



CHAPTER

16

**SQL Operasi**  
**DDL**

*Arif Basofi*

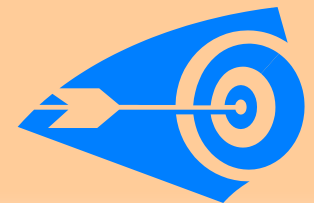
*PENS 2015*

# Objectives

## Tujuan:

Mengenal operasi perintah SQL dalam:

- DDL (Data Definition Language)



# SQL – DDL

## Data Definition Language (DDL)

- Dengan perintah SQL dalam DDL, dapat digunakan dalam:
  - Membuat (CREATE) dan menghilangkan (DROP) table
  - Memodifikasi (ALTER) tabel dalam database
  - Membuat (CREATE) dan menghilangkan (DROP) view
  - Membuat (CREATE) dan menghilangkan (DROP) indeks
- Hasil dari kompilasi perintah DDL berupa **kumpulan tabel** yang disimpan dalam file khusus: **Kamus Data (Data Dictionary)**.
- **Data Dictionary**: merupakan **metadata (superdata)**, yaitu data yang mendeskripsikan data sesungguhnya. Data dictionary ini akan selalu diakses dalam suatu operasi database sebelum suatu file data yang sesungguhnya diakses.

## S Q L – DDL (Create Table)

### Create Table (Membuat Table)

- Untuk membuat skema table baru sekaligus mendefinisikan relasinya, dapat digunakan perintah SQL berikut:

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

- Yang harus dispesifikasikan:
  - Nama Table
  - Nama Kolom, Kolom tipe data dan ukuran kolom
- Aturan nama table dan kolom:
  - Harus diawali dengan huruf
  - Panjang nama / karakter antara 1–30 characters
  - Mengandung A–Z, a–z, 0–9, \_, \$, dan #
  - Tidak boleh terdapat nama yang sama (duplikat) dengan object database lain.
  - Tidak boleh menggunakan keyword SQL database

# S Q L – DDL (Create Table)

## Creating New Table

- **Create the table.**

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

**Table created.**

- **Confirm table creation.**

```
DESCRIBE dept
```

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

# S Q L – DDL (Create Table)

## Constrain Primary Key (tanpa nama)

### Constraint Primary Key (tanpa nama constraint)

- Dapat diberi nama juga tidak.

```
CREATE TABLE dept
(deptno      NUMBER(2) ,
  dname      VARCHAR2(14) ,
  loc        VARCHAR2(13) ,
PRIMARY KEY (deptno)) ;
```

**Table created.**

```
CREATE TABLE dept
(deptno      NUMBER(2) PRIMARY KEY ,
  dname      VARCHAR2(14) ,
  loc        VARCHAR2(13)) ;
```

**Table created.**

# S Q L – DDL (Create Table)

## Constrain Primary Key dengan Nama

### Constraint Primary Key (dengan nama constraint)

- Tujuan: jika suatu saat terjadi adanya perubahan constraint.

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

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

# S Q L – DDL (Create Table)

## Constrain Primary Key & Foreign Key

- **Constraint Primary Key dan Foreign Key**

```
CREATE TABLE dept
(deptno      NUMBER(2),
dname       VARCHAR2(14),
loc_id      VARCHAR2(13),
CONSTRAINT PK_deptno PRIMARY KEY (deptno),
CONSTRAINT FK_loc_id FOREIGN KEY (loc_id)
REFERENCES location(loc_id));
```

**Table created.**

```
CREATE TABLE dept
(deptno      NUMBER(2)
CONSTRAINT PK_deptno PRIMARY KEY,
dname       VARCHAR2(14),
loc_id      VARCHAR2(13)
CONSTRAINT FK_loc_id FOREIGN KEY
REFERENCES location(loc_id));
```

**Table created.**



## S Q L – DDL (Create Table)

### Create Table Menggunakan Subquery

- Statement SQL untuk create table baru sekaligus meng-insert data sehingga sama persis dari table lain, dapat dilakukan dengan menggunakan **subquery** perintah SQL berikut:

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

- Jumlah kolom yang didefinisikan harus sesuai dengan kolom subquery yang dibuat, termasuk tipe datanya.

# S Q L – DDL (Create Table)

## Create Table 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

## S Q L – DDL (Create Table)

### Create Table Kosong Menggunakan Subquery

Untuk membuat table baru dari table lain, tapi **tanpa** termasuk **datanya**, dapat dilakukan dengan statement SQL berikut:

```
CREATE TABLE COPY_TABLE AS
(SELECT *
FROM employees
WHERE 1 = 2);
```

Table created. **False**

# SQL – DDL (Alter Table)

## Alter Table

Gunakan **ALTER TABLE** untuk:

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

## SQL – DDL (Alter Table)

Gunakan statement **ALTER TABLE** untuk **add**, **modify**, atau **drop** columns.

```
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);
```

# SQL – DDL (Alter Table)

## Adding a Column

New column

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

JOB_ID

“Add a new column to the DEPT80 table.”

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	

# SQL – DDL (Alter Table)

## Adding a Column

- Gunakan klausa **ADD** untuk menambahkan kolom.

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

- Kolom baru akan berada pada akhir kolom.

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	

# SQL – DDL (Alter Table)

## Modifying a Column

- Anda dapat merubah tipe data, ukuran dan nilai default kolom.

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



## SQL – DDL (Alter Table)

### Dropping a Column

Gunakan klausa **DROP COLUMN** untuk menghapus kolom.

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

Pastikan bahwa anda benar-benar akan menghapus kolom yang dimaksud, karena tidak bisa dilakukan **ROLLBACK** kembali!!

# SQL – DDL (Drop Table)

## Dropping a Table

- Semua data dan struktur dalam table akan dihapus (**delete**).
- Beberapa transaksi dengan status pending akan di commit (dimasukkan).
- Semua indexes di dropped.
- Anda tidak bisa melakukan **ROLLBACK** dari statement **DROP TABLE** (beda dengan operasi DML) !!

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

# SQL – DDL (Drop Table)

## Rename Objects database

- Untuk merubah nama table, view, sequence, atau synonym, anda dapat execute dengan statement **RENAME**.

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

- Biasanya yang melakukan harus user yang pemilik (**owner**) dari object database yang dibuat.

# S Q L – DDL Excercises

## Latihan:

1. Dengan SQL Query, buat table baru “buku” dengan struktur sebagai berikut:

```
kode_buku      varchar2 (3) ;  
judul_buku     varchar2 (20) ;  
pengarang      varchar2 (10) ;  
penerbit       varchar2 (20) ;  
th_terbit      number (4) ;
```

2. Ubah struktur table diatas untuk ukuran (size) kolom pengarang menjadi 15.
3. Tambahkan kolom baru:

```
jumlah_buku    number (2) ;  
harga_buku     number (7, 2) ;
```

4. Hapus kolom harga\_buku.

## SQL – DDL Exercises

5. Dengan SQL Query, buat struktur table baru “deptku” beserta datanya yang sama persis dengan table departments, dengan struktur kolom table deptku sebagai berikut:

```
no_dept      number(4);  
nama_dept    varchar2(30);  
id_manajer   number(6);  
id_lokasi    number(4);
```

6. Dengan SQL Query, buat struktur table baru kosongan “deptku\_kosong” yang strukturnya sama persis dengan table departments. Struktur kolom sama persis pada no.5.
7. Hapus table deptku dan deptku\_kosong.

# SQL – DDL Exercises

