Data Definition Languange

Database

Database merupakan kumpulan dari objek-objek yang

digunakan untuk menyimpan dan memanipulasi data seperti

tabel, view database, trigger, dan prosedur tersimpan.

Suatu standar perintah (command) yang menentukan perbedaan struktur dalam suatu database.

Statement DDL membuat, memodifikasi dan menghapuskan objek database seperti table, indeks, dan user

Contoh: Create, Alter, Drop

Create Database, sebelum melakukan hla lain, kita perlu membuat database terlebih dahulu. Suatu database merupaakan tempat data.

1. Pertama, log in ke MySQL Serverr menggunakan user “root”

mysql -u root -p

Jika user root tidak memiliki password, maka dapat dituliskan:

mysql -u root

1. Membbuat suatu database

CREATE DATABASE [IF NOT EXIST] database\_name;

Statement CREATE DATABASE DIIKUTI dengan database\_name, yaitu nama database yang ingin kita buat

IF NOT EXISTS adalah klausa opsional dari statement tsb.

NOTE: Klause IF NOT EXISTS mencegah kita untuk membuat database yang sudah ada di server database.

1. Contoh SQL untuk membuat database baru

CREATE DATABASE classicmodels;

1. Menampilkan daftar/list database

Statement SHOW DATABASES menampilkan list database di MySQL database server.

Statement SHOW DATABASES dapat digunakan untuk memeriksa database yang telah dibuat atau untuk melihat semua database yang ada di database server sebelum mebuat database baru.

1. Menggunakan Suatu Database

USE database\_name;

Perintah untuk memillih database classicmodels menggunakan statement USE dapat dilhat dibawah ini

USE classicmodels;

Alter DATABASE

Statement ALTER DATABASE digunakan untuk mengubah, memelihara, atau memulihkan database yang sudah ada

ALTER (DATABASE | SCHEMA) [db\_name] alter\_specification …

Alter\_specification:

[DEFAULT] CHARACTER SET [=] charset\_name

| [DEFAULT]

Drop Database

Menghapus database berarti menghapus semua table yang terdapat dalam database dan database itu sendiri secara permanen.

Sangat penting untuk menjalankan query ini dengan ekstra hati-hati.

1. Menghapus Database

DROP DATABASE [IF EXISTS] database\_name;

Klausa DROP DATABASE diikuti dengan database\_name yang hendak dihapus

IF EXISTS adalah bagian opsional dari statement tersebut.

NOTE: Klausa IF EXISTS untuk mencegah penghapusan database yang tidak ada di server database.

Untuk mengimplementasikan statement DROP DTATABASE, dilakukan dengan membuat database baru,memastikan database sudah dibuat.

Table

Table adalah objek database yang membuat semua data (informasi) dalam database.

Column\_list

column\_list menspesifikasikan daftar kolom untuk suatu tabel. Untuk mendefinisikan suatu kolom di column\_list gunakan sintaks berikut

column\_name data\_type(length) [NOT NULL] [DEFAULT value] [AUTO\_INCREMENT]

column\_name menentukan nama kolom.

Setiap kolom memiliki tipe data dan panjang maksimum

Contoh: VARCHAR(255)

NOT NULL menunjukkan bahwa kolom tersebut tidak dapat bernilai NULL.

Nilai DEFAULT digunakan untuk menentukan nilai default dari suatu kolom.

AUTO\_INCREMENT menunjukkan bahwa nilai kolom dihasilkan secara otomatis setiap baris baru dimasukkan ke dalam tabel

Setiap tabel memiliki satu dan hanya satu kolom AUTO\_INCREMENT.

**Create Table**

Untuk membuat tabel baru dalam database, gunakan statement CREATE TABLE

CREATE TABLE [IF NOT EXISTS] table\_name(

Column\_list

) ENGINE=storage\_engine

Pertama, spesifikasikan table\_name setelah klausa CREATE TABLE.

Note: IF NOT EXISTS adalah klausa opsional yang memungkinan untuk mengecek apakah suatu tabel sudah ada di dalam database atau belum

Kedua, tentukan daftar kolom dari suatu tabel di bagian column\_list, setiap kolom dipisahkan dengan koma.

Ketiga, tentukan mesin penyimpanan (storage engine) untuk tabel di klausa ENGINE (optional).

Note: Jika mesin penyimpanan tidak dideklarasikan, MySQL akan

menggunakan InnoDB secara default

**Primary Key**

Untuk mengatur primary key:

PRIMARY KEY (col1,col2,…)

**Create Table**

Misalkan kita akan membuat tabel baru yang disebut dengan tasks dengan struktur sebagai berikut.

Sintaks Create Table

CREATE TABLE IF NOT EXISTS tasks (

Task\_id INT AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

Start\_date DATE,

Due\_date DATE,

Status TINYINT NOT NULL,

Priority TINYINT NOT NULL,

Description TEXT,

PRIMARY KEY (task\_id)

) ENGINE=INNODB;

**Alter Table**

Untuk mengubah struktur dari tabel yang sudah ada. Biasanya digunakan untuk menambahkan kolom, menghapus kolom, mengubah tipe data kolom, menambahkan primary key, mengganti nama able, dsb

ALTER TABLE table\_name action1[,action2,…]

Pertama, tentukan table\_name yang akan diganti, setelah klausa ALTER TABLE

Kedua, buat daftar serangkaian action yang diterapkan ke suatu tabel Biasanya digunakan untuk menambahkan kolom, menghapus kolom, mengubah tipe data kolom, menambahkan primary key, mengganti nama tabel, dsb

Note: Statement ALTER TABLE dapat dilakukan dengan beberapa action dalam satu statement ALTER TABLE, serangkaian action ini dipisahkandengan koma (,).

Misalkan kita ingin mengubah tabel tasks

Tasks

Task\_id INT(11)

Title VARCHAR(255)

Start\_date DATE

Due\_date DATE

Status TINYINT(4)

Priority TINYINT(4)

Description TEXT

Mengatur kolom task\_id sebagi atribut auto-increment

ALTER TABLE tasks

CHANGE COLUMN task\_id task\_id INT(11) NOT NULL AUTO\_INCREMENT,

Menambahkan kolom baru yang disebut complete

ALTER TABLE tasks

ADD COLUMN complete DECIMAL(2,1) NUL

AFTER description;

Menghapuskan kolom description

ALTER TABLE tasks

DROP COLUMN description.

Untuk mengganti nama table

RENAME TABLE old\_table\_name to new\_table\_name

Untuk menampilkan struktur table

DESCRIBE table\_name

Untuk mengganti nama kolom

ALTER TABLE tablename change old\_column\_name new column\_name new\_data\_type(size);

Drop Table

Untuk menghapus table yang ada, kita dapat menggunakan statement DROP TABLE

DROP TABLE IF EXISTS taks,s nonexistent\_table;

Sintaks di atas mencoba untuk menghapus table yang tidak ada di dalam database untuk mendemonstrasikan statement SHOW WARNING

|  |  |  |
| --- | --- | --- |
| Level | Code | Message |
| Note | 1051 | Unknown table ‘nonexistent\_table’ |

Quit

Untuk keluar dari MySQL

Mysql> exit

Bye

DML

Data Maniplation Language (DML)

Suatu bahasa yang digunakan untuk oleh user untuk mengakses dan memanipulasi data

1. SELECT: Mendapatkan informasi atau record dari suatu database
2. UPDATE: memodifikasi data dari suatu atau beberapa record sesuai dnegna kondisi tertentu
3. INSERT: menambahka nsatu atau beberapa record ke dalam suatu tabel
4. DELETE: Menghapus satu atau beberapa record dari suatu database sesuai dengan kondisi tertentu.

SELECT

Mengambil data dari suatu atau berapa table dalam suatu database

Sintaks

Select [nama kolom]

FROM [nama\_tabel];

**Query**

SELCT

last name,

first name,

jobtitle

FROM

Employees

**SELECT semua kolom**

SELECT ALL FROM employees;

SELECT \* FROM employees;

SELECT DISTINCT

Menghapus baris yang duplikat

SELECT DISTINCT

List\_kolom

FROM

nama\_tabel;

**SELECT**

SELECT

Lastname

FROM

Employees

ORDER BY

Lastname;

**SELECT DISTINCT**

SELECT DISTINCT

Lastname

FROM

Employees

ORDER BY

Lastname;

**INSERT statement**

Menambahkan satu atau bebarap record ke satu database

UPDATE

Mengubah data yang ada dalam table atau mengubah nilai kolom dari satu baris , sekelompok baris, atau semua baris ke am table

Sintaks

UPDATE [nama\_tabel]

SET [nama\_kolom = nilai]

WHERE [kondisi];

DELETE

Menghapus satu atau beberapa record dari suatu database ssuai dengan kondisi tertentu

Untuk membatasi jumlah baris yang akan dihapus

DELETE FROM table

LIMIT row\_count;

Data Manupulation Languange Part 3

Fungsi Aggregat

Suatu fungsi yang perhitugnan

Max

Min

AVG

SUM

COUNT

MAX

Menghasilkan nilai maksimum dari sekumpulan nilai

Contoh

Menemukan angka terbesar

Produk yang palign mahal

Jumlah pembayaran terbesar dari pelanggan

Grouping Data

Kalusa Group BY digunakan untuk mengelompokkan sekumpulan baris menjadi sekumpiulan ringkasan data menurut nilai kolom atau ekspresi

Klausa GROUP BY menghasilkan satu baris untuk setiap grup

Kita sering menggunakan kalusa GROUP BY dengan fungsi agregat seperti SUM, AVG, MAX, MIN, dan COUNT, Fungsi agregat yang digunakan pada klausa SELECT memberikan informasi tentang setiap grup.

Klausa GROUP BY merupakan klausa opsional dari statement SELECT.

HAVING

Klausa Having digunakan dalam statement SELECT untuk menetapkan kondisi yang menjadi filter sekelompok baris atau agregat. Klausa HAVING sering digunakan dengan klausa GROUP BY untuk mem filter grup berdasarkan kondisi tertentu

Apabila klausa GROUP BY dihilangkan, klausa HAVING dijalanka nseperti WHERE

Perhatikan bahwa klausa HAVIGN menerapkan kondisi yang menjadi filter ke setiap grup baris, sdangkan klausa WHERE menerapkan kondisi filter ke setiap baris.

Inner JOIN

Carikan kasus kost dimana kasus tersebut mengambil lebih dari 1 tabel untuk perhitungannya

Sebutkan table apa saja yang dibutuhkan

JOIN

Primary kety dan foreign key digunakan untuk menghubungkan record atau data antara beberapa table. Data dari seluruh table yang ada perlu dilhiat seluruhnya untuk membuat informasi menjadi lengkap. Klausa JOIN digunakan untuk menggabungkan data dari dua atau beberapa table, berdasarkan kolom/ key yang berhubugnan di antara tersebut

Tipe Join

(Inner) join: mengembalikan data yang memiliki nilai key yang sama/ cocok di kedua table.

Right (Outer) Join: Mengembalikan semua dari table sebelah kanan dan menampilkan data dari table sebelah kanan yang cocok dengan kondis ijoin

Right (Outer) Join: mengembalikan semua data dari table sebelah kanan dan menampilakn data dari table sebelah kiri yang cocok dengan kondisi join.

Full (Outer Join): mengembalika nseluruh data/ record dari table kiri dan table kanan yang dikenai ON (termasuk data-data yang bernilai NULL)

Self Join: Inner join dari suatu table dengna dirinya sendiri

Union: mengembalikan data yang gabugnannya dari dua atau lebih statement SELECT

Inner Join

Mengembalikan data yang memiliki nilai key yang sama/cocok di kedua table

Sintaks

* INNER JOIN
* SELECT column1,column2
* FROM table1
* INNER JOIN table2
* ON

Inner Join

* Inner JOIN
* SELECT em.firstName, em.lastName, jo.jobTitle
* FROM employees as em
* INNER JOIN jobs as jo
* ON em.jobID = jo.jobID;

Left Join

SELECT em.firstName, em.lastName, jo.jobTitle

FROM employees as em

INNER JOIN jobs as jo

ON em.jobID = jo.JobsID