

Putty 192.168.8.251

Login :

Password :

MySQL>Create Database <namabasisdata>; membuat databse
 MySQL>Show Databases; menampilkan database
 MySQL>drop database <namabasisdata>; Menghapus databse
 MySQL>Use <namabasisdata>; mengkoneksi/membuka ke databade
 MySQL>Show Tables; menampilkan table
 MySQL> Create Table <namatable> (nama_field1 Tipe_data (ukuran), nama_field2 Tipe_data (ukuran)),); untuk menciptakan tabel
 MySQL>Desc <namatable> ; untuk menampilkan struktur table;

Contoh :

Create table mhs (nim char(14) not null primary key, nama char(25), Alamat char(30), Sex char(1) default 'P' check (sex='W' or sex='P'), Agama enum ('1', '2', '3', '4', '5','6'), tgl_lhr date());

Memaniplulasi Struktur tabel;

MySQL>alter table mhsx **Add** no int(3) first;

MySQL>alter table mhsx **Add** ket char(15) after agama;

MySQL>alter table mhsx **Change** sex jnskel char(1);

MySQL>alter table mhsx **Drop** ket;

MySQL>rename table mhsx to mahasiswa; untuk mengganti tabel

MySQL>drop table dosen; untuk menghapus tabel dosen

MySQL>CREATE INDEX idxmhs On mhs (nim);

- Merubah data:

UPDATE nama_tabel SET kolom_1 = nilai_baru_1, kolom_2 = nilai_baru_2, [Where kondisi];

Contoh : update mhs set nama="Agus", sex="P" where nim="A11.2010.011111";

- Hapus Data: DELETE FROM nama_tabel [WHERE kondisi];
Delete from mhs where nim="A11.2010.011111";

Operator Between, In, Like, and, or, not

Dan lain-lain

1. Mata Kuliah : Sistem Basis Data II
2. Bobot SKS : 2 SKS
3. Program Studi : Manajemen Informatika - Diploma III
4. Tujuan Inst. Umum : Setelah mengikuti mata kuliah ini mahasiswa mampu menggunakan perangkat lunak MySQL untuk pengelolaan database yang kompleks
5. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat mengetahui sejarah tentang MySQL, cara mendapatkan perangkat lunak MySQL dan dokumen manualnya.
6. Persiapan : Mahasiswa harus sudah dapat masuk ke jaringan komputer laboratorium agar dapat bekerja dan menggunakan MySQL.

PRAKTIKUM KE – I

1. Judul Materi : Pengantar MySQL
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat mengetahui sejarah tentang MySQL, cara mendapatkan perangkat lunak MySQL dan dokumen manualnya.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Sistem Basis Data I, Pemrograman Database
5. Teori Dasar :

PENGANTAR MySQL

1. Apa itu MySQL

MySQL adalah sebuah system manajemen database relasi (RDBMS) yang bersifat terbuka (open source), yaitu MySQL dapat di download oleh siapa saja, baik versi kode program aslinya (source code program) maupun versi binernya (executable program) dan bisa digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

Bila kita seorang programmer dan ikut berpartisipasi dalam memodifikasi program asli MySQL untuk keperluan sendiri atau perusahaan sendiri, maka kitadiwajibkan secara moral untuk memberikan hasil modifikasi tersebut kepada MySQL dan komunitas open source. Hal ini merupakan konsekuensi dari konsep open source. Tetapi bila kita ingin merahasiakan hasil modifikasi tersebut, maka kita wajib membayar sejumlah dana kepada MySQL AB sebagai biaya lisensi.

MySQL menggunakan bahasa standar SQL (Structure Query Language) sebagai bahasa interaktif dalam mengelola data (sering disebut : query).

Sejarah MySQL yang merupakan hasil buah pikiran dari Michael “Monty Widenius, David Axmark, dan Allan Larson dimulai tahun 1995. Pada bulan juni 2000, MySQL AB mengumumkan bahwa mulai MySQL versi 3.23.19 diterapkan sebagai General Public License (GPL). Beberapa penghargaan pun diperoleh oleh MySQL, terutama dari majalah-majalah dengan topik Linux. Hal ini disebabkan pada awalnya MySQL hanya berjalan di atas platform Sistem Operasi Linux. Bahkan hampir semua distro

Linux yang beredar saat ini telah memasukkan program MySQL sebagai database standarnya. Tetapi, saat ini MySQL dapat pula berjalan di atas system operasi lainnya seperti Microsoft Windows, Solaris, FreeBSD, IBM's AIX, Mac OS X dan lain-lain.

2. Benarkah MySQL Gratis ?

Ada dua jenis lisensi :

- a. Lisensi gratis (free software / open source GNU general public license (GPL)).
Seperti yang telah dijelaskan di atas, dan MySQL tidak menjamin adanya layanan dukungan dari MySQL AB.
- b. Lisensi komersial (Non GPL commercial license).
Kita harus membayar sejumlah dana kepada MySQL AB sesuai dengan jenis layanan tertentu yang disediakan oleh MySQL AB, atau bila :
 - ✓ Mendistribusikan MySQL dan aplikasinya dengan menggunakan program-program bukan open source.
 - ✓ Menghendaki adanya jaminan dari MySQL AB tentang kinerja MySQL.
 - ✓ Menghendaki dukungan teknis dan non-teknis dari MySQL AB.

3. Mengapa Memilih MySQL ?

Ada beberapa pertimbangan mengapa memilih MySQL :

- a. Kecepatan. MySQL mempunyai kecepatan paling baik disbanding database server lainnya (lihat <http://www.mysql.com/it-resource/benchmarks/eweek.html>)
- b. Mudah Digunakan.
- c. Open Source
- d. Kapabilitas. MySQL telah digunakan untuk mengelola database dengan jumlah 50 juta record. Bahkan sanggup untuk mengelola 60.000 tabel dengan jumlah record 5 juta. Mendukung penggunaan index hingga 32 buah index pertabelnya.
- e. Biaya Rendah. Konektifitas dan Keamanan.
- f. MySQL mendukung dan menerapkan system keamanan dan izin akses tingkat lanjut (advanced permissions and security system), termasuk dukungan pengamanan dengan cara pengacakan lapisan data.
- g. Portabilitas.
- h. MySQL mendukung perintah-perintah ANSI SQL 99 dan beberapa perintah database alternatif lainnya sehingga memudahkan untuk beralih dari dan ke MySQL.
- i. Lintas Platform system operasi. MySQL dapat dijalankan pada beberapa system operasi yang berbeda, seperti Linux, Microsoft Windows, Sun Solaris, Novell Netware, IBM's AIX, Mac OS X dll.

4. Dimana mendapatkan MySQL ?

MySQL dapat di download langsung dari situs MySQL AB (www.mysql.com/downloads/). Sebaiknya kita memilih rilis terbaru yang stabil (biasanya diberi keterangan “recommended”). Rilis terakhir yang disarankan per bulan Agustus 2003 adalah rilis 4.0.14. Secara umum, MySQL terdiri dari dua versi yaitu standar dan max. Versi standar disarankan bagi para pengguna secara umum, sedangkan versi max memiliki beberapa fitur tambahan disbanding versi standar.

PRAKTIKUM KE – II

1. Judul Materi : Memulai Menggunakan MySql
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat menggunakan MySql dan perintah-perintah dasarnya dengan baik.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Sistem Basis Data I, Pemrograman Database
5. Teori Dasar :

MEMULAI MENGGUNAKAN MySQL

MySQL bekerja dibawah sistem operasi Linux, sehingga untuk dapat menggunakan MySQL kita harus masuk dahulu ke jaringan Linux. Urutan langkah untuk dapat menggunakan MySQL adalah sebagai berikut :

- Klik START dari layar desktop anda
- Klik pilihan RUN, kemudian pada isian Open, tuliskan perintah :

Putty 192.168.8.251

Atau

Telnet 192.168.8.251

- Setelah muncul pertanyaan untuk memasukkan nama login

Login :

Isikan nama login anda.

- Tampilan yang muncul selanjutnya adalah prompt sebagai berikut :

[namalogs@gw namalogs]_

tampilan prompt diatas sebagai tanda bahwa anda sudah masuk kedalam jaringan Linux.

- Dan untuk masuk ke MySQL ketikkan : mysql setelah prompt diatas :

[namalogs@gw namalogs] mysql

Setelah anda enter, maka prompt yang muncul selanjutnya adalah :

[mysql] _

Prompt sebagai tanda bahwa anda sudah ke MySQL.

OPERASI DASAR SQL

Sebelum menuliskan operasi-operasi di MySQL, maka ada beberapa hal yang harus diperhatikan, yaitu tentang tipe data dan operator aritmatika yang berlaku di MySQL.

Tipe Data menurut ANSI SQL adalah :

Tipe Data	Keterangan
Char atau Character	Menyatakan eretan karakter (string)
Numeric	Menyatakan data numeric real
Decimal atau Dec	Identik dengan numeric dan dapat mengandung pecahan
Integer	Menyatakan tipe data bilangan bulat
Smallint	Menyatakan tipe data bilangan bulat yang jangkauannya lebih kecil dari integer
Float	Tipe data bilangan real
Double	Tipe data bilangan real dengan presisi lebih tinggi dari float
Date	Untuk menyatakan tanggal
Time	Untuk menyatakan waktu
Boolean	Berisi tipe data logika false atau true

Beberapa Operasi Dasar SQL :

Membuat Basis Data

Create Database <namabasisdata>;

; -> Akhir
dari perintah

Melihat nama-nama Basisdata yang sudah ada

MySQL>Show Databases;

Mengkoneksikan ke Sebuah BasisData

Use <namabasisdata>;

Melihat nama-nama table dalam suatu Basisdata

MySQL>Show Tables;

Menampilkan Struktur Suatu Table

Desc <namatable>;

Contoh :

MySQL>Desc Mhs;

Membuat Tabel Baru

Create Table <namatable> (nama_field1 Tipe_data (ukuran), nama_field2 Tipe_data (ukuran),);

Contoh :

MySQL>Create Table Mhs (NIM Char(14), Nama Char (20), Tgl_lhr Date);

Memasukkan Data

Insert into <namatable> Values ('.....', '.....', '.....');

Contoh :

MySQL>Insert Into Mhs (nama,nim,tgl_lhr)Values ('Susiana', 'A21.2000.01232', '1977-03-11');

Melihat Isi Tabel

Select * From <namatable>;

Contoh :

```
MySQL>Select * From mhs;
      Select nim, nama, alamat, sex, tglhr From mhs;
      Select mhs.nim, mhs.nama, mhs.alamat, mhs.sex, mhs.tglhr From mhs;
      Select A.nim, A.nama, A.alamat, A.sex, A.tglhr From mhs A;
```

Apabila ingin menampilkan nama_field tertentu dengan kriteria tertentu, maka perintah diatas dikembangkan sendiri dengan menggunakan perintah - perintah yang telah dipelajari sebelumnya.

Contoh :

Menampilkan nama mahasiswa yang nimnya A21.2000.01232

```
MySQL>Select nama From Mhs where nim='A21.2000.01232';
```

Menambahkan sebuah field ke dalam Tabel

Alter Table <namatable> Add <namafield> tipe_data (ukuran);

Contoh :

Menambahkan field Sex dengan tipe karakter yang lebarnya 1

```
MySQL>Alter Table Mhs Add sex char(1);
```

Menghapus Field

Alter Table <namatable> Drop <namafield>;

Contoh :

Menghapus nama field sex

```
MySQL>Alter Table Mhs Drop Sex;
```

Mengganti Nama Field

Alter Table <namatable> Rename <namafield_lama> to <namafield_baru>;

Contoh :

Mengganti nama field Sex menjadi jenis_kel

```
MySQL>Alter Table Mhs Rename Sex to Jenis_kel;
```

Mengganti nama field serta ukuran

Alter Table <Namatable> Change <namafield_lama> <namafield_baru> tipe_data (ukuran);

Contoh :

Mengganti nama field NIM yang panjangnya 14 menjadi 15

```
MySQL>Alter Table Mhs Change nim nim char(15);
```

Menghapus Table

```
Drop Table <namatable>;
```

Contoh :

```
MySQL>Drop Table Mhs;
```

Latihan !!!

Buatlah sebuah tabel Mata Kuliah dengan Mkuliah dengan struktur sebagai berikut :

Nama Field	Type	Keterangan
Kode_kul	Char(5)	
Nama_kul	Char(20)	
SKS	Numeric	
Smt_tawar	Decimal(1,0)	1=Ganjil 2=Genap

Dari tabel diatas :

- ◆ Tambahkan field Kode_prak dengan tipe data Char(1).
- ◆ Isikan record sebanyak-banyak kedalam tabel tersebut.
- ◆ Gantilah ukuran field Nama_kul yang semula Char(20) menjadi Char(25)
- ◆ Tampilkan seluruh data mata kuliah
- ◆ Tampilkan hanya nama mata kuliah yang sksnya adalah 2
- ◆ Tampilkan hanya kode dan nama mata kuliah yang sksnya adalah 2 dan smt_tawarnya adalah Ganjil

PRAKTIKUM KE – III

1. Judul Materi : Kekangan Nilai Pada Data
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat Membuat Tabel dengan MySql beserta batasan-batasannya.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat menggunakan MySql dan Mengenal tipe-tipe data di MySql
5. Teori Dasar :

KEKANGAN NILAI PADA DATA

Pada bab ini yang akan dipelajari adalah bagaimana caranya agar data tidak bisa berisi null (tidak kosong), mencegah data yang kembar, mencegah nilai yang tidak valid, mengatur nilai bawaan.

Catatan :

Untuk mempraktekan operasi-operasi dibawah ini, maka hapuslah semua tabel yang ada pada databases Anda masing-masing.

Mengatur agar data selalu diisi :

Untuk mengatur agar data tertentu harus diisi (tidak kosong), dapat dilakukan dengan menambahkan kata kunci NOT NULL pada pendefinisian field yang bersangkutan.

Contoh :

```
MySQL>Create table mhs (nim char(15) not null, nama char(25) not null, alamat char(30));
```

Keterangan :

Field nim dan alamat tidak boleh kosong.

Menentukan Kunci Primer

Untuk menyebutkan kunci primer terhadap suatu field maka sertakan PRIMARY KEY pada pendefinisian kolom.

Contoh :

```
MySQL>Create table mhs (nim char(15) not null primary key,  
→ nama char(25) not null, alamat char(30));
```

Keterangan :

Field nim dan alamat tidak boleh kosong dan field nim dipilih sebagai primary key.

Membuat Kunci Komposit

Untuk membuat kunci primer yang terdiri dari lebih satu field, maka field-field tersebut ditulis setelah PRIMARY KEY sebagai berikut : PRIMARY KEY (field1, field2, ...) yang ditulis pada akhir deklarasi struktur tabel.

Contoh :

```
MySQL>Create table mhs (nim char(15) not null, nama char(25) not null,  
→ alamat char(30), primary key (nim, nama));
```

Keterangan :

Field nim dan alamat tidak boleh kosong dan sebagai primary key adalah field nim dan nama.

Membuat Nilai Yang Unik

Sebenarnya kunci primer sudah bertindak sebagai penjamin bahwa nilai suatu field adalah unik (tidak kembar). Tetapi operasi ini digunakan untuk membuat suatu field yang bukan primer menjadi unik, yaitu dengan menambahkan UNIQUE pada deklarasi field yang bersangkutan.

Contoh :

```
MySQL>Create table mhs (nim char(15) not null primary key,  
→ nama char(25) not null unique, alamat char(30));
```

Keterangan :

Field nama tidak boleh kosong dan nilai juga harus unique.

Memberi Nilai Bawaan

Apabila kita tidak memberikan suatu nilai pada field tertentu maka field tersebut akan terisi dengan null. Kita dapat mengatur agar kalau suatu field tidak diberi nilai eksplisit maka nilai bawaan tertentu akan diisikan ke field tersebut. Yaitu dengan menambahkan kata kunci default pada field bersangkutan sewaktu tabel didefinisikan.

Contoh :

```
MySQL>Create table mhs (nim char(15) not null primary key,  
→ nama char(25) not null unique, alamat char(30), sex char(1) default 'W');
```

Keterangan :

Pada saat sex tidak diisikan dengan suatu nilai, maka nilai bawaan 'W' yang akan disimpan.

Memvalidasi Nilai

Untuk mengatur nilai tertentu saja yang dapat diberikan ke suatu field, maka dapat dilakukan dengan memberikan kata kunci CHECK diikuti dengan suatu kondisi pada field tertentu.

Contoh:

Create table mhs (nim char(14) not null primary key, nama char(25) not null unique, alamat char(30), sex enum ('W','P'), sts int(1) default 0 check(sts=0 or sts=1));

MySQL>Create table mhs (nim char(14) not null primary key,
 → nama char(25) not null unique, alamat char(30),
 → sex char(1) default 'W' check (sex = 'W' or sex = 'P'));

Keterangan :

Pada saat sex tidak diisikan dengan suatu nilai, maka nilai bawaan 'W' yang akan disimpan. Nilai yang dapat diterima oleh field sex adalah W atau P.

Latihan !!!

1. Buatlah sebuah tabel dengan nama mhs dengan struktur tabel sebagai berikut :

Nama Field	Type	Keterangan
Nim	Char(14)	Primary key, unik dan tidak boleh kosong
Nama	Char(25)	Tidak boleh kosong dan unik
Alamat	Char(30)	
Sex	Char(1)	Hanya dapat diisikan P (pria) atau W (wanita)
Agama	Char(1)	Hanya dapat diisikan angka 1 s.d 5 (1=islam, 2=katholik, 3=kristen, 4=hindu, 5=budha) dan apabila tidak diisikan maka nilai yang tersimpan adalah 1.
Tgl_lhr	Date	

Perintah :

Create table mhs (nim char(15) not null primary key, nama char(25) not null unique, Alamat char(30), Sex char(1) check (sex='W' or sex='P'), Agama char(1) default '1' check (agama in ('1', '2', '3', '4', '5')), tgl_lhr date);

2. Tambahkan record-record berikut :

'A21.2001.00234','Dedy Sutanta','Jl. A. Yani 20 Solo','P','1','1979-10-20');
 'A21.2001.00214','Budi Satria','Jl. Dago 10 Yogya','P','1','1978-11-22');
 'A21.2001.00224','Sulis','Jl. Nakula I 25 Smg','W','2','1975-08-16');

Perintahnya :

3. Tambahkan record-record berikut :

'A21.2001.00234','Suci Andriani','Jl. Seroja Raya Smg','W','4','1977-09-21');

Perintahnya :

Pesan apa yang muncul :

Mengapa demikian :

Apakah data tersebut tersimpan :

4. Tambahkan record-record berikut :

'A21.2001.00344','Suci Andriani','Jl. Seroja Raya Smg','W','6','1977-09-21');

Perintahnya :

Pesan apa yang muncul :

Mengapa demikian :

Apakah data tersebut tersimpan :

5. Tuliskan perintah berikut :

Insert into mhs (nim, nama) values ('A21.2001.00355','Fadly');

Apakah data tersebut tersimpan :

Mengapa demikian :

6. Tuliskan perintah berikut :

Insert into mhs (nim, alamat, sex) values ('A21.2001.00556','Jl. Arjuna 10 Semarang','W');

Pesan apa yang muncul :

Apakah data tersebut tersimpan :

Mengapa demikian :

7. Tuliskan perintah berikut :

Insert into mhs (nim, nama, alamat, sex, Agama) values ('A21.2001.00655','Gondo

Suwondo','Jl. Arjuna Raya 20 Semarang','P');

Apakah data tersebut tersimpan :

Apa nilai yang tersimpan dari field agama pada record tersebut :

Mengapa demikian :

8. Tuliskan perintah berikut :

Insert into mhs (nim, nama, alamat, sex, Agama) values ('A21.2001.00655','Gondo

Suwondo','Jl. Arjuna Raya 20 Semarang','P','1975-07-12');

Apakah data tersebut tersimpan :

Mengapa demikian :

Setelah selesai anda tulis semua kejadian diatas, kemudian kumpulkanlah hasil pekerjaan anda.

PRAKTIKUM KE – IV

1. Judul Materi : Memanipulasi Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat menggunakan perintah-perintah MySql untuk memanipulasi tabel
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat membuat tabel beserta dengan batasannya
5. Teori Dasar :

MEMANIPULASI TABEL

Memasukkan Nilai NULL

Apabila kita bermaksud untuk tidak mengisi suatu field, maka kita dapat memberikan nilai NULL kedalam field tersebut.

Contoh :

```
MySQL>Insert into mhs (nim, nama, alamat, sex)
→ values ('A21.2001.02123','Andy', NULL, 'P');
```

Mengubah Data dengan UPDATE

Untuk mengubah data yang terdapat pada record tertentu, digunakan pernyataan UPDATE dengan syntax sebagai berikut :

```
UPDATE nama_tabel SET kolom_1 = nilai_baru_1, kolom_2 = nilai_baru_2, .... [Where kondisi];
```

Contoh 1:

Akan diubah nama mahasiswa yang NIMnya = A21.2001.02123 menjadi Andi Azis.

```
MySQL>Update mhs set nama = 'Andi Azis' Where nim = 'A21.2001.02123';
```

Contoh 2 :

Akan diubah nama dan alamat mahasiswa yang NIMnya = A21.2001.02123 menjadi Andi Achmad, Jl. A. Yani 20. Semarang.

```
MySQL>Update mhs set nama = 'Andi Achmad' , alamat = 'Jl. A. Yani 20. Semarang' Where  
nim = 'A21.2001.02123';
```

Contoh 3 :

Akan diubah field sex semua mahasiswa menjadi NULL.

```
MySQL>Update mhs set sex = null;
```

Menghapus Data dengan DELETE

Untuk menghapus record, maka digunakan pernyataan DELETE, dengan syntax sebagai berikut :

```
DELETE FROM nama_tabel  
[WHERE kondisi];
```

Contoh 1 :

Akan dihapus data mahasiswa yang nimnya = A21.2001.02123

```
MySQL>Delete from mhs  
→ Where nim = 'A21.2001.02123';
```

Contoh 2 :

Akan dihapus data mahasiswa yang sex = 'P'

```
MySQL>Delete from mhs  
→ Where sex = 'P';
```

Contoh 3 :

Akan dihapus semua record dari tabel mahasiswa

```
MySQL>Delete from mhs;
```

Menampilkan semua field dan semua record

Untuk menampilkan seluruh field dan record dari tabel tertentu, maka digunakan perintah sebagai berikut :

```
Select nama_field1, nama_field2, .... From mhs [Where kondisi];
```

Contoh 1 :

Menampilkan seluruh field dan record mahasiswa

MySQL>Select * From mhs;

Contoh 2 :

Menampilkan field nim dan nama mahasiswa yang sex = 'W'

MySQL>Select nim, nama From mhs Where sex = 'W';

Memperoleh Data Yang Unik

Untuk memperoleh data yang unik (tidak ada yang kembar), maka digunakan pernyataan DISTINCT pada saat menuliskan perintah Select.

Bandingkan perintah berikut :

MySQL>Select sex → From mhs;	MySQL>Select Distinct sex → From mhs;
---------------------------------	--

Menggunakan Operator BETWEEN

Operator between digunakan untuk menangani operasi jangkauan.

Contoh :

Akan menampilkan nama mahasiswa yang nimnya antara A21.2001.00100 s.d A21.2001.00300

MySQL>Select nama From mhs

→ Where nim between 'A21.2001.00100' and 'A21.2001.00300';

Menggunakan Operator LIKE

Operator LIKE digunakan untuk mencari data tertentu yang mempunyai sifat kemiripan. Penggunaan operator LIKE dibarengi dengan simbol tertentu (wildcard) yaitu garis bawah (_) atau persen (%).

Tanda garis bawah (_) untuk mewakili sebuah karakter sedangkan (%) mewakili beberapa karakter.

Contoh 1 :

Menampilkan nama mahasiswa yang namanya diawali dengan huruf 'A'.

MySQL>Select nama From mhs Where nama like 'A%';

Contoh 2 :

Menampilkan nama mahasiswa yang namanya diawali dengan huruf B dan huruf ke empatnya adalah i.

MySQL>Select nama From mhs Where nama like 'B__i%';

Menggunakan Operator IN

Operator IN digunakan untuk mencocokkan suatu nilai dengan salah satu nilai yang ada pada suatu daftar nilai.

Contoh :

Bandingkan perintah dibawah ini :

<pre>MySQL> Select alamat From mhs → Where nama = 'Budi' or nama = → 'Wati' or nama = 'Andi' or nama = → 'Susi';</pre>	<pre>MySQL>Select alamat From mhs → Where nama in ('Budi', 'Wati', → 'Andi', 'Susi');</pre>
---	--

Menggunakan Operator Boolean AND, OR dan NOT

Operator boolean OR dan AND digunakan untuk query dengan kondisi yang majemuk.

Hasil ekspresi operator boolean AND, OR dan NOT dapat dilihat seperti tabel berikut :

Kondisi_1	Kondisi_2	AND	OR	NOT (kondisi_1)
Salah	Salah	Salah	Salah	Benar
Salah	Benar	Salah	Benar	Benar
Benar	Salah	Salah	Benar	Salah
Benar	Benar	Benar	Benar	Salah

Contoh 1 :

Akan menampilkan nama mahasiswa yang nimnya A21.2001.02123 dan sex = 'P'

```
MySQL>Select nama From mhs
  → Where nim = 'A21.2001.02123' and sex = 'P';
```

Contoh 2 :

Menampilkan nama mahasiswa yang sex nya selain P

```
MySQL>Select nama From mhs Where not sex = 'P';
```

Latihan !!!

Untuk mempratekkan perintah-perintah dibawah ini gunakan tabel mhs yang telah anda miliki.

Tuliskan perintah - perintah untuk menyelesaikan masalah-maslah dibawah ini dikertas, kemudian dikumpulkan.

- Tambahkan record berikut kedalam tabel mhs :
 - 'A21.2001.00001', 'Agus Wahyu', 'Jl. Nakula I No. 5 Smg', 'P', '2', '1976-08-12'
 - 'A21.2001.00002', 'Sari Wijaya', 'A. Yani. 20 Solo', 'W', '1', '1977-10-11'
 - 'A21.2001.00003', 'Lestari', 'Nakula raya 20 Semarang', 'W', '3', '1975-11-15'
- Tampilkan seluruh field dan record mahasiswa.
- Tampilkan nama mahasiswa yang sex = 'P'
- Tampilkan nama dan alamat mahasiswa yang nim antara A21.2001.00001 s.d A21.2001.00100'
- Tampilkan nama mahasiswa yang nama depannya adalah 'Sari'
- Tampilkan nim, nama dan alamat mahasiswa yang sex nya selain 'W'
- Gantilah nama mahasiswa yang nimnya 'A21.2001.00003' menjadi 'Lestari Handayani'

8. Gantilah alamat dan tanggal lahir mahasiswa yang nimnya 'A21.2001.00002' menjadi 'Sari Wijayanti', '1977-10-17'
9. Hapuslah record mahasiswa yang sex = 'P' atau agamanya = '3'
10. Hapuslah record-record mahasiswa yang nimnya antara A21.2001.00100 s.d A21.2001.00300'

PRAKTIKUM KE – V dan VI

1. Judul Materi : Memanipulasi Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat menggunakan perintah-perintah MySql untuk memanipulasi tabel
3. Jumlah Pertemuan : 2 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat membuat tabel beserta dengan batasannya
5. Teori Dasar :

MEMANIPULASI DATA (lanjutan)

Mengurutkan Data dg ORDER BY

Untuk mengurutkan data, maka kita dapat menggunakan ORDER BY. Pengurutan dapat dilakukan secara ascending (urut baik) maupun urut menurun (descending).

Contoh 1 :

Mengurutkan berdasarkan sebuah field (field yang dijadikan field kunci adalah field nama berdasarkan urutan naik).

```
MySQL>Select * From mhs Order by nama;
```

Contoh 2 :

Mengurutkan dengan urutan menurun (descending).

```
MySQL>Select * From mhs Order by nama Desc;
```

Contoh 3:

Mengurutkan berdasarkan beberapa field (field yang dijadikan field kunci adalah field sex dan nama).

```
MySQL>Select * From mhs Order by sex, nama;
```

Contoh 4:

Mengurutkan berdasarkan beberapa field (field yang dijadikan field kunci adalah field sex secara ascending dan nama secara descending).

MySQL>Select * From mhs Order by sex Asc, nama Desc;

Contoh 5:

Mengurutkan berdasarkan nomor kolom tertentu (akan diurutkan data berdasarkan kolom kedua dari field yang disebutkan dalam Select).

MySQL>Select nim, nama From mhs Order by 2 Desc;

Mengelompokkan Data dengan GROUP BY

Untuk mengelompokkan data, maka dapat digunakan perintah GROUP BY.

1. Tampilkan data mahasiswa yang Fakultasnya Ilmu Komputer?
2. Hitung jumlah mahasiswa Fasilkom?
3. Ada berapa Fakultas di udinus dan berapa jumlah mahasiswanya per Fakultas?
4. Ada berapa Progdi di Fakultas Ilmu Komputer dan berapa jumlah mahasiswanya per Progdi?
5. Berikan Diskon/potongan Sumbangan Fakultas (A=10%,B=20%,C=30%)

Jawab :

1. Select * from mhs where substr(nim,1,1)='A'
2. Select count(*) from mhs where substr(nim,1,1)='A'
3. Select left(nim,1) as Fakultas , count(*) as jmlmhs from mhs group by left(nm,1)
4. Select left(nim,3) as progdi , count(*) as jmlmhs from mhs where left(nim,1)='A' group by left(nm,3)

Contoh 1 :

Akan dikelompokkan data mahasiswa berdasarkan sex.

MySQL>Select nama, sex From mhs Group by sex;

Menentukan Kondisi pada GROUP BY

Untuk memberikan kondisi pada saat mengelompokkan data, maka kita dapat menggunakan HAVING.

Contoh 1 :

Akan dikelompokkan data mahasiswa berdasarkan agama yang agamanya adalah selain Budha (kode=5)

MySQL>Select agama From mhs Group by agama Having agama <> '5';

Menggunakan Fungsi Agregasi

Fungsi agregasi adalah fungsi yang disediakan oleh MySQL untuk menghasilkan sebuah nilai berdasarkan sejumlah data. Adapun beberapa fungsi agregasi pada MySQL adalah :

Fungsi Agregasi	Keterangan
AVG	Memperoleh nilai rata – rata
Count	Menghitung jumlah data
Max	Menghasilkan nilai terbesar
Min	Menghasilkan nilai terkecil
Sum	Menjumlahkan suatu nilai

Fungsi AVG

Digunakan untuk memperoleh nilai rata-rata.

Contoh 1 :

Menghitung nilai rata gaji pegawai.

MySQL>Select AVG(gaji) From pegawai;

Contoh 2 :

Menghitung nilai rata-rata pegawai per departemen.

MySQL>Select Depart, AVG(gaji) From pegawai Group by depart;

Contoh 3 :

Menghitung gaji rata pegawai di departemen Akunting.

MySQL>Select avg(gaji) From Pegawai Group by depart Having = 'Akunting';

Fungsi COUNT

Digunakan untuk melakukan pencacahan terhadap data.

Contoh 1 :

Menghitung jumlah pegawai

MySQL>Select Count(*) From pegawai;

Contoh 2 :

Menghitung jumlah pegawai per departemen

MySQL>Select depart, count(*) From Pegawai Group by depart;

Contoh 3 :

Menghitung jumlah pegawai per departemen, khusus untuk yang jumlah lebih dari 1.

MySQL>Select depart, count(*) From Pegawai Group by depart Having Count(depart) > 1;

Fungsi MAX

Digunakan untuk memperoleh nilai yang terbesar.

Contoh 1 :

Memperoleh gaji terbesar.

MySQL>Select Max(Gaji) From Pegawai;

Contoh 2 :

Menghitung gaji terbesar per departemen

MySQL>Select Max(Gaji) From Pegawai Group by depart;

Fungsi MIN

Digunakan untuk memperoleh nilai terkecil.

Contoh 1:

Memperoleh nilai terkecil

MySQL>Select Min(Gaji) From Pegawai;

Contoh 2 :

Menghitung gaji terkecil per departemen

MySQL>Select Min(Gaji) From Pegawai Group by depart;

Fungsi SUM

Digunakan untuk melakukan penjumlahan data.

Contoh 1 :

Menghitung total gaji

MySQL>Select sum(gaji) From Pegawai;

Contoh 2:

Menghitung total pegawai per departemen

MySQL>Select depart, sum(gaji) From Pegawai Group by depart;

Latihan !!!

1. Buatlah sebuah tabel pada database yang sudah anda miliki dengan nama tabel mkuliah dan struktur tabelnya adalah sebagai berikut :

Nama Field	Type Data	Keterangan
Kode_kul	Char(7)	Primary key, tidak boleh, unik
Nama_kul	Char(20)	Tidak boleh kosong
SKS	Decimal(2)	Tidak boleh kosong
Smt_tawar	Char(1)	Diisikan 1,2,3,4 dst
Klp_kul	Char(1)	Dapat diisikan angka 1-3 : 1 = MPK 2 = MKK 3 = MKB default adalah 1
Progdi	Char(1)	Dapat diisikan angka 1-2 : 1 = MI-D3 2 = SI-S1

2. Isikan data – data berikut :

'A21-101','Pancasila', 2, '1', '1', '1'
 'A21-102','Logika Algoritma', 4, '1', '2', '1'
 'A21-103','Pemrograman Basic', 2, '1', '2', '1'
 'A21-201','Sistem Basis Data I', 2, '2', '2', '1'
 'A21-202','Pemrograman Pascal', 4, '2', '2', '1'
 'A21-203','Ilmu Sosial Dasar', 2, '2', '1', '1'
 'A21-501','Bimbingan Karier', 2, '5', '3', '1'
 'A21-601','Proyek Akhir', 4, '6', '3', '1'

3. Tampilkan seluruh data dari tabel mkuliah.
4. Tampilkan Nama mata kuliah yang sksnya 2 dan ditawarkan disemester 6
5. Tampilkan nama mata kuliah yang sksnya lebih besar dari 2 sks
6. Tampilkan nama mata kuliah yang huruf depannya dimulai dengan 'P'
7. Gantilah nama mata kuliah 'Pancasila' menjadi 'Pendidikan Pancasila'
8. Gantilah nama mata kuliah 'Pemrograman Basic' menjadi 'Pemrograman (Basic)'
9. Tambahkan record-record berikut :
 'A12-202','Pemrograman Pascal', 4, '2', '2', '2'
 'A12-203','Kewiraan', 2, '2', '1', '2'
 'A12-501','Bimbingan Karier', 2, '5', '3', '2'
 'A12-601','Tugas Akhir', 6, '6', '3', '2'
10. Gantilah nama mata kuliah 'Kewiraan' menjadi 'Pendidikan Kewarganegaraan'
11. Hitunglah seluruh data yang telah tersimpan pada tabel mkuliah tersebut.
12. Tampilkan nama mata kuliah dan sks yang telah diurutkan berdasarkan kunci nama mata kuliah secara ascending
13. Tampilkan nama mata kuliah, sks, semester tawar dan program studi yang telah diurutkan berdasarkan field kunci semester tawar dengan urutan ascending dan field kunci program studi dengan urutan descending
14. Kelompokkan data mata kuliah berdasarkan kelompok mata kuliah.
15. Tampilkan kelompok mata kuliah dan hitung jumlah sks berdasarkan kelompok mata kuliah
16. Tampilkan kode program studi dan hitung jumlah sks berdasarkan kelompok program studi
17. Tampilkan jumlah sks terkecil untuk program studi MI-D3
18. Tampilkan jumlah sks terbesar untuk program studi MI-D3
19. Tampilkan jumlah sks terkecil untuk program studi SI-S1
20. Tampilkan sks rata-rata untuk mata kuliah kelompok MKP
21. Tampilkan sks rata-rata untuk mata kuliah kelompok MKB
22. Hitung jumlah sks untuk mata kuliah kelompok MPK

23. Hitung jumlah sks untuk mata kuliah kelompok MKK
24. Berapa jumlah mata kuliah yang termasuk dalam kelompok MPK

PRAKTIKUM KE – VII dan VIII

1. Judul Materi : Memanipulasi Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat menggunakan ekspresi-ekspresi waktu, tanggal dan ekspresi logika lainnya
3. Jumlah Pertemuan : 2 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat membuat tabel beserta dengan batasannya
5. Teori Dasar :

Operator Relasional

Ada beberapa macam operator relasional pada My Sql, yaitu :

Operator	Keterangan
=	Sama dengan
>	Lebih besar dari
<	Lebih Kecl dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan
<>	Tidak sama dengan

Contoh :

Menampilkan nama pegawai yang gajinya diatas 1,000,000 dari tabel pegawai.

```
MySQL>Select nama_peg From Pegawai Where gaji > 1000000;
```

Lebih lanjut dengan SELECT

Sejauh ini nilai yang ditampilkan pada select selalu berupa nama field. Tetapi sebetulnya kita juga dapat menggunakan ekspresi pada kolom-kolom yang ditentukan pada perintah Select.

- ◆ Menuliskan ekspresi pada kolom

Contoh :

Akan menampilkan seluruh nama dan gaji pegawai setelah gaji dinaikkan 5% dari gaji semula

```
MySQL>Select nama, gaji + (0.05*gaji) From Pegawai;
```

- ◆ Memberi nama lain terhadap kolom

Contoh :

Akan menampilkan seluruh nama dan gaji pegawai setelah gaji dinaikkan 5% dari gaji semula dan nama kolomnya adalah gjbaru

```
MySQL>Select nama, gaji + (0.05*gaji) as gjbaru From Pegawai;
```

- ◆ Menambahkan Teks dalam baris keluaran

Pernyataan select memungkinkan pemakai menambahkan suatu teks pada baris-baris keluaran.

Contoh :

Akan menampilkan nama dan gaji seluruh pegawai dengan menambahkan teks rupiah setelah angka gaji.

```
MySQL>Select nama, gaji , 'Rupiah' From pegawai;
```

- ◆ Pemakaian Select tanpa klausa FROM

Cara ini digunakan untuk mengevaluasi ekspresi tertentu yang tidak ada kaitannya dengan tabel.

Contoh 1 :

Akan dihitung nilai penambahan $2 * 1$.

```
MySQL>Select 2 * 1;
```

Contoh 2 :

Akan dicek nilai kondisi $2 > 1$.

```
MySQL>Select 2 > 1;
```

- ◆ Ekspresi Berkondisi

Ekspresi dimanfaatkan untuk mengubah keluaran yang berbeda dengan data asalnya.

Syntax :

```

Case nilai_ekspresi
When nilai_ekspresi
Then nilai_ekspresi
[Else nilai_ekspresi]
End

```

Contoh 1 :

Akan ditampilkan kota 'Yogya' menjadi 'Jogjakarta'

```
MySQL>Select nama, Case kota when 'Yogya' Then 'Jogjakarta' End From Pegawai;
```

Contoh 2 :

Akan ditampilkan nama, nilai dan skor nilai mahasiswa.

```
MySQL>Select nama_mhs, Nilai, Case
```

```

→ When nilai >= 80 Then 'A'
→ When nilai >= 70 Then 'B'
→ When nilai >= 60 Then 'C'
→ When nilai >= 50 Then 'D'
→ Else 'E'
→ End as Skor
→ From Ujian;

```

Ekspresi Tentang Waktu

SQL92 menyediakan ekspresi DATE, TIME, dan TIMESTAMP.

Date menyatakan sebuah tanggal dengan susunan : yyyy-bb-tt, misalnya 1964-9-1.

Time menyatakan waktu, dengan format : jj:mm:dd

Timestamp menyatakan sebuah tanggal dan waktu, misalnya 1964-9-1 5:32:24

Contoh 1 :

Menampilkan nama mahasiswa dan tanggal lahir yang tanggal lahirnya diatas tanggal 1964/9/1.

```
MySQL>Select nama_mhs, tgl_lahir From Mhs Where tgl_lahir > Date '1964/9/1';
```

Contoh 2 :

Melakukan proses perhitungan data tanggal

```
Select Date '1998/7/2' – Date '1998/5/2'
```

Contoh 3 :

Melakukan proses perhitungan data waktu

```
MySQL>Select Time '23:12:00' – Time '22:10:32'
```


Fungsi Nilai

Sql menyediakan sejumlah fungsi nilai yang mempunyai ciri ‘pemanggilannya tanpa disertai tanda kurung’. Fungsi-fungsi tersebut adalah :

- Current_date : menghasilkan tanggal sekarang
- Current_time : menghasilkan waktu sekarang
- Current_timestamp : menghasilkan tanggal dan waktu sekarang

Contoh 1 :

Akan menampilkan usia masing-masing mahasiswa pada saat pernyataan ini dijalankan.

```
MySQL>Select nama_mhs, ( Current_date – tgl_lahir ) / 365 as usia From Mhs;
```

Contoh 2 :

Akan menampilkan nama mahasiswa yang usianya lebih dari atau sama dengan 20 tahun.

```
MySQL>Select nama_mhs From Mhs Where (Current_date – tgl_lahir ) >= 20 * 365;
```

Latihan !!!

- Buatlah tabel dengan nama pegawai pada databases masing-masing dan struktur tabelnya adalah sebagai berikut :

Nama Field	Type Data	Keterangan
NIP	Char(7)	Primary key, tidak boleh, unik
Nama_peg	Char(20)	Tidak boleh kosong
Alamat	Char(25)	
Tgl_lahir	Date	
Tgl_masuk	Date	
Gol	Char(1)	Dapat diisikan angka A-D : A = Adm. B = Staff C = Kabag. D = Direktur
Agama	Char(1)	Diisikan dengan angka 1-5 1 = Islam 2 = Katholik 3 = Kristen 4 = Budha 5 = Hindu default nilai adalah 1
Sex	Char(1)	Diisikan dengan karakter W untuk wanita atau P untuk pria
Didik	Char(1)	Diiskan dengan angka 1 – 5 : 1 = SD, 2 = SLTP, 3 = SLTA 4 = Diploma, 5 = Sarjana

- Isikan data kedalam tabel tersebut minimal 6 record.
- Gunakan perintah-perintah yang telah dijelaskan diatas untuk melakukan query pada tabel tersebut.

PRAKTIKUM KE – IX dan X

1. Judul Materi : Queri Antar Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat merelasikan beberapa tabel untuk penyajian informasi
3. Jumlah Pertemuan : 2 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat melakukan pengelolaan satu tabel
5. Teori Dasar :

QUERY ANTAR TABEL

Menggabungkan Tabel

Sql mempunyai kemampuan untuk menggabungkan dua atau lebih tabel untuk membentuk sebuah informasi. Sedangkan prosesnya disebut join.

Contoh :

Akan menampilkan nama mahasiswa yang mengambil mata kuliah ‘Pancasila’

Keterangan : Untuk menampilkan informasi tersebut kita harus merelasikan tabel mhs dengan krs dengan kunci nim dan tabel mkuliah dengan tabel krs dengan kunci kode_kul

```
MySQL>Select mhs.nama  
→      From mhs, mkuliah, krs
```

- Where mhs.nim = krs.nim and mkuliah.kode_kul = krs.kode_kul and
 → mkuliah.nama_kul = 'Pancasila';

Garis bawah : digunakan untuk merelasikan tabel mhs dengan krs dengan kunci nim dan tabel mkuliah dengan tabel krs dengan kunci kode_kul

Menggunakan Alias

Nama tabel yang digunakan dalam query antartabel dapat menggunakan nama alias.

Contoh :

MySQL>Select a.nama

- From mhs a, mkuliah b, krs c
 → Where a.nim = c.nim and b.kode_kul = c.kode_kul and b.nama_kul = 'Pancasila';

Operator Cross Join / Perkalian Kartesian

Operator Cross Join digunakan untuk melakukan operasi perkalian kartesian.

Contoh :

P		Q	
A	B	A	X
1	2	5	6
3	4	7	8

Dengan perintah :

MySQL>Select * From P, Q;

atau dapat ditulis

MySQL>Select * From P cross join Q;

Hasilnya adalah :

A	B	A	X
1	2	5	6
1	2	7	8
3	4	5	6
3	4	7	8

Equijoin

Adalah penggabungan antar tabel yang menggunakan operator = pada kondisi Where. (sebagai contoh perhatian penggabungan antar tabel seperti diatas)

Self-join

Adalah penggabungan antar tabel yang sama.

Contoh :

Akan ditampilkan nama mahasiswa pria dan wanita.

```
MySQL>Select x.nama, y.nama
```

```
→ From mhs x, mhs y
```

```
→ Where x.sex = 'P' and y.sex = 'W';
```

Operator Union

Digunakan untuk menggabungkan hasil dari dua buah query.

Contoh :

Pinjam

Bname	Loan#	Cname	Amount
BCA Pemuda	17	Jony	1000000
BCA Majapahit	23	Seno	2000000

Tabungan

Bname	account#	Cname	Balance
BCA Pemuda	102	Hari	400000
BCA Pemuda	204	Agus	700000

Akan menampilkan semua nama nasabah yang mempunyai tabungan, pinjaman atau keduanya

```
MySQL>(Select Cname From Pinjam)
```

```
→ Union
```

```
→ (Select Cname From Tabungan);
```

Operator Intersect

Digunakan untuk memperoleh baris-baris yang terdapat pada kedua tabel.

Contoh :

Akan menampilkan semua nama nasabah yang mempunyai pinjaman dan pinjaman

```
MySQL>(Select Cname From Pinjam)
```

```
→ Intersect
```

```
→ (Select Cname From Tabungan);
```

Operator Except / Minus

Jika terdapat tabel A dan B, operasi except akan menghasilkan semua yang ada pada tabel A tetapi tidak terdapat pada tabel B.

Contoh :

Akan menampilkan semua nama nasabah yang mempunyai simpanan tetapi tidak mempunyai pinjaman.

```
MySQL>(Select Cname From Pinjam)
```

```
→ Except
```

```
→ (Select Cname From Tabungan);
```

Operator Natural Join

Operator natural join akan melakukan operasi equijoin dengan memperlakukan nama-nama field yang sama sebagai field penghubung antar tabel.

Contoh 1 :

Apabila terdapat dua buah tabel, yaitu:

Salesman yang terdiri dari field (id_sales, nama_sales dan kota) dan

Pelanggan yang terdiri dari field (id_plg, nama_plg, kota_plg dan id_sales)

Keterangan : Tabel pelanggan mengandung id_sales yang menyatakan bahwa setiap setiap pelanggan ditangani oleh seorang salesman

Dari kedua tabel diatas akan direlasikan dengan menggunakan operator natural join

```
MySQL>Select salesman.id_sales,
```

```
→ salesman.nama_sales,
```

```
→ salesman.kota,
```

```
→ pelanggan.nama_plg
```

```
→ pelanggan.id_sales
```

```
→ pelanggan.kota_plg
```

```
→ From salesman s, pelanggan p
```

```
→ Where s.id_sales = p.id_sales;
```

Atau dapat ditulis

```
MySQL>Select salesman.id_sales,
```

```
→ salesman.nama_sales,
```

```
→ salesman.kota,
```

```
→ pelanggan.nama_plg
```

```
→ pelanggan.id_sales
```

```
→ pelanggan.kota_plg
```

```
→ From salesman natural join pelanggan;
```

Contoh 2 :

Akan menampilkan nama mahasiswa yang mengambil mata kuliah ‘Pancasila’

```
MySQL>Select a.nama
→      From mhs a, mkuliah b, krs c
→      Where a.nim = c.nim and
→          b.kode_kul = c.kode_kul and
→          b.nama_kul = ‘Pancasila’;
```

Keterangan :

Query diatas adalah relasi antara mhs – krs dan mkuliah – krs, sehingga untuk mewujudkan kedua relasi tersebut harus menggunakan dua buah natural join. Sedangkan penulisan natural join seperti itu tidak bisa di jalankan di MySql, oleh karena cara yang dapat ditempuh adalah natural join diberikan untuk relasi 2 tabel kemudian hasilnya dibuat view dan dari view inilah kita gunakan natural join untuk merelasikan dengan tabel berikutnya.

Latihan !!!

1. Buatlah sebuah tabel dengan nama krs dan struktur tabelnya adalah :

Nama Fied	Type Data	Keterangan
Kode_kul	Char(7)	tidak boleh kosong
Nim	Char(14)	tidak boleh kosong
Nilai	Char(1)	Diisikan dengan huruf : A, B, C, D ataupun E

Keterangan : tabel diatas kunci primer adalah Kode_kul dan Nim

2. Pada tabel tersebut tambahkan data-data berikut :

- ‘A21-101’, ‘A21.2001.00234’, ‘B’
- ‘A21-201’, ‘A21.2001.00214’, ‘A’
- ‘A21-201’, ‘A21.2001.00234’, ‘C’
- ‘A21-601’, ‘A21.2001.00214’, ‘B’
- ‘A21-501’, ‘A21.2001.00214’, ‘A’
- ‘A21-501’, ‘A21.2001.00234’, ‘A’
- ‘A21-101’, ‘A21.2001.00224’, ‘B’

3. Sebelum anda mempraktekan query antar tabel, maka pastikan dulu bahwa telah mempunyai :

Tabel mhs dengan struktur :

Nama Field	Type	Keterangan
Nim	Char(14)	Primary key, unik dan tidak boleh kosong
Nama	Char(25)	Tidak boleh kosong dan unik
Alamat	Char(30)	
Sex	Char(1)	Hanya dapat diisikan P (pria) atau W (wanita)
Agama	Char(1)	Hanya dapat diisikan angka 1 s.d 5 (1=islam, 2=katholik, 3=kristen, 4=hindu, 5=budha) dan apabila tidak diisikan maka nilai yang tersimpan adalah 1.
Tgl_lhr	Date	

Dan isi pada tabel tersebut minimal adalah :

'A21.2001.00234', 'Dedy Sutanta', 'Jl. A. Yani 20 Solo', 'P', '1', '1979-10-20');
 'A21.2001.00214', 'Budi Satria', 'Jl. Dago 10 Yogya', 'P', '1', '1978-11-22');
 'A21.2001.00224', 'Sulis', 'Jl. Nakula I 25 Smg', 'W', '2', '1975-08-16');

Tabel mkuliah dengan struktur :

Nama Field	Type Data	Keterangan
Kode_kul	Char(7)	Primary key, tidak boleh, unik
Nama_kul	Char(20)	Tidak boleh kosong
SKS	Decimal(2)	Tidak boleh kosong
Smt_tawar	Char(1)	Diisikan 1,2,3,4 dst
Klp_kul	Char(1)	Dapat diisikan angka 1-3 : 1 = MPK, 2 = MKK, 3 = MKB default adalah 1
Progdi	Char(1)	Dapat diisikan angka 1-2 : 1 = MI-D3, 2 = SI-S1

Dan isi tabel tersebut minimal adalah :

'A21-101', 'Pendidikan Pancasila', 2, '1', '1', '1'
 'A21-102', 'Logika Algoritma', 4, '1', '2', '1'
 'A21-103', 'Pemrograman (Basic)', 2, '1', '2', '1'
 'A21-201', 'Sistem Basis Data I', 2, '2', '2', '1'
 'A21-202', 'Pemrograman Pascal', 4, '2', '2', '1'
 'A21-203', 'Ilmu Sosial Dasar', 2, '2', '1', '1'
 'A21-501', 'Bimbingan Karier', 2, '5', '3', '1'
 'A21-601', 'Proyek Akhir', 4, '6', '3', '1'
 'A12-202', 'Pemrograman Pascal', 4, '2', '2', '2'
 'A12-203', 'Kewiraan', 2, '2', '1', '2'
 'A12-501', 'Bimbingan Karier', 2, '5', '3', '2'
 'A12-601', 'Tugas Akhir', 6, '6', '3', '2'

4. Tampilkan nama mahasiswa, nama mata kuliah dan nilai untuk setiap mahasiswa yang telah mengambil mata kuliah yang sudah ada nilainya
5. Tampilkan nama mahasiswa yang sudah mengambil mata kuliah 'Bimbingan Karier'
6. Tampilkan nama mahasiswa yang sudah mengambil mata kuliah dan nilainya adalah 'A'
7. Tampilkan nama mahasiswa yang belum mengambil mata kuliah 'Proyek Akhir'
8. Berapa jumlah sks kumulatif dari mahasiswa yang nimnya adalah 'A21.2001.00214'
9. Berapa jumlah mata kuliah dari mahasiswa yang nimnya adalah 'A21.2001.00214'
10. Tampilkan nama mahasiswa yang telah mengambil mata kuliah 'Proyek Akhir' dan nilainya adalah 'A'
11. Tampilkan daftar nama mahasiswa yang mengambil mata kuliah 'Bimbingan Karier' dan 'Proyek Akhir' (gunakan operator Union).
12. Tampilkan daftar nama mahasiswa yang sudah mengambil mata kuliah 'Bimbingan Karier' tetapi belum mengambil 'Proyek Akhir' (gunakan operator Minus).

PRAKTIKUM KE – XI

1. Judul Materi : Sub Query Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat pengelolaan tabel dengan menggunakan sub query sehingga penyajian informasi lebih mudah
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat materi sebelumnya
5. Teori Dasar :

MENGENAL SUB QUERY

Pengertian Sub Query

Merupakan query yang terletak dalam query yang lain.

Contoh :

Akan ditampilkan nama pegawai yang mempunyai gaji paling besar

```
MySQL>Select nama_peg      sub query
→      From Pegawai
→      Where gaji = (select max(gaji) From Pegawai);
```

Operator Exists

Merupakan jenis operator boolean, yang menghasilkan nilai benar (true) atau salah (false).

Operator Exists akan memberikan nilai benar (true) kalau sub query menghasilkan paling tidak sebuah baris / record.

Contoh 1 :

Akan ditampilkan daftar nama mahasiswa yang sudah melakukan krs.

```
MySQL>Select nama
→      From mhs
→      Where Exists
→      (select * From krs Where nim = mhs.nim);
```

Keterangan :

Perintah pada sub query di atas berisi pengecekan nim mahasiswa pada tabel krs

Contoh 2 :

Akan ditampilkan daftar nama mahasiswa yang tidak melakukan krs.

```
MySQL>Select nama
→      From mhs
```


→ Where Not Exists (select * From krs Where nim = mhs.nim);

Keterangan :

Perintah pada sub query di atas sama dengan contoh 1, hanya ditambahkan dengan Not sebelum Exists

Operator Any (Some)

Operator Any (some) hampir sama penggunaannya seperti Exists. Tetapi operator relasi yang digunakan biasanya selain = (sama dengan), hal tersebut disebabkan apabila operator relasi = yang digunakan, maka sebetulnya fungsi operator Any (some) sama seperti operator IN, sehingga kondisi seperti itu tidak dianjurkan karena lebih mudah pemahamannya apabila menggunakan operator IN.

Contoh :

Akan menampilkan daftar nama pegawai dan gaji yang gajinya tidak paling sedikit.

MySQL>Select nama_peg, gaji

→ From pegawai

→ Where gaji > any (select gaji from pegawai);

Operator All

Operator all digunakan untuk melakukan perbandingan dengan sub query. Kondisi dengan all menghasilkan nilai benar jika perbandingan menghasilkan benar untuk setiap nilai dalam sub query.

Contoh :

Akan menampilkan nama dan gaji pegawai yang gajinya lebih rendah daripada semua pegawai di departemen 'Akunting'.

MySQL>Select nama_peg, gaji

→ From pegawai

→ Where gaji < all (select gaji From Pegawai Where depart = 'Akunting');

PRAKTIKUM KE – XI

1. Judul Materi : Sub Query Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat membuat view agar pengelolaan tabel-tabel yang rumit dapat lebih mudah
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat materi sebelumnya
5. Teori Dasar :

MENGENAL PANDANGAN (VIEW)

Pengertian Pandangan

Pandangan (view) adalah tabel semu karena secara fisik tidak ada.

Keuntungan view adalah :

- ✓ Query yang kompleks dapat di sembunyikan sehingga informasi dapat diakses dengan cepat
- ✓ Untuk mengatur data tertentu menjadi tersembunyi bagi pemakai
- ✓ Untuk melindungi data yang bersifat rahasia

Cara menciptakan Pandangan

Untuk membuat pandangan / view digunakan perintah CREATE VIEW.

Contoh :

```
MySQL>Create View mhs_view as
→      Select nim, nama, alamat From Mhs;
```

Keterangan :

Mhs_view : nama view (pandangan)

Memperoleh Informasi pada View

Setelah suatu view dibuat, maka view tersebut dapat diakses sebagaimana tabel.

Contoh :

```
MySQL>Select * From mhs_view;
```

Membuat View dari BerepaTabel

Sebuah view juga dapat dibuat dari relasi beberapa tabel.

Contoh :

```
MySQL>Create View hasil_view as
→      Select a.nim, a.nama, a.alamat
```

- From mhs a, mkuliah b, krs c
- Where a.nim = c.nim and b.kode_kul = c.kode_kul and b.nama_kul = 'Pancasila';

Membuat View Pengelompokan

View yang di dasarkan pada pengelompokkan juga dapat dibuat.

Contoh :

```
MySQL>Create View klp_view as
```

- Select depart, count(*)
- From Pegawai Group by depart;

Memberi nama lain terhadap kolom

Judul kolom / field yang akan ditampilkan dapat diubah namanya dengan cara meletakkan nama-nama judul kolom diantara nama view dan as.

Contoh :

```
MySQL>Create View klp_view
```

- (nama_depart, jumlah) as
- Select depart, count(*) From Pegawai Group by depart;

PRAKTIKUM KE – XII

1. Judul Materi : Mengetahui Indeks
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat membuat indeks dan pengelolaan terhadap file indeks tersebut
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat materi sebelumnya
5. Teori Dasar :

MENGENAL INDEKS

Pengertian Indeks

Indeks dapat dibayangkan sebagai indeks buku, sehingga melalui indeks buku tersebut dapat dicari letak item tertentu dalam buku dengan mudah. Keberadaan indeks dalam basis data antara lain adalah untuk mempercepat pencarian data berdasarkan kolom tertentu, misalnya diberikan perintah :

```
MySQL>Select * from pegawai where nip = '12345';
```

Jika nip tidak dijadikan sebagai indeks, pencarian data akan dilakukan terhadap seluruh tabel, sama seperti kalau akan mencari sesuatu dalam buku tetapi buku tersebut tidak dilengkapi dengan indeks. Namun sekiranya indeks yang berkaitan dengan nip ada, maka sistem akan menemukannya dengan cepat.

Beberapa hal yang sangat terbantu dengan adanya indeks adalah :

1. Proses penggabungan sejumlah tabel
2. Proses dengan ORDER BY
3. Proses fungsi agregat seperti MIN dan MAX

Menciptakan Indeks

Untuk membuat file indeks digunakan perintah CREATE INDEX.

Misalnya terdapat sebuah tabel dengan perintah pembuatannya adalah sebagai berikut :

```
MySQL>Create Table Pelamar (nama char(20) not null, no_tes char(3) not null);
```

Maka kita dapat membuat indeks dari tabel tersebut, seperti :

```
MySQL>Create Index no_tes_idx on pelamar (no_tes);
```

Keterangan :

No_tes_idx adalah nama indeks yang dibuat

Pelamar adalah nama tabel yang diindeks

No_tes yang berada di dalam tanda kurung menyatakan kolom / field / atribut yang digunakan untuk mengindeks.

Menciptakan Indeks yang Unik

Untuk menciptakan indeks yang unik maka harus ditambahkan kata UNIQUE diantara create dan index.

Contoh :

```
MySQL>Create unique index no_tes_idx on pelamar (no_tes);
```

Menciptakan Indeks Untuk Beberapa Field

Untuk menciptakan indeks beberapa field, maka perlu penyebutan nama-nama field yang dimaksud sebagai berikut :

```
MySQL>Create Table nilai (no_mhs Char(5) not null,  
→      kode_mkuliah char950 not null, nilai char);
```

Selanjutnya untuk menciptakan indeks berdasarkan tabel tersebut :

```
MySQL>Create unique index nilai)idx on nilai (no_mhs, kode_mkuliah);
```

Latihan !!!

1. Indeks tabel mhs yang telah anda miliki berdasatrkan kunci field NIM.
2. Indeks tabel mkuliah yang telah anda miliki berdasatrkan kunci field KODE_KUL
3. Indeks tabel krs yang telah anda miliki berdasatrkan kunci field KODE_KUL dan NIM

PRAKTIKUM KE – XIII

1. Judul Materi : Mengatur Hak Akses
2. Tujuan Inst. Khusus : Setelah mengikuti materi perkuliahan ini mahasiswa dapat mengatur hak akses terhadap basis data sehingga basis data lebih terjamin keamanannya.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat materi sebelumnya
5. Teori Dasar :

MENGATUR HAK AKSES

Untuk Apa Hak Akses Perlu Diatur

Mengontrol basis data merupakan hal yang sangat perlu diperhatikan dalam rangka menjaga keamanan dari orang-orang yang tidak berhak. Data yang bersifat rahasia tidak dapat disembunyikan dan bahkan kemungkinan data menjadi sampah karena setiap orang bisa melakukan apa saja sesuka hati.

Cara Mengatur Hak Akses

SQL menyediakan perintah Grant untuk mengatur hak akses seseorang terhadap suatu tabel.

Bentuk perintahnya adalah :

```
MySQL>Grant hak_akses On nama_tabel_atau_view To pemakai;
```

Keterangan :

- Hak akses merupakan hak akses yang akan diberikan (misalnya Select atau insert atau gabungannya). Jika hak akses lebih dari satu, antar hak akses perlu dipisahkan dengan koma.
- Nama_tabel_atau_view menyatakan tabel atau pandangan (view) yang hak aksesnya akan diatur
- Pemakai adalah nama pemakai yang telah didaftarkan pada sistem basis data. Sejumlah pemakai dapat disebutkan dengan antar pemakai dipisah dengan koma.

Macam-macam hak akses

Hak Akses	Keterangan
Select	Hak akses yang memungkinkan operasi Select
Insert	Hak akses yang memungkinkan operasi Insert
Update	Hak akses yang memungkinkan operasi Update
Delete	Hak akses yang memungkinkan operasi Delete
Index	Hak akses yang memungkinkan operasi membuat indeks
Drop	Hak akses yang memungkinkan operasi Drop

Contoh 1 :

MySQL>Grant Select on Pegawai To Terra;

Contoh 2 :

MySQL>Grant Select, Insert on Pegawai To Terra;

Membatasi Hak Akses pada Field tertentu

SQL memungkinkan pemberian hak akses hanya terbatas pada kolom tertentu.

Contoh :

MySQL>Grant Select, Update (Nip, nama_peg, depart) on pegawai to Dewi;

Memberikan Segala Hak

Jika semua hak akan diberikan kepada seorang pemakai, maka dapat dituliskan ALL PRIVILEGES pada bagian hak akses.

Contoh :

MySQL>Grant All Privileges On Pegawai To Dewi;

PRAKTIKUM KE – XIV

1. Judul Materi : Latihan Soal
2. Tujuan Inst. Khusus : -
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan Prasyarat : Mahasiswa harus sudah dapat menguasai materi yang telah diberikan

Latihan Soal Ujian

1. Pada database anda, tambahkanlah sebuah tabel dengan nama DOSEN dan struktur tabelnya adalah :

Nama Field	Type Data	Keterangan
Kode_dos	Char(3)	Primary key, tidak boleh kosong dan unik
Nama_dos	Char(20)	Tidak boleh kosong dan unik
Alamat	Char(30)	

Create table dosen (kode_dos Char(3)primary key, nama_dos char(20), alamatchar(30));

2. Masukkan record-record berikut :

```
'101','Drs. Xyz','Jl. Baru 10'
'102','Dr. Pqrs','Jl. Lama 20'
'103','Abcd, SE. MM.','Jl. Barusan'
'104','Qwer, S.Kom.','Jl. Lamaan'
```

Jawab: Insert into dosen values ('101','Drs. Xyz','Jl. Baru 10'),('102','Dr. Pqrs','Jl. Lama 20'),('103','Abcd, SE. MM.','Jl. Barusan'),('104','Qwer, S.Kom.','Jl. Lamaan');

3. Tambahkan field berikut pada tabel krs :

Nama Field	Type Data	Keterangan
Kode_dos	Char(3)	
Kode_hari	Char(1)	Kode_hari : 1 = Senin, 2 = Selasa, 3 = Rabu, 4 = Kamis, 5 = Jumat
Kode_waktu	Char(1)	Kode_waktu : 1 = 08.00-10.00 2 = 10.00-12.00 3 = 12.00-14.00 4 = 14.00-16.00 5 = 16.00-18.00 6 = 18.00-20.00

4. Kemudian update-lah isi tabel krs menjadi :

```
'A21-101','A21.2001.00234','B','101','1','1'
'A21-201','A21.2001.00214','A','102','3','2'
'A21-201','A21.2001.00234','C','103','3','1'
'A21-601','A21.2001.00214','B','105','4','4'
'A21-501','A21.2001.00214','A','104','2','3'
'A21-501','A21.2001.00234','A','101','1','3'
'A21-101','A21.2001.00224','B','102','2','1'
```

Kode_dos Kode_hari Kode_waktu

Dari keempat tabel yang sudah ada, yaitu mhs, mkuliah, krs dan dosen tuliskan perintah MySQL untuk menjawab wuery-query berikut :

5. Tampilkan jadwal kuliah untuk mahasiswa yang mempunyai NIM = A21.2001.00214 dengan contoh bentuk tampilan adalah :

```

-----
Mata Kuliah  Hari    Jam    Dosen
-----
Bimbingan Karier  Selasa 12.00-14.00 Qwer, S.Kom.
.....
.....
-----
Select      matkul.nmmt, krs.kode_hari,krs.kode_waktu, dosen.nama_dos from krs, matkul,dosen where
krs.kdmt=matkul.kdmt and krs.kode_dos=dosen.kode_dos;
    
```

6. Tampilkan jadwal mengajar untuk seorang dosen yang mempunyai kode = 102 dengan contoh bentuk tampilan adalah :

```

-----
Mata Kuliah  SKS    Hari  Jam
-----
Pendidikan Pancasila  2        Selasa 08.00-10.00
.....
.....
-----
    
```

7. Tampilkan jumlah sks kumulatif untuk setiap mahaiswa dengan contoh tampilan :

```

-----
Nama mahasiswa    Jumlah Sks Kum.
-----
Dedi S            10
.....
.....
-----
    
```

8. Tampilkan beban sks mengajar untuk setiap dosen dengan contoh tampilan :

```

-----
Nama Dosen        Jumlah Sks Kum.
-----
Qwer, S.Kom.     10
-----
    
```

9. Tampilkan daftar nama dosen yang mengajar jam 08.00-10.00 (apabila terdapat data double, maka ditampilkan sekali)
10. Tampilkan daftar nama mahasiswa yang kuliah jam 08.00-10.00 (apabila terdapat data double, maka ditampilkan sekali)