# DDL (DATA DEFINITON LANGUAGE)

1. **Pengertian DDL (Data Definition Language)**

DDL adalah singkatan dari Data Definition Language yaitu kumpulan perintah pada SQL untuk menggambarkan desain dari database secara menyeluruh, selain itu DDL (Data Definition Language) juga digunakan untuk membuat, merubah maupun menghapus struktur atau definisi tipe data dari obyek yang ada pada database.

1. **Fungsi DDL (Data Definition Language)**

DDL (Data Definition Language) berfungsi untuk membuat, merubah dan menghapus struktur data pada database.

1. **Type Data pada MySQL**

Tipe data adalah suatu bentuk pemodelan data yang dideklarasikan pada saat melakukan pembuatan tabel. Tipe data ini akan mempengaruhi setiap data yang akan dimasukkan ke dalam sebuah tabel. Data yang akan dimasukkan harus sesuai dengan tipe data yang dideklarasikan.

Berbagai type data pada MySQL dapat dilihat pada tabel berikut :

|  |  |
| --- | --- |
| **Type Data** | **Keterangan** |
| TINYINT | Ukuran 1 byte. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda: -128 sampai dengan 127 dan untuk yang tidak bertanda : 0 s/d 255. Bilangan tak bertandai dengan kata UNSIGNED |
| SMALLINT | Ukuran 2 Byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -32768 s/d 32767 dan untuk yang tidak bertanda : 0 s/d 65535 |
| MEDIUMINT | Ukuran 3 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -8388608 s/d 8388607 dan untuk yang tidak bertanda : 0 s/d 16777215 |
| INT | Ukuran 4 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -2147483648 s/d 2147483647 dan untuk yang tidak bertanda : 0 s/d 4294967295 |
| INTEGER | Ukuran 4 byte. Sinonim dari int |
| BIGINT | Ukuran 8 byte. Bilangan bulat terbesar dengan jangkauan untuk bilangan bertanda : -9223372036854775808 s/d 9223372036854775807 dan untuk yang tidak bertanda : 0 s/d 1844674473709551615 |
| FLOAT | Ukuran 4 byte. Bilangan pecahan |
| DOUBLE | Ukuran 8 byte. Bilangan pecahan |
| DOUBLEPRECISION | Ukuran 8 byte. Bilangan pecahan |
| REAL | Ukuran 8 byte. Sinonim dari DOUBLE |
| DECIMAL (M,D) | Ukuran M byte. Bilangan pecahan, misalnya DECIMAL(5,2 dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99 |
| NUMERIC (M,D) | Ukuran M byte. Sinonim dari DECIMAL, misalnya NUMERIC(5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99 |
|  | ***Type Data untuk Bilangan (Number)*** |

|  |  |
| --- | --- |
| **Type Data** | **Keterangan** |
| DATETIME | Ukuran 8 byte. Kombinasi tanggal dan jam, dengan jangkauan dari ‘1000-01-01 00:00:00’ s/d ‘9999-12-31 23:59:59’ |
| DATE | Ukuran 3 Byte. Tanggal dengan jangkauan dari ‘1000-01-01’ s/d ‘9999-12-31’ |
| TIMESTAMP | Ukuran 4 byte. Kombinasi tanggal dan jam, dengan jangkauan dari ‘1970-01-01 00:00:00’ s/d ‘2037’ |
| TIME | Ukuran 3 byte. Waktu dengan jangkauan dari ‘839:59:59’ s/d ‘838:59:59’ |
| YEAR | Ukuran 1 byte. Data tahun antara 1901 s/d 2155 |
|  | ***Type Data untuk Tanggal dan Jam*** |

******

|  |  |
| --- | --- |
| **Type Data** | **Keterangan** |
| CHAR | Mampu menangani data hingga 255 karakter. Tipe data CHAR mengharuskan untuk memasukkan data yang telah ditentukan oleh kita. |
| VARCHAR | Mampu menangani data hingga 255 karakter. Tipe data VARCHAR tidak mengharuskan untuk memasukkan data yang telah ditentukan oleh kita. |
| TINYBLOB, TINYTEXT | Ukuran 255 byte. Mampu menangani data sampai 2^8-1 data. |
| BLOB, TEXT | Ukuran 65535 byte. Type string yang mampu menangani data hingga 2^16-1 (16M-1) data. |
| MEDIUMBLOB, MEDIUMTEXT | Ukuran 16777215 byte. Mampu menyimpan data hingga 2^24-1 (16M-1) data. |
| LONGBLOB, LONGTEXT | Ukuran 4294967295 byte. Mampu menyimpan data hingga berukuran GIGA BYTE. Tipe data ini memiliki batas penyimpanan hingga 2^32-1 (4G-1) data. |
| ENUM(‘nilai1’,’nilai2’,…,’nilaiN’) | Ukuran 1 atau 2 byte. Tergantung jumlah nilai enumerasinya (maksimum 65535 nilai) |
| SET(‘nilai1’,’nilai2’,…,’nilaiN’) | 1,2,3,4 atau 8 byte, tergantung jumlah anggota himpunan (maksimum 64 anggota) |
|  | ***Type Data untuk Karakter dan Lain-lain*** |

1. **Menciptakan Database**

Database adalah sebuah media utama yang harus dibuat dalam membangun sebuah basis data agar nantinya dapat kita letakkan beberapa tabel dengan field-fieldnya.

Perintah yang digunakan untuk menciptakan database pada MySQL dengan Syntax berikut :

**CREATE DATABASE** nama\_database;

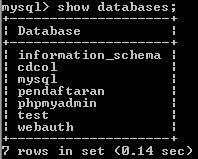
Contoh :

mysql> create database pendaftaran;

Query OK, 1 row affected (0.11 sec)

Pada contoh diatas, query OK menyatakan bahwa pembuatan database dengan nama pendaftaran berhasil dibuat, untuk melihat database yang ada pada MySQl dapat menggunakan SIntax berikut ;

**SHOW DATABASES;**

Contoh :

Latihan 1:

Buatlah 2 database baru dengan nama sekolah dan industri dan tampilkan semua database yang ada. Screenshot hasil pekerjaan.

1. **Menghapus Database**

Untuk menghapus Database yang telah dibuat dapat menggunakan query SQL berikut :

**DROP DATABASE** nama\_database;

Drop berarti menghapus. Query SQL ini berfungsi untuk menghapus sebuah database, seperti contoh berikut :

mysql> drop database pendaftaran;

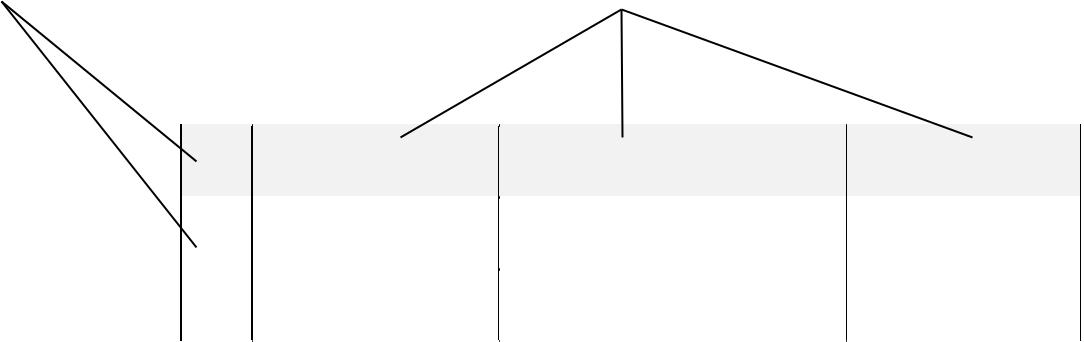
Query OK, 0 rows affected (0.02 sec)

Latihan 2:

Hapus 1 database yang sudah dibuat yaitu industri. Screenshot hasil pekerjaan.

1. **Menciptakan Tabel**

Tabel adalah obyek utama yang harus ada pada sebuah basis data karena di dalamnya semua data akan disimpan. Tabel terletak pada sebuah database, sehingga pembuatan tabel dilakukan setelah sebuah database telah dibuat. Dalam tabel terdapat bari dan kolom. Baris diistilahkan dengan recordset dan kolom dengan field.

**Recordset** **Field**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Nama | Alamat | Phone |
|  |  |  |  |
| 1 | Boy Trimoyo | Jl. Ujung berung | 08156849511 |
|  |  |  |  |
| 2 | Irfan Nurhudin | Kp. Panyileukan Cibiru | 08122295434 |
|  |  |  |  |

Untuk membuat sebuah tabel atau lebih, database harus diaktifkan dulu karena tabel akan dimasukkan ke dalam database yang akan diaktifkan. Sintax untuk mengaktifkan Database adalah :

**USE** nama\_database;

Contoh :

mysql> use pendaftaran;

Database changed

Setelah masuk ke dalam database anda dapat membuat sebuah tabel atau lebih. Untuk membuat tabel dapat menggunakan sintax dibawah ini :

**CREATE TABLE** nama\_tabel ( field-1 type(length), field-2 type(length), field-3 type(length), …… ….(….));

Contoh :

mysql> create table data\_diri (

-> no int(3),

-> nama varchar(35),

-> alamat varchar(60),

-> email varchar(40),

-> no\_telepon varchar(15),

-> sex char(1));

Query OK, 0 rows affected (0.08 sec)

Pada contoh diatas, query OK menyatakan bahwa pembuatan tabel dengan nama data\_diri berhasil dibuat, untuk melihat tabel yang ada pada database dapat menggunakan SIntax berikut ;

**SHOW TABLES;**

Contoh :



Latihan 3:

* Buatlah 3 table dengan nama :

1. Data\_siswa dengan atribut nis type data integer primary key, nama type data varchar, alamat type data text, gender typedata enum(laki-laki, perempuan)
2. Kelas dengan atribut kode\_kelas type data varchar primary key, nama\_kelas typedata varchar
3. Mapel dengan atribut kkode\_mapel typedata integer primary key dan auto\_increment, nama\_mapel type data varchar, jenis\_mapel type data varchar.

* Tampilkan semua table yang sudah dibuat.
* Screenshot hasil pekerjaan.

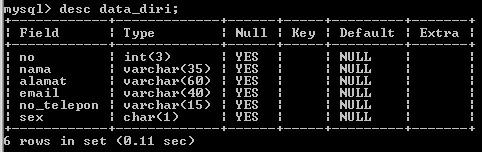
1. **Melihat Struktur Tabel**

Setelah tabel dibuat, anda dapat melihat tipe data dan panjang recordset dengan cara menampilkan struktur tabel. Perintah yang digunakan untuk menampilkan struktur tabel adalah :

**DESC** nama\_tabel;

Atau

**DESCRIBE** nama\_tabel;

Contoh :

Latihan 4:

Tampilkan struktur table data\_siswa, kelas dan mapel. Screenshot hasil pekerjaan.

1. **Menghapus Tabel**

Untuk menghapus Tabel yang telah dibuat dapat menggunakan query SQL berikut :

**DROP TABLE** nama\_tabel;

Drop berarti menghapus. Query SQL ini berfungsi untuk menghapus sebuah Tabel, seperti contoh berikut :

mysql> drop table data\_diri;

Query OK, 0 rows affected (0.03 sec)

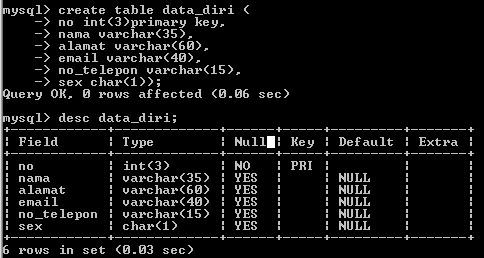
Latihan 5:

Hapuslah table mapel. Screenshot hasil pekerjaan.

1. **Membuat Kunci Primer ( Primary Key )**

Dalam membuat sebuah database, kita akan menemukan sebuah record yang data nya tidak boleh sama dengan record yang lain. Agar data tidak kembar maka harus membuat sebuah kolom yang di deklarasikan sebagai kunci primer (primary key), Primary key hanya diperboleh kan dibuat satu kunci. Syntax untuk menciptakan kunci primer (primary key) adalah :

**CREATE TABLE** nama\_tabel ( field-1 type(length)**PRIMARY KEY**, field-2 type(length), …… ….(….));

Contoh :

Latihan 6:

Buatlah tabel data\_guru dengan atribut nip typedata varchar primary key, nama\_guru typedata varchar, alamat typedata text,tanggal\_lahir typedata date

1. **Membuat Kolom Unik ( Unique )**

Kolom Unique adalah sebuah bentuk kolom yang tidak mengizinkan adanya data kembar. Apabila pada proses input terdapat data kembar maka proses tersebut akan digagalkan atau ditolak oleh database.

Syntax untuk menciptakan Kolom unik (Unique) adalah :

**CREATE TABLE** nama\_tabel ( field-1 type(length), field-2 type(length), …… ….(….),**UNIQUE** (field-1,field-2));

Contoh :

mysql> Create table pribadi (

-> kd\_pribadi CHAR(3),

-> panggilan char(4),

-> nama varchar(35),

-> email varchar(50),

-> sex char(1),

-> UNIQUE (kd\_pribadi,panggilan));

Query OK, 0 rows affected (0.08 sec)

Latihan 7:

Buatlah tabel karyawan dengan atribut kode\_karyawan typedata integer primary key auto\_increment, nama\_karyawan varchar, telp varchar unique. Screenshot hasil pekerjaan.

1. **Manipulasi Tabel**

Perubahan tabel yang telah dibuat akan selalu dilakukan mengingat perkembangan database, termasuk diantaranya menambahkan beberapa field pada tabel, mengganti nama field maupun tabel.

* ***Mengganti nama tabel***

Query SQL untuk merubah nama tabel dengan menggunakan **RENAME**, Sintax seperti berikut :

**RENAME TABLE** tabel\_lama **TO** tabel\_baru;

Contoh :

mysql> rename table pribadi to data\_pribadi;

Query OK, 0 rows affected (0.02 sec)

Latihan 8:

Gantilah nama table karyawan dengan data\_karyawan. Creenshot hasil pekerjaan.

* ***Menambah Field pada Tabel***

Menambah kolom dapat diartikan sebagai langkah untuk menyisipkan field baru pada sebuah tabel. Untuk melakukan penambahan Field maka **ALTER** spesifikasi yang digunakan adalah **ADD**. Sintax yang digunakan adalah :

**ALTER TABLE** nama\_tabel **ADD** nama\_field Type\_data(length);

Contoh :

mysql> alter table data\_diri add gol\_darah char(1);

Query OK, 0 rows affected (0.14 sec)

Records: 0 Duplicates: 0 Warnings: 0

Latihan 9:

Tambahkan kolom pada table data\_karyawan dengan nama kolom gender type data enum(laki-laki,perempuan). Sreenshot hasil pekerjaan.

* ***Menghapus Field pada Tabel***

Pada pembuatan database pasti terdapat kesalahan seperti pada field tabel yang berlebihan dan lain-lain. Untuk melakukan Penghapusan Field maka ALTER spesifikasi yang digunakan adalah **DROP**. Sintax yang digunakan adalah :

**ALTER TABLE** nama\_tabel **DROP** nama\_field;

Contoh :

mysql> alter table data\_diri drop gol\_darah;

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

Latihan 10:

Hapus kolom gender pada tabel data\_karyawan. Screenshot hasil pekerjaan.

* ***Mengganti nama Field pada Tabel***

Pada pembuatan database pasti terdapat kesalahan seperti pada field tabel. Untuk melakukan penggantian Field maka ALTER spesifikasi yang digunakan adalah **Change**. Sintax yang digunakan adalah :

**ALTER TABLE** nama\_tabel **change** nama\_field\_lama nama\_field\_baru type\_data;

Contoh :

mysql> alter table data\_diri change nama nama\_lengkap varchar(100);

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

Latihan 11:

Gantilah nama kolom gender beserta typedatanya menjadi jenis\_kelamin typedata varchar pada tabel data\_karyawan. Screenshot hasil pekerjaan.