

## PERTEMUAN 10

### PEMBUATAN TABEL

#### Tujuan Pembelajaran :

- 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

#### TEORI DAN PERCOBAAN

##### 10.1. 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  |

##### 10.2. Statement CREATE TABLE

Untuk membuat table dibuat perintah CREATE TABLE.

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

##### 10.3. Penggunaan 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, ...**

#### 10.4. Pembuatan Tabel dalam Database Oracle

Contoh pembuatan suatu table :

```
SQL> CREATE TABLE dept
```

```
2          (deptno NUMBER(2),  
3          dname VARCHAR2(14),  
4          loc VARCHAR2(13));
```

Untuk menampilkan struktur dari table :

```
SQL> DESCRIBE DEPT
```

| Name   | Null?    | Type         |
|--------|----------|--------------|
| DEPTNO | NOT NULL | NUMBER(2)    |
| DNAME  |          | VARCHAR2(14) |
| LOC    |          | VARCHAR2(13) |

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.

#### 10.5. Melakukan query pada 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.

Percobaan 1 : Untuk menampilkan table yang dimiliki oleh user :

```
SQL> SELECT *  
      2 FROM user_tables;
```

Percobaan 2 : Untuk menampilkan tipe object secara distinct yang dimiliki oleh user

```
SQL> SELECT DISTINCT object_type  
      2 FROM user_objects;
```

```
OBJECT_TYPE  
-----  
INDEX  
SEQUENCE  
TABLE  
VIEW
```

Percobaan 3 : Untuk menampilkan table, view, synonyms dan sequence yang dimiliki oleh user

```
SQL> SELECT *  
      2 FROM user_catalog;
```

| TABLE_NAME | TABLE_TYPE |
|------------|------------|
| ACCOUNT    | TABLE      |
| BONUS      | TABLE      |
| CUSTID     | SEQUENCE   |
| CUSTOMER   | TABLE      |
| DEPT       | TABLE      |
| .....      |            |

### 10.6. Pembuatan Tabel dengan Subquery

Table dapat dibuat dengan perintah CREATE TABLE dikombinasikan dengan pilihan *AS subquery*. Untuk mencocokkan kolom yang sesuai dengan kolom yang ada pada subquery.

Percobaan 4 : Buat table DEPT30, yang berisi detail semua pegawai yang bekerja di department 30. Baris datanya didapatkan dari table pegawai (EMP)

```
SQL> CREATE TABLE dept30  
      2 AS  
      3 SELECT empno, ename, sal*12 gaji_tahunan, hiredate  
      4 FROM EMP  
      5 WHERE deptno=30;
```

Table created.

Hasilnya adalah table dept30 yang berisi data pegawai yang bekerja di department 30 pada table pegawai.

```
SQL> select * from dept30;
```

| EMPNO | ENAME  | GAJI_TAHUNAN | HIREDATE  |
|-------|--------|--------------|-----------|
| 7698  | BLAKE  | 34200        | 01-MAY-81 |
| 7654  | MARTIN | 15000        | 28-SEP-81 |
| 7499  | ALLEN  | 19200        | 20-FEB-81 |
| 7844  | TURNER | 18000        | 08-SEP-81 |
| 7900  | JAMES  | 11400        | 03-DEC-81 |
| 7521  | WARD   | 15000        | 22-FEB-81 |

```
6 rows selected.
```

### 10.7. Statement ALTER TABLE

Perintah ALTER TABLE digunakan untuk :

- Menambah kolom baru
- Memodifikasi kolom yang sudah ada
- Mendefinisi nilai default untuk kolom baru

### 10.8. Menambahkan Kolom baru ke dalam Tabel

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

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

Percobaan 5 : Tambahkan kolom baru 'job' ke dalam table dept30. Kolom job mempunyai tipe data VARCHAR2(9).

```
SQL> ALTER TABLE dept30
2 ADD (job VARCHAR2(9));
```

```
Table altered.
```

### 10.9. Memodifikasi Kolom pada Tabel

Sintak dari perintah ALTER TABLE untuk memodifikasi kolom pada Tabel :

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

Percobaan 6 : Modifikasi tipe data kolom, dan ukuran dari kolom *ename* pada table *dept30*, menjadi VARCHAR2(15)

```
SQL> ALTER TABLE dept30  
2 MODIFY (ename VARCHAR2(15));  
  
Table altered.
```

#### 10.10. 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
- Perintah drop ini tidak bisa di-rollback

Percobaan 7 : Hapus table *dept30* dengan perintah DROP

```
SQL> DROP TABLE dept30;  
  
Table dropped.
```

#### 10.11. Merubah Nama dari 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.

Percobaan 7 : Ubah nama table DEPT menjadi DEPARTMENT

```
SQL> RENAME dept TO department;  
  
Table renamed.
```

### 10.12. Statement TRUNCATE TABLE

Melakukan penghapusan table dengan perintah DROP, 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.

### 10.13. Menambahkan Komentar (COMMENT) pada Tabel

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

Percobaan 8 : Tambahkan komentar ke table *EMP* dengan keterangan 'Informasi Pegawai'

```
SQL> COMMENT ON TABLE EMP  
2 IS 'Informasi Pegawai';  
  
Comment created.
```

Komentar dapat ditampilkan melalui data dictionary view :

- ALL\_COL\_COMMENTS
- USER\_COL\_COMMENTS
- ALL\_TAB\_COMMENTS
- USER\_TAB\_COMMENTS

### LATIHAN SOAL

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 menjadi seperti ini :

```
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      |

.....

- Drop table EMPLOYEE
- Ubah nama table EMPLOYEE2 menjadi EMPLOYEE
- Tambahkan perintah ke table DEPARTMENT dan EMPLOYEE yang mendeskripsikan isi dari kedua table.