

Oracle Academic Initiative

Oracle9i Introduction to SQL



Oleh:

Tessy Badriyah, SKom.MT

**Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Surabaya**

BAB 10 : Pembuatan dan Pengelolaan Tabel

10.1. Sasaran

- Memahami object utama dari database
- Membuat table
- Memahami tipe data yang digunakan pada saat pendefinisian kolom pada table
- Merubah definisi table
- Drop, rename dan truncate table

10.2. Object Database

Object-object Database terdiri dari :

Object	Deskripsi
Table	Unit dasar penyimpanan, terdiri dari baris dan kolom
View	Secara logika merepresentasikan sub himpunan dari data yang berasal dari satu atau lebih table
Sequence	Men-generate nilai primary key
Index	Meningkatkan performansi query
Synonym	Memberi alternatif penamaan pada object

10.3. Aturan Penamaan

Aturan pemberian nama pada table dan kolom :

- Dimulai dengan huruf
- Panjangnya 1-30 karakter
- Hanya diperbolehkan berisi karakter A-Z, a-z, 0-9, _, \$ dan #.
- Bukan merupakan nama dari object yang lain yang dimiliki oleh user yang sama.
- Bukan merupakan keyword dari Oracle server.

10.4. Statement CREATE TABLE

Untuk membuat table dibuat perintah CREATE TABLE.

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

Contoh pembuatan table :

```
CREATE TABLE dept
            (deptno NUMBER(2) ,
            dname  VARCHAR2(14) ,
            loc    VARCHAR2(13) );
Table created.
```

Memeriksa keberadaan table :

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

10.5. Referensi Tabel User yang lain

Tabel yang dimiliki oleh user yang lain, tempatnya tidak berada pada schema dari current user. Untuk mengakses table yang dimiliki oleh user lain, digunakan nama owner, tanda titik kemudian nama table.

10.6. Option DEFAULT

DEFAULT digunakan untuk memberi nilai default pada kolom yang dimaksud.

Contoh : secara default tanggal mulai bekerja adalah sama dengan tanggal system saat data dimasukkan, jadi : ... hiredate DATE DEFAULT SYSDATE, ...

10.7. Pembuatan Tabel

Contoh pembuatan tabel :

```
CREATE TABLE dept
      (deptno NUMBER(2) ,
       dname  VARCHAR2(14) ,
       loc    VARCHAR2(13) );
Table created.
```

Konfirmasi pembuatan tabel, dengan melihat strukturnya menggunakan perintah DESCRIBE :

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

10.8. Tabel dalam Database Oracle

Terdapat 2 (dua) macam table dalam Oracle Database :

1. Tabel User
Yaitu table yang dibuat dan dipelihara oleh user, berisi informasi user.
2. Data Dictionary
Yaitu table yang dibuat dan dipelihara oleh Oracle server, dan berisi informasi database.

10.9. Query dalam Kamus Data (Data Dictionary)

Ada 4 (empat) kategori dalam Data Dictionary :

1. USER_ Berisi object yang dimiliki oleh user.
2. ALL_ Semua object yang dapat diakses oleh user, baik yang dimiliki atau pun yang tidak tapi user punya hak akses atas object yang tidak dimiliki tsb
3. DBA_ User dengan privilege DBA yang dapat mengakses sembarang object dalam database
4. V\$_ Menampilkan performansi server database dan penguncian, hanya dapat diakses oleh DBA.

10.10. Melakukan Query pada Data Dictionary

Untuk melihat nama tabel apa saja yang dimiliki oleh user :

```
SELECT table_name
FROM user_tables ;
```

Untuk menampilkan object distinct yang dimiliki oleh user :

```
SELECT DISTINCT object_type
FROM user_objects ;
```

Untuk menampilkan tabel, view, synonym dan sequence yang dimiliki oleh user :

```
SELECT *
FROM user_catalog ;
```

10.11. Tipe Data

Data Type	Description
VARCHAR2 (<i>size</i>)	Variable-length character data
CHAR (<i>size</i>)	Fixed-length character data
NUMBER (<i>p</i> , <i>s</i>)	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data up to 2 gigabytes
CLOB	Character data up to 4 gigabytes
RAW and LONG RAW	Raw binary data
BLOB	Binary data up to 4 gigabytes
BFILE	Binary data stored in an external file; up to 4 gigabytes
ROWID	A 64 base number system representing the unique address of a row in its table.

10.12. Tipe Data DateTime

Tipe data **TIMESTAMP** adalah perluasan dari tipe data **DATE**. **TIMESTAMP** menyimpan tahun, bulan dan hari dari tipe data **DATE**, ditambah dengan jam, menit dan detik sebagai nilai *fractional second*. Bentuk dari tipe data **TIMESTAMP** :

```
TIMESTAMP [(fractional_seconds_precision)]
```

10.13. Tipe Data **TIMESTAMP WITH TIME ZONE**

TIMESTAMP WITH TIME ZONE adalah variant dari tipe data **TIMESTAMP** yang meliputi time zone. Time zone displacement adalah perbedaan dalam jam dan menit antara local time dengan UTC. Bentuk umumnya :

```
TIMESTAMP [(fractional_seconds_precision)]
WITH TIME ZONE
```

10.14. Tipe Data **TIMESTAMP WITH LOCAL TIME**

TIMESTAMP WITH LOCAL TIME ZONE adalah varian lain dari TIMESTAMP meliputi timezone sebagai nilai tambahan. Data disimpan dalam database yang dinormalisasi terhadap database time zone.

```
TIMESTAMP [(fractional_seconds_precision)]
WITH LOCAL TIME ZONE
```

10.15. Tipe Data INTERVAL YEAR TO MONTH

INTERVAL YEAR TO MONTH menyimpan periode waktu dengan menggunakan YEAR dan MONTH.

```
INTERVAL YEAR [(year_precision)] TO MONTH
```

```
INTERVAL '123-2' YEAR(3) TO MONTH
Indicates an interval of 123 years, 2 months.
```

```
INTERVAL '123' YEAR(3)
Indicates an interval of 123 years 0 months.
```

```
INTERVAL '300' MONTH(3)
Indicates an interval of 300 months.
```

```
INTERVAL '123' YEAR
Returns an error, because the default precision is 2,
and '123' has 3 digits.
```

10.16. Tipe Data INTERVAL DAY TO SECOND

INTERVAL_DAY_TO_SECOND menyimpan periode waktu dalam istilah hari, jam, menit dan detik.

```
INTERVAL DAY [(day_precision)]
TO SECOND [(fractional_seconds_precision)]
```

```
INTERVAL '4 5:12:10.222' DAY TO SECOND(3)
Indicates 4 days, 5 hours, 12 minutes, 10 seconds,
and 222 thousandths of a second. INTERVAL '123' YEAR(3).
```

```
INTERVAL '7' DAY
Indicates 7 days.
```

```
INTERVAL '180' DAY(3)
Indicates 180 days.
```

10.17. Pembuatan Tabel Menggunakan Sintak SubQuery

Sintak untuk membuat table dengan mengkombinasikan CREATE TABLE statement dengan AS subquery.

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

10.18. Pembuatan Tabel menggunakan SubQuery

```
CREATE TABLE dept80
AS
SELECT  employee_id, last_name,
        salary*12 ANNSAL,
        hire_date
FROM    employees
WHERE   department_id = 80;
```

Table created.

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

10.19. Statement ALTER TABLE

Statement ALTER TABLE digunakan untuk :

- Menambahkan kolom baru
- Memodifikasi kolom yang sudah ada
- Mendefinisikan nilai default untuk kolom baru
- Menghapus kolom

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

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

```
ALTER TABLE table
DROP     (column);
```

10.20. Menambahkan Kolom

Sintak dari perintah ALTER TABLE untuk menambah kolom baru pada table :

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

Ilustrasi penambahan kolom baru :

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

Kolom baru

JOB_ID

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Kolom baru ditambahkan dengan menggunakan klausa ADD. Setelah penambahan, kolom baru akan menempati urutan terakhir.

```
ALTER TABLE dept80
ADD      (job_id VARCHAR2(9));
Table altered.
```

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

10.21. Memodifikasi Kolom

Kita dapat memodifikasi kolom dengan mengubah tipe datanya, ukuran dan nilai defaultnya. Sintak dari perintah ALTER TABLE untuk memodifikasi kolom pada Tabel :

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

```
ALTER TABLE dept80
MODIFY      (last_name VARCHAR2(30));
Table altered.
```

10.22. Menghapus Kolom

Klausa DROP COLUMN digunakan untuk menghapus kolom yang tidak diperlukan lagi pada table.

```
ALTER TABLE dept80
DROP COLUMN job_id;
Table altered.
```

10.23. Option SET UNUSED

Option SET UNUSED digunakan untuk menandai satu atau lebih kolom yang tidak digunakan. Untuk menghapus tanda yang sudah diberikan, digunakan perintah DROP UNUSED COLUMNS.

```
ALTER TABLE table
SET UNUSED (column);
OR
ALTER TABLE table
SET UNUSED COLUMN column;
```

```
ALTER TABLE table
DROP UNUSED COLUMNS;
```

10.24. Menghapus Tabel

Melakukan penghapusan table dengan perintah DROP, berarti mengerjakan hal berikut :

- Semua data dan struktur dari table akan dihapus
- Semua transaksi pending akan di-commit
- Semua indeks akan dihapus

```
DROP TABLE dept80;
Table dropped.
```

- Perintah drop ini tidak bisa di-rollback

10.25. Merubah Nama suatu Object

Untuk merubah nama object (table, view, sequence, atau synonym), digunakan perintah RENAME. Untuk melakukan RENAME, user harus merupakan owner (pemilik) dari object yang diubah.

```
RENAME dept TO detail_dept;
Table renamed.
```

10.26. Men-Truncate Tabel

Melakukan penghapusan table dengan perintah TRUNCATE, berarti mengerjakan hal berikut :

- Menghapus semua baris dari table
- Membebaskan kembali ruang penyimpanan yang digunakan oleh table
- Baris yang telah dihapus tidak dapat di-rollback

Sebagai alternatif dari perintah TRUNCATE, supaya baris data yang dihapus dapat di-rollback, maka gunakan DELETE. Perintah DELETE bisa menghapus semua baris data dari table, tapi tetap tidak membebaskan ruang penyimpanan yang digunakan oleh table.

```
TRUNCATE TABLE detail_dept;
Table truncated.
```

10.27. Menambahkan Komentar pada Tabel

Komentar atau COMMENT ditambahkan pada table untuk tujuan dokumentasi atau keterangan.

```
COMMENT ON TABLE employees
IS 'Employee Information';
Comment created.
```


10.28. Latihan

1. Buat table DEPARTMENT sehingga mempunyai struktur seperti ini :

Name	Null?	Type
ID		NUMBER(7)
NAME		VARCHAR2(25)

2. Dapatkan data dari table DEPT, masukkan ke dalam table DEPARTMENT. Periksa hasilnya

```
SQL> SELECT * FROM DEPARTMENT;
```

ID	NAME
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

2. Buat table EMPLOYEE sehingga mempunyai struktur seperti ini :

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

3. Modifikasi struktur dari table EMPLOYEE, sehingga strukturnya menjadi :

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

4. Buat table EMPLOYEE2 yang memiliki struktur dari table EMP hanya pada field empno,ename, dan deptno. Beri nama kolom yang baru ini pada table EMPLOYEE2 : ID, LAST_NAME, DEPT_ID.

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

ID	LAST_NAME	DEPT_ID
7839	KING	10
7698	BLAKE	30
7782	CLARK	10

.....

5. Drop table EMPLOYEE
6. Ubah nama table EMPLOYEE2 menjadi EMPLOYEE