

1. MEMBUAT DAN MENGELOLA TABEL

Objektif :

Setelah menyelesaikan materi ini, mahasiswa dapat melakukan hal sebagai berikut :

1. Membuat tabel
2. Mengetahui tipe data yang dapat digunakan pada pendefinisian kolom
3. Dapat menggunakan statement Alter Table
4. Dapat melakukan Drop, Rename, dan Truncate Table



1.1. Data Object

Sebuah database oracle dapat terdiri dari beberapa struktur data. Setiap strukturnya harus tercantum dalam database design sehingga dapat dibuat selama tahap pembuatan database.

- Table* : menyimpan data
View : subset data dari satu atau lebih table
Sequence : Menghasilkan nilai primary key
Index : meningkatkan kinerja *query*
Synonym : memberikan alternatif nama obyek database

1.2. Struktur Table Oracle

Table dapat dibuat kapan saja, bahkan pada saat beberapa user sedang menggunakan database. Ukuran table tidak perlu dispesifikasi. Ukuran table ditentukan oleh jumlah ruang yang dialokasikan untuk database secara keseluruhan.

Aturan Penamaan

Penamaan table dan kolom sesuai dengan aturan penamaan standar yang digunakan untuk obyek database Oracle lainnya :

- Nama table dan kolom harus dimulai dengan huruf dan panjangnya dapat mencapai 30 karakter.
- Nama table hanya terdiri dari karakter A – Z, a – z, 0 – 9, _ (*underscore*), \$, dan # (namun penggunaannya tidak disarankan)
- Nama suatu table tidak boleh sama dengan nama obyek database lainnya yang berada di satu Oracle server.
- Nama table tidak boleh sama dengan *reserved word* Oracle

Petunjuk Penamaan

- Gunakan nama deskriptif untuk table maupun obyek database lainnya.

- Gunakan nama yang sama untuk satu entitas di table yang berbeda. Sebagai contoh, kolom department number diberi nama DEPTNO pada table EMP dan table DEPT.

Catatan : nama tidak bersifat *case sensitive*. Sebagai contoh, EMP dianggap sama dengan eMP ataupun eMp.

1.3. Create Table Statement

Pembuatan tabel dilakukan dengan menjalankan perintah SQL CREATE TABLE. perintah ini termasuk dalam *data definition language* (DDL). Perintah ini memberikan pengaruh langsung pada database, dan menyimpan informasi pada *data dictionary*.

Untuk membuat tabel, user harus memiliki hak untuk CREATE TABLE dan *storage area* tempat penyimpanan obyek.

Sintaks :

```
CREATE [ GLOBAL TEMPORARY] TABLE [schema] table
( column      datatype      [DEFAULT expr], [.....]);
```

dengan :

GLOBAL TEMPORARY menspesifikasikan bahwa table bersifat sementara dan definisinya dapat dipakai oleh semua session. Data yang terdapat pada temporary table hanya dapat digunakan pada session yang memasukkan data tersebut ke dalam table.

schema sama dengan nama pembuat table

nama table

DEFAULT expr menspesifikasikan nilai default jika nilainya dihilangkan dari statement insert

column nama kolom datatype

tipe dan panjangnya data kolom **Contoh :**

Membuat tabel mata kuliah dengan kolom id mata kuliah dan nama mata kuliah

```
SQL> create table mata_kuliah(  
2 id_matkul number(5) not null,  
3 matkul varchar2(50) not null);
```

Table created.

1.4. Querying Data Dictionary

Ada sekumpulan tabel dan view lain pada database Oracle yang dikenal sebagai *data dictionary*. *Data dictionary* dibuat dan dipelihara oleh server Oracle dan berisi informasi database.

Tabel dasarnya jarang diakses karena informasi yang terkandung didalamnya sulit dimengerti. User biasanya mengakses view dari *data dictionary* karena informasinya disajikan dalam format yang lebih mudah untuk dimengerti.

Terdapat empat (4) kategori view data dictionary, setiap kategori memiliki prefix tersendiri yang menunjukkan kegunaannya :

Prefix	Keterangan
USER_	View berisi informasi mengenai obyek-obyek yang dimiliki user
ALL	View berisi informasi mengenai semua table (object table dan relational table) yang dapat diakses oleh user
DBA_	View ini merupakan view terbatas, yang hanya dapat diakses oleh DBA
V\$_	View ini berisi informasi mengenai kinerja dinamis, kinerja database server, dan locking

Melakukan query terhadap tabel *data dictionary* berfungsi untuk melihat obyek database yang sudah ada / tersedia.

Table data dictionary yang sering dipergunakan adalah :

USER_TABLES

USER_OBJECTS

USER_CATALOG

Catatan : USER_TABLES memiliki sinonim yaitu TABS. TABS dapat digunakan sebagai pengganti dari USER_TABLES pada SQL statement **Contoh** :

```
SQL> select table_name from tabs;
```

```
TABLE_NAME
```

```
-----
DEPT
EMP
BONUS
SALGRADE
MATA_KULIAH
```

1.5. Tipe Data

Tipe Data	Deskripsi
VARCHAR2(size)	Data karakter variable length (ukuran maksimum harus disebutkan.) Ukuran minimum dan defaultnya adalah 1, maksimum 4000.
CHAR(size)	Data karakter fixed length dengan ukuran bytes (ukuran minimum dan defaultnya 1, maksimum 2000)
NUMBER(p,s)	Number memiliki presisi p dan skala s (presisi adalah jumlah keseluruhan digit desimal, skala adalah jumlah digit poin desimal. Presisi berkisar dari 1 sampai 38 dan skala berkisar dari – 84 sampai 127)
DATE	Nilai Date dan time mulai dari 1 Januari 4712 B.C sampai 31 Desember 9999 A.D
LONG	Karakter variable length sampai dengan 2 gigabytes
CLOB	Karakter single byte sampai dengan 4 gigabytes
RAW(size)	Data biner Raw (ukuran maksimum harus disebutkan. Ukuran maksimum adalah 2000)
LONG RAW	Data biner Raw variable length sampai dengan 2 gigabytes
BLOB	Data biner sampai dengan 4 gigabytes
BFILE	Data biner tersimpan di external file hingga 4 gigabytes

1.6. Membuat Tabel Dari Baris-Baris Di Tabel Lain

Sintak :

```
CREATE TABLE table [ ( column, column, ... ... ) ] As subquery
```

dengan :

column nama kolom, nilai default, dan integrity constraint
subquery statement SELECT yang mendefinisikan sekumpulan baris
yang dimasukkan ke dalam table yang dibuat

Petunjuk :

- Table dibuat dengan kolom yang disebutkan, dan baris-baris yang diperoleh melalui statement SELECT dimasukkan ke dalam tabel.
- Definisi kolom boleh hanya berisi nama kolom dan nilai default.
- Jika spesifikasi kolom disertakan, jumlah kolom harus sesuai dengan jumlah kolom yang terdapat pada *subquery*.
- Jika spesifikasi kolom tidak disertakan, nama kolom pada table sama dengan nama kolom di *subquery*.

Contoh : membuat tabel employee2 yang terdapat id dan last name dengan mengambil data dan struktur tabel dari tabel emp;

```
SQL> create table employee2 as select empno id, ename last_name from emp;
```

Table created.

```
SQL> select * from employee2;
```

ID	LAST_NAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

14 rows selected.

Setelah membuat table, mungkin perlu mengubah struktur table. Untuk melakukan hal tersebut dapat dengan menggunakan statement ALTER TABLE

1.7. Statement Alter Tabel

Untuk menambah kolom ke sebuah table menggunakan statement ALTER TABLE dengan klausa ADD, dan memodifikasi kolom yang ada menggunakan statement ALTER TABLE dengan klausa MODIFY. Modifikasi kolom meliputi perubahan tipe data, ukuran, dan nilai default kolom.

Sintaks :

```
ALTER TABLE table ADD (column datatype [DEFAULT expr]
[ , column datatype] ... );
```

```
ALTER TABLE table MODIFY (column datatype [DEFAULT expr][ , column
datatype ] ... );
```

dengan :

column nama kolom baru *datatype*
 tipe data dan panjang kolom baru
DEFAULT *expr* menspesifikasikan nilai default untuk kolom baru

Catatan : kolom baru yang ditambahkan berada di urutan terakhir dalam table

Contoh :

Menambahkan kolom job pada table employee2

```
SQL> alter table employee2 add job varchar2(25);
```

```
Table altered.
```

```
SQL> desc employee2;
```

```
Name
```

```
-----
```

```
ID
```

```
LAST_NAME
```

```
JOB
```

Menambahkan jumlah karakter pada kolom job dari 25 menjadi 50

```
SQL> alter table employee2 modify job varchar2(50);  
  
Table altered.
```

1.8. Menghapus Kolom

Untuk menghapus kolom dari sebuah table menggunakan statement ALTER TABLE dengan klausa DROP COLUMN. Fasilitas ini hanya terdapat pada Oracle 8i keatas.

Petunjuk :

- Kolom mungkin kosong atau berisi data
- Kolom hanya dapat dihapus satu persatu
- Table harus memiliki paling tidak 1 kolom setelah dilakukan proses ALTER • Setelah kolom dihapus, tidak dapat dikembalikan seperti semula

Contoh :

Menghapus kolom job pada tabel employee2

```
SQL> alter table employee2 drop column job;  
  
Table altered.
```

```
SQL> desc employee2;  
Name  
-----  
ID  
LAST_NAME
```

1.9. Menghapus Tabel

Statement DROP TABLE menghapus table beserta definisinya, database kehilangan semua data yang ada pada table serta semua indeks yang berhubungan dengannya.

Sintaks :

```
DROP TABLE nama_table;
```

Contoh : menghapus tabel employee

```
SQL> DROP TABLE EMPLOYEE;  
  
Table dropped.
```


Petunjuk :

- Semua data dihapus dari table.
- Semua view dan synonym tetap namun bernilai invalid.
- Semua transaksi tertunda (pending transaction) akan di-commit
- Hanya pembuat table atau user dengan hak DROP ANY TABLE yang dapat menghapus table.

1.10. Mengubah Nama Tabel

Perintah DDL RENAME digunakan untuk mengganti nama *table*, *view*, *sequence*, atau *synonym*.

Perintah RENAME dapat dijalankan jika pemilik dari obyek tersebut.

Sintaks :

```
RENAME old_name TO new_name
```

Contoh :

Mengubah nama tabel employee2 menjadi employee

```
SQL> RENAME EMPLOYEE2 TO EMPLOYEE  
2 ;  
  
Table renamed.
```

1.11. Truncate Table

Statement DDL yang lain adalah TRUNCATE TABLE yang digunakan untuk menghapus semua baris dari suatu tabel dan membebaskan storage yang digunakan tabel tersebut. Setelah menggunakan statement TRUNCATE TABLE, tidak dapat melakukan rollback.

Sintaks :

```
TRUNCATE TABLE table_name;
```

Contoh :

Melakukan perintah TRUNCATE pada tabel employee

```
SQL> TRUNCATE TABLE EMPLOYEE;  
  
Table truncated.  
  
SQL> select * from employee;  
  
no rows selected
```

User harus merupakan pemilik tabel atau memiliki *system privilege* DELETE TABLE untuk dapat melakukan perintah ini. Statement DELETE juga dapat menghapus semua baris pada tabel namun tidak melakukan pengosongan *storage space*.

1.12. Menambahkan Komentar Tabel

Untuk menambahkan komentar pada *table*, *view*, *kolom*, atau *snapshot* hingga 2000 *bytes* dapat menggunakan statement COMMENT. Komentar disimpan dalam data dictionary dan dapat dilihat pada salah satu dari view berikut ini :

```
ALL_COL_COMMENTS  
USER_COL_COMMENTS  
ALL_TAB_COMMENTS USER_TAB_COMMENTS
```

Sintaks :

```
COMMENT ON  TABLE table | COLUMN table.column  
IS 'text' dengan :  
    table      nama table  
column  nama kolom  text  
teks komentar
```

Jika ingin menghapus komentar dari database dapat dengan mengaturnya menjadi string kosong (' ')