

4. OTHER DATABASE OBJECTS

Objektif

Setelah mengikuti materi ini, diharapkan dapat :

1. Menjelaskan beberapa obyek database dan kegunaannya
2. Membuat, memelihara dan menggunakan sequence
3. Membuat dan memelihara index
4. Membuat private synonym dan public synonym



4.1. Obyek-Obyek Database

Dalam Oracle, terdapat beberapa obyek-obyek:

Obyek	Keterangan
Tabel	Unit penyimpanan dasar, terdiri dari baris dan kolom
View	Secara logika mewakili subset data untuk 1 atau lebih tabel
Sequence	Membentuk nilai primary key
Index	Meningkatkan kualitas kerja dari query
Synonym	Nama alternatif dari suatu obyek

Tabel Obyek-Obyek Database

4.2. Sequence

Sequence adalah obyek database yang dibuat oleh user, dan dapat diakses banyak user untuk membuat integer unik. Kegunaan utama sequence adalah untuk membuat nilai primary key, yang nilainya harus unik untuk setiap baris. Sequence dibuat dan ditambahkan (dikurangi) oleh sebuah routine Oracle8 internal. Jumlah sequence disimpan dan dibuat terpisah dari tabel oleh karena itu, sebuah sequence dapat digunakan oleh banyak table

Membuat Sebuah Sequence

Bentuk umum membuat sebuah sequence:

```
SQL> CREATE SEQUENCE sequence_name
2  [INCREMENT BY n]
3  [START WITH n]
4  [{MAXVALUE n | NOMAXVALUE}]
5  [{MINVALUE n | NOMINVALUE}]
6  [{CYCLE | NOCYCLE}]
7  [{CACHE n | NOCACHE}];
```

Contoh membuat sebuah sequence:

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

Sequence created.
```

Lihat Video

Contoh diatas adalah cara membuat sequence yang diberi nama DEPT_DEPTNO akan digunakan untuk kolom DEPTNO pada tabel DEPT. Sequence dimulai dari 91, tidak disimpan dalam memori dan berulang.

Konfirmasi Sequence

Pada saat pertama kali membuat sequence, maka sequence tersebut akan didokumentasikan di dalam kamus data. Selama sequence merupakan obyek dari database, maka dapat sequence diidentifikasi dalam kamus data USER_OBJECTS.

Untuk menkonfirmasi pengaturan sequence dapat memilih tabel USER_SEQUENCES dari kamus data.

Contoh mengkonfirmasi sebuah sequence:

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

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_DEPTNO	1	100	1	91

Lihat Video

Pseudocolumn NEXTVAL dan CURRVAL

Pseudocolumn NEXTVAL digunakan untuk menambahkan sequence berturut-turut dari sequence khusus. NEXTVAL harus diberikan pada nama sequence. Ketika sequence NEXTVAL direferensikan, angkaurut baru dibentuk dan angkaurut tersebut ditempatkan dalam CURRVAL. Pseudocolumn CURRVAL digunakan untuk acuan angka sequence yang baru saja dibentuk. NEXTVAL digunakan untuk membentuk angkaurut dalam current user's session sebelum CURRVAL dapat direferensikan. CURRVAL harus diberikan nama sequence. Ketika sequence CURRVAL direferensikan, nilai terakhir dikembalikan ke user proses akan ditampilkan.

Aturan Penggunaan NEXTVAL dan CURRVAL

NEXTVAL dan CURRVAL dapat digunakan didalam hal berikut ini:

- Daftar SELECT dari statement SELECT yang bukan bagian dari subquery.
- Daftar SELECT dari subquery dalam statement INSERT.
- Klausa VALUES dari statement INSERT.
- Klausa SET dari statement UPDATE.

NEXTVAL dan CURRVAL tidak dapat digunakan didalam hal berikut ini:

- Daftar SELECT dari view.
- Statement SELECT dengan menggunakan keyword DISTINCT.
- Statement SELECT dengan klausa GROUP BY, HAVING atau ORDER BY.
- Subquery dalam statement SELECT, DELETE atau UPDATE.
- Ekspresi DEFAULT dalam statement CREATE TABLE atau ALTER TABLE.

Menggunakan Sequence

Contoh menggunakan sebuah sequence:

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

1 row created.
```

Lihat Video

Contoh di atas adalah menyisipkan departemen baru dalam tabel DEPT. Menggunakan sequence DEPT_DEPTNO untuk membentuk angka departemen baru tersebut.

Untuk dapat menampilkan nilai sequence yang ada maka gunakan query sebagai berikut:

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

CURRVAL
-----
      91
```

Lihat Video

Menyimpan Nilai Sequence

Menyimpan sequence ke dalam memori berguna untuk mempercepat akses ke nilai sequence tersebut. Penyimpanan tersebut disatukan pada saat referensi pertama kali ke suatu sequence. Setiap permintaan untuk nilai sequence berikutnya diambil dari sequence yang disimpan. Setelah sequence terakhir digunakan, permintaan selanjutnya untuk sequence memasukkan penyimpanan sequence lainnya kedalam memory.

Memodifikasi Sequence

Jika sequence telah mencapai batas MAXVALUE, tidak ada nilai tambahan dari sequence yang akan dialokasikan dan akan ditampilkan error yang menunjukkan bahwa sequence tersebut melebihi MAXVALUE. Untuk melanjutkan penggunaan sequence, sequence dapat dimodifikasi dengan menggunakan statement ALTER SEQUENCE.

Bentuk umum memodifikasi sequence:

```
SQL> ALTER SEQUENCE sequence_name
2   [INCREMENT BY n]
3   [{MAXVALUE n | NOMAXVALUE}]
4   [{MINVALUE n | NOMINVALUE}]
5   [{CYCLE | NOCYCLE}]
6   [{CACHE n | NOCACHE}];
```

Contoh memodifikasi sequence:

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

Sequence altered.
```

[Lihat Video](#)

Petunjuk Untuk Memodifikasi Sequence

- Merupakan seorang pemilik sequence atau memiliki hak akses ALTER untuk sebuah sequence.
- Hanya angka sequence selanjutnya yang akan dibuat dengan statement ALTER SEQUENCE.

- Option START WITH tidak dapat diubah menggunakan ALTER SEQUENCE. Sequence harus dihapus dan dibuat ulang untuk memulai kembali sequence pada angka yang berbeda.
- Beberapa validasi dibentuk. Sebagai contoh, MAXVALUE baru tidak dapat ditentukan lebih kecil dari nilai sequence sekarang.

Contoh MAXVALUE baru tidak dapat ditentukan lebih kecil dari nilai sequence sekarang:

```
SQL> ALTER SEQUENCE dept_deptno
2 INCREMENT BY 1
3 MAXVALUE 90
4 NOCACHE
5 NOCYCLE;
ALTER SEQUENCE dept_deptno
*
ERROR at line 1:
ORA-04009: MAXVALUE cannot be made to be less than the current value
```

Lihat Video

Menghapus Sequence

Untuk menghapus sequence dari kamus data, gunakan statement DROP SEQUENCE. Untuk menghapus sequence seseorang harus menjadi pemilik sequence atau memiliki hak akses DROP ANY SEQUENCE.

Bentuk umum menghapus sequence:

```
SQL> DROP SEQUENCE sequence_name;
```

4.3. Index

Sebuah Index Oracle Server adalah obyek schema yang dapat meningkatkan pengambilan baris dengan menggunakan suatu pointer. Index dapat dibuat secara explicit atau otomatis. Jika tidak ada suatu index pada kolom, maka pembacaan tabel secara keseluruhan akan terjadi.

Sebuah index menyediakan akses langsung dan cepat terhadap baris dalam tabel. Index berguna untuk mengurangi kebutuhan I/O disk dengan menggunakan jalur terindex untuk menempatkan data secara cepat. Index digunakan dan dipelihara secara otomatis oleh Oracle Server. Pertama kali index dibuat, maka tidak ada aktifitas langsung yang disyaratkan oleh user.

Index secara logika dan fisik terpisah dari tabel yang diindex. Hal ini berarti bahwa index dapat dibuat atau dihapus kapanpun dan tidak berefek pada tabel dasar atau index lainnya.

Catatan : Ketika tabel dihapus, index yang berkorespondensi akan ikut terhapus juga.

Membuat Index

Dua tipe index yang dapat dibuat. Tipe pertama adalah unik index. Oracle Server secara otomatis membuat index ini ketika kolom didefinisikan pada sebuah tabel untuk mempunyai constraint PRIMARY KEY atau UNIQUE KEY. Nama index adalah nama yang diberikan untuk constraintnya.

Tipe index yang lain adalah user dapat membuat non-unik index. Sebagai contoh, ketika membuat index kolom FOREIGN KEY untuk penggabungan dalam query dalam rangka mempercepat pengambilan data. Membuat sebuah index pada satu atau lebih kolom dengan menggunakan statement CREATE INDEX.

Bentuk umum membuat index:

```
SQL> CREATE INDEX index_name  
2 ON table_name(column_name, ....);
```

Contoh membuat index:

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

Lihat Video **Kapan**

Index Dibuat ?

- Kolom sering digunakan didalam klausa WHERE atau dalam kondisi join.
- Kolom mengandung range nilai yang besar.
- Kolom mengandung sejumlah besar nilai null.
- Tabel besar dan banyaknya query yang diharapkan terambil kurang dari 2-4% dari barisbaris.

Perlu diingat bahwa jika ingin menjalankan ketidakunikan, diharuskan untuk mendefinisikan unik constraint dalam pendefinisian tabel. Kemudian unik index dibuat secara otomatis.

Kapan Tidak Perlu membuat Index

- Tabelnya kecil
- Kolom tidak sering digunakan sebagai kondisi dalam query
- Banyaknya query yang diharapkan terambil lebih dari 2-4% dari baris
- Tabel sering diupdate. Jika terdapat satu atau lebih index pada tabel, statement DML yang mengakses tabel tersebut menghabiskan waktu lebih banyak.

Konfirmasi Index

Mengkonfirmasi keberadaan index melalui view kamus data USER_INDEXES. Kolom yang dilibatkan dalam index dengan menquery view juga dapat dilihat.

Contoh megkonfirmasi index:

```
SQL> SELECT ic.index_name, ic.column_name  
2 FROM user_ind_columns ic  
3 WHERE ic.table_name = 'EMP';
```

INDEX_NAME	COLUMN_NAME
PK_EMP	EMPNO
EMP_ENAME_IDX	ENAME

Lihat Video

Contoh di atas menampilkan semua index yang pernah dibuat, pada tabel EMP.

Function-Based Index

Function-based index adalah index yang didasarkan pada ekspresi. Function-based index didefinisikan menggunakan atribut UPPER(nama kolom) atau LOWER(nama kolom) yang memperbolehkan pencarian dengan case-sensitive.

Contoh membuat function_base index:

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

Index created.

Memberikan fasilitas pemrosesan query seperti :


```
SQL> SELECT empno, ename, job
2 FROM emp
3 WHERE UPPER(ename) = 'KING';
```

EMPNO	ENAME	JOB
7839	KING	PRESIDENT

Lihat Video

Untuk meyakinkan, Oracle akan menggunakan index dibandingkan mencari pada keseluruhan tabel, yakinkan bahwa nilai fungsi tidak null dalam subsequent query. Sebagai contoh, statement dibawah menggunakan index, tetapi tanpa klausa Oracle WHERE akan mencari dikeseluruhan tabel.

```
SQL> SELECT empno, ename, job
2 FROM emp
3 WHERE UPPER(ename) IS NOT NULL
4 ORDER BY UPPER(ename);
```

EMPNO	ENAME	JOB
7876	ADAMS	CLERK
7499	ALLEN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER

Lihat Video

Oracle memperlakukan index dengan kolom yang ditandai DESC sebagai function-based indexes. Kolom yang ditandai DESC akan diurutkan secara descending.

Menghapus Sebuah Index

Index dapat dimodifikasi. Untuk merubah sebuah index, index harus dihapus dan kemudian dibuat kembali. Hapus definisi index dari kamus data dengan menggunakan statement DROP INDEX. Untuk menghapus sebuah index, seseorang harus memiliki hak akses index tersebut DROP ANY INDEX.

Bentuk umum menghapus sebuah index:

```
SQL> DROP INDEX index_name;
```

4.4. Synonym

Untuk menggunakan tabel yang dimiliki oleh user lain, perlu ditambahkan nama tabel dengan nama user yang memiliki obyek tersebut diikuti oleh waktunya. Pembuatan synonym menghilangkan kebutuhan akan kualifikasi nama obyek dengan schema dan memberikan nama alternatif untuk tabel, view, sequence, procedure, atau obyek lainnya. Metode ini dapat secara khusus digunakan untuk nama obyek yang panjang, seperti view.

Bentuk umum membuat sebuah synonym:

```
SQL> CREATE SYNONYM synonym_name  
2 FOR object_name;
```

Petunjuk :

- Obyek tidak dapat berada didalam paket
- Nama private synonym harus berbeda dari semua obyek lain yang dimiliki oleh user yang sama

Menghapus Synonym

Untuk menghapus synonym, gunakan statement DROP SYNONYM. Hanya DBA yang dapat menghapus public synonym.

Bentuk umum menghapus sebuah synonym:

```
SQL> DROP SYNONYM synonym_name;
```