**MODUL PRAKTIKUM**

Mata Kuliah : Basis Data  
Pokok Bahasan : SQL (Structured Query Language)  
Disusun Oleh : Rosihan Ari Yuana, S.Si, M.Kom

SQL (Structured Query Language) di MySQL  
Disusun oleh: Rosihan Ari Y.

# Koneksi ke Database

Command: mysql –u username –p

# DDL (Data Definition Language)

## Membuat Database

Syntax: create database namadb;

Contoh:

create database siakad;

## Memilih Database

Syntax: use namadb;

Contoh:

use siakad;

## Membuat Tabel

Syntax:

create table namatabel  
(  
 Field1 tipedata,  
 Field2 tipedata,  
 .  
 .  
 [primary key (Field)]  
);

Contoh:

create table mhs

(

nim char(10),

nama varchar(100),

tgllhr date,

alamat text,

primary key (nim)

);

Tipe Data:

* Char(n) -> string(n) -> fixed length
* Varchar(n) -> string(n)
* Int -> bilangan bulat
* Float/double -> bilangan riil
* Date -> YYYY-MM-DD
* Datetime -> YYYY-MM-DD hh:ii:ss
* Text -> string()

## Menampilkan Daftar Tabel

Syntax:

show tables;

Contoh:

show tables;

## Menambahkan Field Ke Dalam Tabel

Syntax:

alter table namatabel add namafield tipe data;

Contoh:

alter table mhs add nohp varchar(20);

## Menghapus Field Dalam Tabel

Syntax:

alter table namatabel drop namafield;

Contoh:

alter table mhs drop nohp;

## Mengubah Nama Field atau Tipe Datanya

Syntax:

alter table namatabel change fieldlama fieldbaru tipedata;

Contoh:

alter table mhs change nohp nomorhp varchar(30);

## Menghapus Tabel

Syntax:

drop table namatabel;

Contoh:

drop table mhs;

## Mengubah Nama Tabel

Syntax:

Rename table namatabellama to namatabelbaru;

Contoh:

rename table mhs to mahasiswa;

## Menghapus Primary Key

Syntax:

alter table namatabel drop primary key;

Contoh:

alter table mhs drop primary key;

## Setting Primary Key

Syntax;

alter table namatabel add primary key (field1, field2, ...);

Contoh:

alter table mhs add primary key (nim);

## Melihat Struktur Fields Tabel

Syntax;

show columns in namatabel;

Contoh:

show columns in mhs;

## Latihan

Buat Database : siakad

Buat Tabel :

* Mhs
  + Nim varchar(10) Primary Key
  + NamaMhs varchar(100)
  + TglLhr date
  + noTelp varchar(20)
  + Alamat text
* Mk
  + KodeMK varchar(5) Primary Key
  + NamaMK varchar(100)
  + SKS int
  + Smt int
  + KodeDosen varchar(5)
* AmbilMK
  + NIM varchar(5) Primary Key
  + KodeMK varchar(5) Primary Key
  + NilaiMK int
* Dosen
  + KodeDosen varchar(5) Primary Key
  + NamaDosen varchar(100)

# DML (Data Manipulation Language)

## Insert Data (Record)

Syntax:

INSERT INTO mhs VALUES (‘K35170001’, ‘Eko’);

Contoh:

INSERT INTO mhs (nim, namamhs, tgllhr, notelp, alamat)  
VALUES (‘K1234’, ‘ROSIHAN ARI Y’, ‘1979-09-01’, ‘0271-1234567’, ‘JL. KUCING NO. 200 SOLO’);

## Latihan

Masukkan data-data berikut ini ke masing-masing tabel dalam database ‘siakad’

Tabel: mhs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NIM** | **NAMA MHS** | **TGL LAHIR** | **NO TELP** | **ALAMAT** |
| K1001 | AMIR HAMZAH | 1980-10-23 | 08560001234 | JL. SAMBAL NO 1 SOLO |
| K1002 | BUDI WASKITA | 1980-12-01 | 08130008888 | JL. CUKA NO 8 SOLO |
| K1003 | BUDIMAN | 1981-04-19 | 08126172839 | JL. KECAP NO 10 SOLO |
| K1004 | CANDRA | 1982-10-20 | 08110028172 | JL. SAUS NO 3 SURABAYA |
| K1005 | DEDI SUTORO | 1982-03-18 | 08152716266 | JL. MERICA NO 12 SEMARANG |

Tabel: mk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KODE MK** | **NAMA MK** | **SKS** | **SMT** | **KODE DOSEN** |
| MK001 | ALGORITMA | 2 | 1 | DS003 |
| MK002 | BASIS DATA | 3 | 2 | DS003 |
| MK003 | PBO | 3 | 2 | DS002 |
| MK004 | OPEN SOURCE | 2 | 1 | DS001 |
| MK005 | NETWORKING | 2 | 2 | DS004 |

Tabel: dosen

|  |  |
| --- | --- |
| **KODE DOSEN** | **NAMA DOSEN** |
| DS001 | MR. A |
| DS002 | MR. B |
| DS003 | MR. C |
| DS004 | MR. D |
| DS005 | MR. E |

Tabel: ambilmk

|  |  |  |
| --- | --- | --- |
| **NIM** | **KODE MK** | **NILAI** |
| K1001 | MK001 | 3 |
| K1001 | MK002 | 2 |
| K1001 | MK004 | 4 |
| K1002 | MK001 | 4 |
| K1002 | MK002 | 1 |
| K1003 | MK001 | 3 |
| K1003 | MK003 | 2 |
| K1003 | MK004 | 4 |

## Menampilkan Data (Filter Data)

Syntax:

SELECT field1, field2, ... FROM namatabel [WHERE syarat] [GROUP BY field] [HAVING syarat] [ORDER BY field DESC|ASC]

atau

SELECT \* FROM namatabel [WHERE syarat] [GROUP BY field] [HAVING syarat] [ORDER BY field DESC|ASC]

Contoh:

/\* menampilkan semua data (record) pada semua field dari tabel ‘mhs’ \*/

SELECT \* FROM mhs;

/\* menampilkan semua data pada semua field dari tabel ‘mhs’ diurutkan berdasarkan nama secara ascending \*/

SELECT \* FROM mhs ORDER BY namamhs ASC;

/\* hanya menampilkan field NIM dan nama mhs dari tabel ‘mhs’ \*/

SELECT nim, namamhs FROM mhs;

/\* menampilkan data lengkap mahasiswa yang bernim ‘K1001’ \*/

SELECT \* FROM mhs WHERE nim = ‘K1001’;

/\* menampilkan data matakuliah yang ada di semester 1 atau berSKS 2 \*/

SELECT \* FROM mk WHERE smt = ‘1’ OR sks = ‘2’;

/\* menampilkan data lengkap matakuliah yang ada di semester 1 dan diampu oleh dosen ‘DS003’ \*/

SELECT \* FROM mk WHERE smt = ‘1’ AND kodedosen = ‘DS003’;

/\* menampilkan data lengkap mk yang tidak diampu oleh dosen dengan kode ‘DS001’ \*/

SELECT \* FROM mk WHERE kodedosen <> ‘DS001’;

/\* menampilkan data mahasiswa yang lahir di tahun ‘1980’ \*/

SELECT \* FROM mhs WHERE year(tgllahir) = ‘1980’;

Atau

SELECT \* FROM mhs WHERE tgllahir LIKE ‘1980%’;

/\* menampilkan data mahasiswa yang lahirnya sebelum tahun 1982 \*/

SELECT \* FROM mhs WHERE year(tgllahir) < ‘1982’;

/\* menampilkan data mahasiswa yang menggunakan nomor HP dari operator TELKOMSEL \*/

SELECT \* FROM mhs WHERE notelp LIKE ‘0811%’ OR notelp LIKE ‘0812%’ OR notelp LIKE ‘0813%’;

/\* menampilkan data mhs yang rumahnya di SOLO (diurutkan berdasarkan nama scr descending) \*/

SELECT \* FROM mhs WHERE alamat LIKE ‘%SOLO’ ORDER BY namamhs DESC;

**Keterangan:**

Untuk penggunaan klausa GROUP dan HAVING akan dibahas kemudian.

## Hapus Data

Syntax:

DELETE FROM namatabel [WHERE syarat];

Contoh:

/\* menghapus semua data mahasiswa di dalam tabel mhs \*/

DELETE FROM mhs;

/\* menghapus data mahasiswa yang bernim K001 \*/

DELETE FROM mhs WHERE nim = ‘K001’ ;

/\* menghapus data mahasiswa yang tanggal lahirnya sebelum 10 Januari 1980 \*/

DELETE FROM mhs WHERE tgllhr < ‘1980-01-10’;

/\* menghapus data mahasiswa yang tgl lahirnya antara 01/01/1980 s.d 31/01/1980 \*/

DELETE FROM mhs WHERE tgllhr > ‘1980-01-01’ AND tgllhr < ‘1980-01-31’;

/\* menghapus data mahasiswa yang nama depannya ‘EKO’ \*/

DELETE FROM mhs WHERE namamhs LIKE ‘EKO%’;

/\* menghapus data mahasiswa yang namanya mengandung kata ‘EKO’ \*/

DELETE FROM mhs WHERE namamhs LIKE ‘%EKO%’;

/\* menghapus data mahasiswa yang beralamat di ‘SOLO’ atau namanya mengandung kata ‘EKO’ \*/

DELETE FROM mhs WHERE alamat = ‘SOLO’ OR namamhs LIKE ‘%EKO%’

/\* menghapus data mahasiswa yang alamatnya tidak ‘SOLO’ \*/

DELETE FROM mhs WHERE alamat != ‘SOLO’;

Atau

DELETE FROM mhs WHERE NOT(alamat = ‘SOLO’);

## Update Data

Syntax:

UPDATE namatabel SET namamhs = ‘xxxx’, field2 = ‘nilai2’, ... [WHERE syarat];

Contoh:

/\* mengubah semua data mahasiswa dengan nama ‘EKO’

UPDATE mhs SET namamhs = ‘EKO’;

/\* mengubah semua data mahasiswa dengan nama ‘EKO’ dan beralamat di ‘SOLO’ \*/

UPDATE mhs SET namamhs = ‘EKO’, alamat = ‘SOLO’;

/\* mengubah nama mahasiswa yang bernim ‘M01’ menjadi ‘AMIR’ \*/

UPDATE mhs SET namamhs = ‘AMIR’ WHERE nim = ‘M01’;

/\* mengubah tanggal lahir mahasiswa bernama ‘BUDI’ dan beralamat di SOLO menjadi ‘1981-12-30’ \*/

UPDATE mhs SET tgllhr = ‘1981-12-30’ WHERE namamhs = ‘BUDI’ AND alamat = ‘SOLO’;

## SELECT Statement (Lanjutan)

### Query Relasi Tabel

/\* menampilkan NIM, NAMA MHS, NILAI yang mengambil mata kuliah berkode ‘MK001’ \*/

SELECT mhs.nim, mhs.namamhs, ambilmk.nilai  
FROM mhs, ambilmk  
WHERE mhs.nim = ambilmk.nim AND ambilmk.kodemk = ‘MK001’;

/\* menampilkan KODE MK, NAMA MK, SKS, NAMA DOSEN PENGAMPU \*/

SELECT mk.kodemk, mk.namamk, mk.sks, dosen.namadosen  
FROM mk, dosen  
WHERE mk.kodedosen = dosen.kodedosen;

/\* menampilkan NAMA DOSEN, NAMA MK yang SKS nya lebih dari 2 \*/

SELECT dosen.namadosen, mk.namamk  
FROM dosen, mk  
WHERE dosen.kodedosen = mk.kodedosen AND mk.sks > 2;

/\* menampilkan NIM, NAMA MHS , KODE MK, NAMA MK, SKS, SMT, NILAI, NAMA DOSEN \*/

SELECT mhs.nim, mhs.namamhs, mk.kodemk, mk.namamk, mk.sks, mk.smt, ambilmk.nilai, dosen.namadosen  
FROM mhs, mk, ambilmk, dosen  
WHERE mhs.nim = ambilmk.nim AND mk.kodemk = ambilmk.kodemk AND mk.kodedosen = dosen.kodedosen;

### Aggregate Function

/\* mencari total SKS semua matakuliah \*/

SELECT sum(sks) FROM mk;

/\* mencari rata-rata SKS semua matakuliah \*/

SELECT avg(sks) FROM mk;

/\* mencari banyaknya mahasiswa yang mengambil matakuliah MK001 \*/

SELECT count(\*) FROM ambilmk WHERE kodemk = ‘MK001’;

/\* mencari rata-rata nilai mahasiswa yang mengambil matakuliah ‘MK001’ \*/

SELECT avg(nilai) FROM ambilmk WHERE kodemk = ‘MK001’;

/\* mencari IP mahasiswa bernim ‘K1001’ \*/

SELECT sum(ambilmk.nilai \* mk.sks)/sum(mk.sks)   
FROM ambilmk, mk  
WHERE ambilmk.kodemk = mk.kodemk AND ambilmk.nim = ‘K1001’;

SELECT sum(ambilmk.nilai\*mk.sks)/sum(mk.sks) as IP

FROM ambilmk.mk

WHERE ambilmk.kodemk = mk.kodemk

GROUP BY ambilmk.nim

HAVING IP < 2

Tampilkan KODEMK, NAMAMK, NAMADOSEN dari matakuliah yang rata-rata nilainya > 3

SELECT mk.kodemk, mk.namamk, dosen.namadosen

FROM mk, ambilmk, dosen

WHERE mk.kodemk = ambilmk.kodemk AND mk.kodedosen = dosen.kodedosen

GROUP BY ambilmk.nim

HAVING avg(ambilmk.nilai)>3;

Tampilkan KODEMK, NAMAMK, SKS, SMT, NAMADOSEN, JUMLAH MHS yang jumlah mahasiswanya kurang dari 2

SELECT mk.kodemk, mk.namamk, mk.sks, dosen.kodedosen, count(\*) as jumlah

FROM mk, ambilmk, dosen

WHERE mk.kodemk = ambilmk.kodemk AND dosen.kodedosen = mk.kodedosen

GROUP BY mk.kodemk

HAVING jumlah < 2;

Tampilkan NIM, NAMAMHS, IP diurutkan dari yang IP nya tertinggi

SELECT mhs.nim, mhs.namamhs,sum(ambilmk.nilai\*mk.sks)/sum(mk.sks) as IP

FROM mhs, ambilmk, mk

WHERE mhs.nim = ambilmk.nim AND mk.kodemk = ambilmk.kodemk

GROUP BY mhs.nim

ORDER BY DESC

LIMIT 0,2; (Biasanya digunakan untuk Paging)

Limit(Data ke berapa, berapa yang ditampilkan)

/\* mencari berapa jumlah SKS paling besar di antara semua matakuliah \*/

SELECT max(SKS) FROM mk;

### GROUP BY Clause

/\* menampilkan KODEMK, NAMA MK, JUMLAH MAHASISWA \*/

SELECT mk.kodemk, mk.namamk, count(\*) as `jum mhs`  
FROM mk, ambilmk  
WHERE mk.kodemk = ambilmk.kodemk  
GROUP BY mk.kodemk;

/\* menampilkan KODEMK, NAMAMK, RATA-RATA NILAI \*/

SELECT mk.kodemk, mk.namamk, avg(ambilmk.nilai) as `Rata-rata nilai`  
FROM mk, ambilmk  
WHERE mk.kodemk = ambilmk.kodemk  
GROUP BY mk.kodemk;

/\* menampilkan KODE DOSEN, NAMA DOSEN, TOTAL SKS yang diampunya pada semester 2 \*/

SELECT dosen.kodedosen, dosen.namadosen, sum(mk.sks)  
FROM dosen, mk, ambilmk  
WHERE mk.kodemk = ambilmk.kodemk AND dosen.kodedosen = mk.kodedosen AND mk.smt = '2'  
GROUP BY dosen.kodedosen;

/\* menampilkan NIM, NAMA MHS, IP seluruh mahasiswa \*/

SELECT mhs.nim, mhs.nama, sum(ambilmk.nilai \* mk.sks)/sum(mk.sks) AS `IP`   
FROM ambilmk, mk, mhs  
WHERE ambilmk.kodemk = mk.kodemk AND mhs.nim = ambilmk.nim  
GROUP BY mhs.nim;

/\* menampilkan KODE DOSEN, NAMA DOSEN, TOTAL SKS yang total SKS diampunya kurang dari 4 \*/

SELECT dosen.kodedosen, dosen.namadosen, sum(mk.sks)  
FROM dosen, mk, ambilmk  
WHERE mk.kodemk = ambilmk.kodemk AND dosen.kodedosen = mk.kodedosen  
GROUP BY dosen.kodedosen  
HAVING sum(mk.sks) < 4;

/\* menampilkan KODE MK, NAMA MK, JUMLAH MAHASISWA yang jumlah mahasiswanya kurang dari 3 orang \*/

SELECT mk.kodemk, mk.namamk, COUNT(\*)  
FROM ambilmk, mk  
WHERE ambilmk.kodemk = mk.kodemk  
GROUP BY mk.kodemk  
HAVING count(\*) < 3;

# Data Control Language (DCL)

## Mengubah Password ROOT

Syntax:

SET PASSWORD FOR 'root'@'localhost' = PASSWORD('XXX');

Keterangan:

XXX adalah password yang baru

## Menambah User MySQL Baru

Syntax:

CREATE USER ‘namauser’@’namahost’ IDENTIFIED BY ‘password’

Keterangan:

* namahost adalah nama host atau nomor IP dari mana si user bisa login ke server MySQL.
* Jika ingin si user hanya bisa akses MySQL dari komputer di mana MySQL berada, maka isikan ‘localhost’
* Jika ingin si user bisa akses dari sembarang host ke server MySQL, maka isikan ‘%’

Contoh:

CREATE USER ‘rosihanari’@’192.168.1.7’ IDENTIFIED BY ‘12345’;

CREATE USER ‘rosihanari’@’localhost’ IDENTIFIED BY ‘12345’;

CREATE USER ‘rosihanari’@’%’ IDENTIFIED BY ‘12345’;

## Memberi Hak Akses Suatu User ke Database Tertentu

Syntax:

GRANT hak1, hak2, ..., hakn ON namadatabase.namatabel   
TO 'namauser'@'namahost';

Keterangan:

Macam-macam hak yang bisa diberikan kepada seorang user:

* SELECT (pencarian data), INSERT (insert data), UPDATE (update data), DELETE (hapus data)
* CREATE (buat tabel), ALTER (ubah tabel), DROP (hapus tabel)

Contoh:

GRANT SELECT, INSERT, DELETE, CREATE ON siakad.\* TO 'user1'@'192.168.1.7';

Keterangan:

Tanda \* menunjukkan hak di atas diterapkan pada semua tabel dalam database ‘siakad’. Jika hak tersebut ingin diberikan pada salah satu tabel, maka query nya

GRANT SELECT, INSERT, DELETE, CREATE ON siakad.mhs TO 'user1'@'192.168.1.7';

Contoh:

GRANT ALL PRIVILEGES ON namadb.\* TO namauser@'192.168.1.2';

Keterangan:  
Memberi hak akses FULL ke database tertentu kepada user tertentu dari IP 192.168.1.2

## Import Data

mysql –h noip -u username -p namadatabase < dataku.sql

## Hapus User

DROP USER 'username'@'nomor ip';

## Mencabut Semua Hak Akses Seorang User

Syntax:

REVOKE ALL PRIVILEGES ON namadatabase.namatabel FROM 'namauser'@'namahost';

Contoh:

REVOKE ALL PRIVILEGES ON siakad.\* FROM 'user2'@'localhost';

## Cara Mereset Password ROOT Jika Lupa

Silakan mempelajari di <http://dev.mysql.com/doc/refman/5.0/en/resetting-permissions.html>