

LATIHAN MODUL 2
LAPORAN PRAKTIKUM BASIS
DATA Data Definition Language (DDL)



Disusun oleh:

NAMA: ANTONIUS MUNTHE

NIM: 121140032

KELAS: BASIS DATA RA

PROGRAM STUDI TEKNIK
INFORMATIKA INSTITUT TEKNOLOGI
SUMATERA LAMPUNG SELATAN

2022

BAB 1

TEORI DASAR

A. Data Definition Language (DDL)

DDL (Data Definition Language) adalah kumpulan perintah SQL yang dapat anda gunakan untuk mengelola, mengubah struktur datatype dari objek pada database seperti index, table, trigger, view dan lain sebagainya. MySQL itu sendiri adalah DBMS atau Database management system yang menggunakan bahasa SQL sebagai penghubung antara software dengan database server. DDL digunakan untuk membuat skema, tabel, indeks, dan lain sebagainya.

B. Perintah Dasar DDL

1. Create

- Create database => Untuk membuat database yang baru
- Create Table => Untuk membuat tabel yang baru
- Create Procedure => Untuk membuat prosedur yang baru
- Create index => Untuk membuat index yang baru
- Create function => Untuk membuat fungsi yang baru
- Create trigger => Untuk membuat reaksi tertentu atau trigger pada data base

2. Alter

Alter merupakan perintah dalam DDL yang digunakan untuk mengubah struktur pada tabel yaitu Alter. Perintah ini digunakan untuk memodifikasi bentuk kolom, menambah dan juga mengganti tabel yang sudah ada

- Menambah Kolom Tabel
Berikut perintah Alter yang digunakan untuk menambah kolom table.
`ALTER TABLE nama_table ADD nama_field tipe_data`
- Modifikasi Kolom Tabel
Berikut perintah Alter yang digunakan untuk memodifikasi kolom table. `ALTER TABLE nama_table MODIFY nama_field tipe_data`
- Menghapus Kolom Tabel
Berikut perintah alter untuk menghapus kolom table.
`ALTER TABLE nama_table DROP nama_field`

3. Drop

Drop merupakan perintah dalam DDL yang digunakan untuk membuat perintah menghapus objek dalam database.

- Menghapus Database
Berikut perintah pada drop yang digunakan untuk menghapus database. `DROP DATABASE nama_database`
- Menghapus tabel
Berikut perintah pada drop yang digunakan untuk menghapus tabel dalam database. `DROP TABLE nama_table`

BAB 2

PEMBAHASAN DAN ANALISIS

1. Membuat Database dengan nama “ Northwind”

Sintaks yang digunakan untuk membuat database yang baru adalah “CREATE DATABASE nama_database”. Sehingga, untuk membuat database yang baru sesuai dengan nama database di atas ialah dengan cara mengetikkan perintah “CREATE DATABASE Northwind” pada command prompt atau cmd. Perintah dan hasil tersebut dapat dilihat pada gambar sebagai berikut. Kemudian, ketikkan perintah SHOW DATABASES; untuk melihat list database yang sudah dibuat.

```
MariaDB [(none)]> create database northwind
-> ;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| bank_mino |
| information_schema |
| mysql |
| northwind |
| performance_schema |
| phpmyadmin |
| test |
| universitas |
+-----+
8 rows in set (0.001 sec)
```

2. Membuat table produk, suppliers, pegawai dan pembeli

Membuat tabel dengan nama produk, suppliers, pegawai dan pembeli pada database Northwind. Pada Tabel tersebut berisi atribut masing-masing produk id, suppliers id, id pegawai dan id pembeli dan sebagai *primary key*. Masing-masing atribut tersebut juga terdapat tipe datanya. Untuk memilih database yang ingin dimanipulasi ialah dengan cara mengetikkan perintah “USE Northwind”. Terdapat output “Database changed” yang menyatakan bahwa database tersebut telah berubah atau sedang diakses. Selanjutnya, ketikkan perintah “CREATE table produk, suppliers, pegawai dan pembeli” sesuai sintaks berikut yang diikuti dengan atribut serta tipe data dan juga *primary key* nya. Hal itu dapat dilihat pada gambar sebagai berikut.

```

MariaDB [northwind]> create table produk(
-> produk_id varchar(10),
-> produk_nama char(20),
-> primary key (produk_id));
Query OK, 0 rows affected (0.030 sec)

MariaDB [northwind]> create table suppliers(
-> suppliers_id varchar(10),
-> company_name char(20),
-> nama_kontak char(20),
-> primary key (suppliers_id));
Query OK, 0 rows affected (0.019 sec)

MariaDB [northwind]> create table pegawai(
-> id_pegawai varchar(10),
-> nama_pegawai char(10),
-> jabatan char(10),
-> primary key (id_pegawai));
Query OK, 0 rows affected (0.019 sec)

MariaDB [northwind]> create table pembeli(
-> id_pembeli varchar(10),
-> nama_pembeli char(10),
-> kontak_pembeli int (13),
-> primary key (id_pembeli));
Query OK, 0 rows affected (0.014 sec)

MariaDB [northwind]> show tables;
+-----+
| Tables_in_northwind |
+-----+
| pegawai              |
| pembeli              |
| produk               |
| suppliers             |
+-----+
4 rows in set (0.007 sec)

```

3. Mengubah Struktur Tabel

Berikut cara dan sintaks aturan perubahan struktur pada table rekening

a. ganti kolom “nama_pegawai” menjadi “pegawai_nama”

Pada tabel pegawai, ubah atribut “nama_pegawai” menjadi “pegawai_nama” yang dimana atribut tersebut bertipe char. Sintaks yang digunakan adalah ALTER TABLE table_name CHANGE old_name_column new_name_new_column new-data_type. Hal itu dapat dilihat pada gambar sebagai berikut.

```

MariaDB [northwind]> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pegawai | varchar(10) | NO   | PRI | NULL    |       |
| nama_pegawai | char(10)   | YES  |     | NULL    |       |
| jabatan    | char(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.025 sec)

MariaDB [northwind]> alter table pegawai change nama_pegawai pegawai_nama char(10);
Query OK, 0 rows affected (0.031 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pegawai | varchar(10) | NO   | PRI | NULL    |       |
| pegawai_nama | char(10)   | YES  |     | NULL    |       |
| jabatan    | char(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.027 sec)

```

b. ganti kolom “nama_pembeli” menjadi “pembeli_nama”

Pada tabel pembeli, ubah atribut “nama_pembeli” menjadi “pembeli_nama” yang dimana atribut tersebut bertipe char. Sintaks yang digunakan adalah ALTER TABLE table_name CHANGE old_name_column new_name_new_column new-data_type. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [northwind]> desc pembeli;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pembeli | varchar(10) | NO | PRI | NULL | |
| nama_pembeli | char(10) | YES | | NULL | |
| kontak_pembeli | int(13) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.025 sec)

MariaDB [northwind]> alter table pembeli change nama_pembeli pembeli_nama char(10);
Query OK, 0 rows affected (0.018 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> alter table pembeli change kontak_pembeli pembeli_kontak int(13);
Query OK, 0 rows affected (0.028 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> desc pembeli;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pembeli | varchar(10) | NO | PRI | NULL | |
| pembeli_nama | char(10) | YES | | NULL | |
| pembeli_kontak | int(13) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.026 sec)
```

c. Tambahkan kolom “jumlah_stock” setelah produk_nama

Pada table produk, tambahkan atribut “jumlah_stock” yang dimana atribut tersebut bertipe int. Kolom atribut tersebut diletakkan setelah kolom kode cabang. Sintaks yang digunakan adalah “ALTER TABLE nama_table ADD new_column new-data_type after name_column;”. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [northwind]> alter table produk add jumlah_stock int(5) after produk_nama;
Query OK, 0 rows affected (0.026 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> desc produk;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| produk_id | varchar(10) | NO | PRI | NULL | |
| produk_nama | char(20) | YES | | NULL | |
| jumlah_stock | int(5) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.019 sec)
```

4. Tambahkan kolom “suppliers_id” ke dalam tabel produk. Kemudian relasikan kolom “suppliers_id” di tabel Produk dengan kolom “suppliers_id” di tabel Suppliers .
Untuk cara merelasikan kolom tersebut dapat dilihat pada gambar:.

```
MariaDB [northwind]> desc produk;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| produk_id  | varchar(10) | NO   | PRI | NULL    |       |
| produk_nama | char(20)   | YES  |     | NULL    |       |
| jumlah_stock | int(5)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.019 sec)

MariaDB [northwind]> alter table produk add suppliers_id varchar(10) after produk_id;
Query OK, 0 rows affected (0.029 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> desc produk;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| produk_id  | varchar(10) | NO   | PRI | NULL    |       |
| suppliers_id | varchar(10) | YES  |     | NULL    |       |
| produk_nama | char(20)   | YES  |     | NULL    |       |
| jumlah_stock | int(5)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.026 sec)

MariaDB [northwind]> alter table produk add foreign key (suppliers_id)
-> references suppliers (suppliers_id)
-> on delete restrict
-> on update cascade;
Query OK, 0 rows affected (0.107 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [northwind]> desc produk;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| produk_id  | varchar(10) | NO   | PRI | NULL    |       |
| suppliers_id | varchar(10) | YES  | MUL | NULL    |       |
| produk_nama | char(20)   | YES  |     | NULL    |       |
| jumlah_stock | int(5)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.026 sec)
```

Perintah ON UPDATE RESTRICT itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka hal itu tidak akan diperbolehkan jika pada suatu tabel lainnya masih terdapat relasi datanya.

Perintah ON UPDATE CASCADE itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka akan secara otomatis akan dapat menghapus data yang sesuai dalam tabel lainnya.

BAB III

KESIMPULAN

1. Terdapat penerapan dan cara-cara untuk membuat database yang baru pada MySQL. Sintaks yang digunakan ialah `CREATE DATABASE nama_database`.
2. Terdapat penerapan dan cara-cara untuk membuat tabel beserta atribut dan tipe datanya masing-masing. Sintaks yang digunakan untuk membuat tabel ialah `CREATE TABLE nama_table` kemudian didalamnya terdapat database yang berisi atribut dan tipe datanya serta atribut yang dijadikan sebagai *primary key*.
3. Terdapat penerapan dan cara-cara untuk mengubah struktur tabel pada database. Sintaks yang digunakan ialah `ALTER TABLE table_name CHANGE old_name_column new_name_new_column new-data_type`.
4. Terdapat penerapan dan cara-cara untuk menambahkan kolom struktur tabel pada database. Sintaks yang digunakan ialah `ALTER TABLE nama_table add new_column new data_type`.
5. Terdapat penerapan dan cara-cara untuk menghubungkan suatu tabel dengan tabel lainnya dengan melalui *foreign key*. Sintaks yang digunakan ialah `"ALTER TABLE table_name ADD FOREIGN KEY(column_name_of_foreign_key) kemudian diikuti dengan REFERENCES table_name(column_name_as_references) ON UPDATE RESTRICT, ON UPDATE CASCADE; "`.
6. Perintah `ON UPDATE RESTRICT` itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka hal itu tidak akan diperbolehkan jika pada suatu tabel lainnya masih terdapat relasi datanya.
7. Perintah `ON UPDATE CASCADE` itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka akan secara otomatis akan dapat menghapus data yang sesuai dalam tabel lainnya.

PUSTAKA

<http://teknik-informatika-s1.stekom.ac.id/informasi/baca/Pengertian-DDL-dan-DML-Beserta-Contoh-Perintahnya-dalam-Database/764d4cc8477ab0f54654ba0e80a71e687a3f987c#:~:text=Untuk%20pengertian%20DDL%20dan%20DML,trigger%2C%20view%20dan%20lain%20sebagainya>

Modul Praktikum Basis Data Pertemuan 2