27	1	622
PRODID	1	1.0000E+27	1	200381

### 13.3. NEXTVAL, CURRVAL dan Penggunaan Sequence

Next sequence ditempatkan pada **NEXTVAL**, sedangkan **CURRVAL** menyimpan current sequence.

Percobaan 3 : Gunakan sequence dept\_deptno untuk mengisi baris baru untuk department 'MARKETING' yang berlokasi di 'SAN DIEGO'

```
SQL> INSERT INTO dept(deptno, dname, loc)
  2 VALUES (dept_deptno.NEXTVAL, 'MARKETING', 'SAN DIEGO');
```

1 row created.

Untuk melihat nilai dari sequence `dept_deptno` saat ini :

```
SQL> SELECT dept_deptno.CURRVAL
2 FROM DUAL;

CURRVAL
-----
      91
```

#### 13.4. Modifikasi dan Penghapusan Sequence

Perintah `ALTER SEQUENCE nama_sequence` bisa digunakan untuk memodifikasi sequence, misal merubah increment value, maximum value, pilihan cycle, atau cache.

*Percobaan 4* : Modifikasi sequence `dept_deptno`

```
SQL> ALTER SEQUENCE dept_deptno
2 INCREMENT BY 1
3 MAXVALUE 999999
4 NOCACHE
5 NOCYCLE;

Sequence altered.
```

Untuk menghapus sequence digunakan perintah :

`DROP SEQUENCE nama_sequence`

Sekali dihapus, sequence tidak bisa direferensi lagi.

*Percobaan 5* : Menghapus sequence `dept_deptno`

```
SQL> DROP SEQUENCE dept_deptno;

Sequence dropped.
```

#### 13.5. Definisi dan Pembuatan Index

Apa itu `index` ?

- `Index` adalah skema object
- Digunakan oleh Oracle server untuk meningkatkan kecepatan untuk mendapatkan baris data yang diinginkan dengan menggunakan pointer

- Dapat mereduksi disk I/O dengan menggunakan metode pengaksesan untuk melokasikan data secara cepat.
- Independent dari table yang diindeks
- Digunakan dan dipelihara secara otomatis oleh Oracle server.

Bagaimana cara indeks dibuat :

- Secara otomatis : Index yang unik dibuat secara otomatis pada saat mendefinisikan constraint PRIMARY KEY atau UNIQUE dalam definisi table.
- Secara manual : User dapat membuat index non-unik pada kolom yang ada untuk meningkatkan kecepatan akses.

Untuk membuat index pada satu atau lebih kolom, sintak penulisannya :

```
CREATE INDEX index
ON table {column[, ... column] ...};
```

Percobaan 6 : Buat index pada kolom *ename* yang ada pada table EMP.

```
SQL> CREATE INDEX emp_ename_idx
2 ON emp(ename);

Index created.
```

### 13.6. Kapan perlu dan tidak perlu membuat Index

Index perlu dibuat pada saat :

- Kolom sering digunakan dalam klausa WHERE atau kondisi join
- Kolom berisi jangkauan nilai yang sangat luas
- Kolom berisi banyak sekali nilai NULL
- Dua atau lebih kolom sering digunakan bersama-sama dalam klausa WHERE atau kondisi join
- Table berukuran besar dan baris yang didapatkan pada saat query paling banyak diperkirakan kurang dari 2-4% dari baris yang ada.

Index tidak perlu dibuat jika :

- Tabel berukuran kecil
- Kolom tidak terlalu sering digunakan sebagai kondisi dalam query

- Baris yang didapatkan pada saat query lebih dari 2-4% dari baris yang ada.
- Tabel sering di-update

### 13.7. Melakukan pemeriksaan terhadap keberadaan Index

Untuk mengetahui keberadaan index, dapat dengan menggunakan data dictionary **USER\_INDEXES** dan **USER\_IND\_COLUMNS**.

*Percobaan 7 :* Periksa keberadaan index dari data dictionary

```
SQL> column column_name format a15
SQL> SELECT ic.index_name, ic.column_name,
2         ic.column_position col_pos, ix.uniqueness
3 FROM user_indexes ix, user_ind_columns ic
4 WHERE ic.index_name=ix.index_name
5 AND ic.table_name='EMP';
```

INDEX_NAME	COLUMN_NAME	COL_POS	UNIQUENES
EMP_PRIMARY_KEY	EMPNO	1	UNIQUE
EMP_ENAME_IDX	ENAME	1	NONUNIQUE

### 13.8. Menghapus Index

Untuk menghapus index dari data dictionary digunakan perintah **DROP INDEX**.

*Percobaan 8 :* Hapus index *emp\_ename\_idx* pada table EMP.

```
SQL> DROP INDEX emp_ename_idx;
Index dropped.
```

### 13.9. Definisi dan Synonim

Synonym dibuat dengan tujuan menyederhanakan akses ke object, dengan cara :

- Merujuk ke table yang dimiliki oleh user lain
- Memperpendek nama object yang panjang.

### 13.10. Pembuatan dan Penghapusan Synonim

Sintak umum pembuatan synonym :

```
CREATE [PUBLIC] SYNONYM synonym
FOR object;
```

*Percobaan 9 :* Beri nama view DEPT\_SUM\_VU dengan nama *d\_sum*

```
SQL> CREATE SYNONYM d_sum  
2 FOR dept_sum_vu;  
  
Synonym created.
```

Untuk **menghapus** synonym digunakan perintah **DROP SYNONYM** . **Hanya DBA** yang bisa menghapus public synonym.

*Percobaan 10 :* Hapus synonym *d\_sum*

```
SQL> DROP SYNONYM d_sum;  
  
Synonym dropped.
```

### LATIHAN SOAL

1. Buat sequence untuk digunakan sebagai PRIMARY KEY pada table DEPARTMENT. Sequence dimulai dari 60, dan mempunyai nilai maksimum 200. Sequence di-increment dengan angka 10. Nama sequence yang dibuat adalah DEPT\_ID\_SEQ.
2. Tampilkan keberadaan sequence dari dictionary USER\_SEQUENCES

```
SQL> select sequence_name,max_value,increment_by,last_number  
2 from user_sequences;
```

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
CUSTID	1.0000E+27	1	109
DEPT_ID_SEQ	200	10	60
ORDID	1.0000E+27	1	622
PROPID	1.0000E+27	1	200381

3. Buat non-unique index pada kolom FOREIGN KEY yang ada pada table EMPLOYEE.
4. Tampilkan index yang ada dalam data dictionary untuk table EMPLOYEE

INDEX_NAME	TABLE_NAME	UNIQUENES
EMPLOYEE_DEPT_ID_IDX	EMPLOYEE	NONUNIQUE
EMPNO_PK	EMPLOYEE	UNIQUE