**Basis Data – Database**

Teori --> target matkul : mhs dapat merancang sebuah basis data

Praktek --> target :

mengolah data dalam database

(rancangan sudah ada --> pengolahan data -->info)

Contoh kasus :

Membuat sistem untuk mengelola data jual beli sebuah toko objek yang anda amati dalam proses jual beli.

barang ---> kodebarang, namabarang, jenis, ukuran, stok(ini data)

pegawai

pelanggan

penjualan

data adalah kumpulan fakta dari sebuah objek

data --> proses --> informasi

database adalah tempat penyimpan data (bukan fisik) / kumpulan file2 (tabel2)

DBMS = Database management system --> software / aplikasi

contoh: mysql

Bahasa : SQL (Strucured Query Language) --> PL/SQL

Struktur dasar bahasa SQL :

1. DDL = Data Definition Language

SQL yang digunakan utk pendefinisian struktur dalam database

struktur dalam database

database, tabel, kolom, user, procedure, function,

view, trigger, cursor,....

contoh :

CREATE, ALTER, DROP

SHOW, USE, DESC

1. DML = Data Manipulation Language

SQL yang digunakan untuk melakukan perubahan data dalam database

contoh :

INSERT, UPDATE, DELETE, SELECT

1. DCL = Data Control Language

SQL yang digunakan untuk pengendalian akses database

contoh :

GRANT : pemberian hak akses

REVOKE : pencabutan hak akses

Basis data => kumpulan file/tabel

Database => tabel

Tabel => kumpulan baris dan kolom

Tabel / skema relasi

Database / workspace

Langkah masuk MySQL

* aktifkan terlebih dahulu module mysql yang ada pda panel XAMP dengan cara klik tombol start
* utk bekerja dalam mysql pastikan posisi command sesuai dengan posisi instalasi mysql C:\xampp\mysql\bin

CARA 1 :

* buka command prompt
* ganti posisi command prompt

awal : C:\Users\Lenovo>

target : C:\xampp\mysql\bin

* ganti directory bertahap, perintah CD atau CHDIR

C:\Users\Lenovo>cd..

C:\Users>cd..

C:\>cd xampp

C:\xampp>cd mysql

C:\xampp\mysql>cd bin

C:\xampp\mysql\bin>

* untuk masuk kedalam mysql, perintah dasar
* mysql -u namauser -p password -h alamathost

user tertinggi (super administrator) : root

password : tidak ada

alamat host : localhost

CARA 2 :

* buka command prompt
* ganti posisi command prompt

awal : C:\Users\Lenovo>

target : C:\xampp\mysql\bin

* ganti directory bertahap, perintah CD atau CHDIR secara langsung

C:\Users\Lenovo>cd C:\xampp\mysql\bin

CARA 3 :

* langsung klik button shell pada panel XAMPP

**DATA DEFINITION LANGUAGE**

1. Melihat database yang sudah ada

SHOW databases;

1. Masuk kedalam sebuah database

USE namadatabase;

contoh :

masuk kedalam database mysql

USE mysql;

1. Melihat tabel yang ada dalam sebuah database

SHOW tables;

1. Melihat struktur sebuah tabel

DESC namatabel;

contoh : melihat struktur tabel user pada database mysql

DESC user;

desc time\_zone;

1. Melihat isi tabel keseluruhan

SELECT \* FROM namatabel;

\* => semua /all

Kalau data dalam tabel ada, maka akan tampil berupa tabel

Kalau data tidak ada => response : empty set

1. Menampilkan hanya beberapa field/kolom dari sebuah tabel

SELECT namafield1, namafield2

FROM namatabel;

Contoh :

select host,user, password from user;

Tampilkan kolom host, user, Password , select\_priv, Insert\_priv d tabel user

1. Membuat database

CREATE database namadatabase;

catatan :

* pastikan namadatabase belum ada sebelumnya
* penamaan struktur mengikuti aturan penamaan variabel

contoh :

buat database db\_latihan1

CREATE database db\_latihan1;

1. menghapus database

=>ingat dihapus bukan database bawaan

DROP DATABASE namadatabase;

contoh :

Hapus database db\_latihan1

DROP database db\_latihan1;

**DDL => Data Definition Language**

1. Membuat database baru

2. Membuat tabel baru

3. Merubah struktur tabel (ALTER TABLE)

4. Merubah nama tabel dan menghapus tabel

5. Menghapus database

Primary Key

=> Mengidentifikasi setiap baris didalam sebuah tabel. Kolom/field yang dijadikan sbg PK harus unik.

Contoh : Tabel Mahasiswa

no\_bp

nama

alamat

tgl\_lahir

6. Membuat primary key pada sebuah tabel

Pada tabel barang :

kdbrg --> kodebarang

nmbrg --> namabarang

satuan

1. Mendefinisikan PK diawal saat pembuatan tabel.

**Struktur1 :**

CREATE TABLE namatabel

(

namafield1 tipedata(constraint),

namafield2 tipedata(constraint),

................

primary key(namafield1)

);

**Struktur2 :**

CREATE TABLE namatabel

(

namafield1 tipedata(constraint) primary key,

namafield2 tipedata(constraint),

................

);

**Contoh :**

Buatlah tabel tbrg pada database dbpenjualan dengan field :

kdbrg --> varchar(5) primary key

nmbrg --> varchar(25)

satuan --> varchar(15)

**cara 1:**

CREATE TABLE tbrg

(

kdbrg varchar(5),

nmbrg varchar(25),

satuan varchar(15),

primary key(kdbrg)

);

**cara 2:**

CREATE TABLE tbrg2

(

kdbrg varchar(5) primary key,

nmbrg varchar(25),

satuan varchar(15)

);

PK --> NO (NOT NULL)

Selain PK --> YES (NULL)

NULL != NOL (0)

1. Melakukan perubahan pada tabel yang sudah ada

(ALTER TABLE)

**Contoh :**

Buatlah sebuah tabel baru dengan nama tpelanggan, dengan field:

kdlgn --> varchar(5)

nmlgn --> varchar(25)

almlgn --> varchar(50)

kota --> varchar(25)

kdpos --> varchar(5)

notelp --> varchar(12)

**Struktur :**

ALTER TABLE namatabel

ADD primary key(namafield);

ALTER TABLE tpelanggan

ADD primary key(kdlgn);

DESC tpelanggan;

1. Mengubah sebuah field menjadi NOT NULL (harus diisi)

Contoh : ubahlah field nmlgn menjadi NOT NULL

ALTER TABLE tpelanggan

CHANGE nmlgn nmlgn varchar(25) NOT NULL;

1. Menghilangkan primary key dari sebuah tabel

**Struktur :**

ALTER TABLE namatabel

DROP primary key;

**Contoh :**

Hilangkan PK dari tabel tpelanggan

ALTER TABLE tpelanggan

DROP primary key;

DESC tpelanggan;

1. Membuat tabel dengan PK lebih dari satu (composite PK)

CREATE TABLE tbrg

(

kdbrg varchar(5),

nmbrg varchar(25),

satuan varchar(15),

primary key(kdbrg, nmbrg)

);

**DML => Data Manipulation Language**

=> Bahasa SQL yang digunakan untuk melakukan memanipulasi data.

Contoh : INSERT, UPDATE, DELETE, SELECT

1. Perintah untuk menampilkan semua nilai/record dalam sebuah tabel

SELECT \* FROM namatabel;

1. INSERT

=> Perintah yang digunakan untuk menginputkan nilai dalam sebuah tabel

1. Mengisi seluruh field/kolom dalam sebuah tabel

**Struktur:**

INSERT INTO namatabel

VALUES (nilai1, nilai2, .....);

**Catatan :**

* Pastikan banyaknya nilai yang diinputkan sesuai dengan banyaknya kolom didalam tabel.
* Untuk kolom dengan tipe primary key, pastikan nilainya adalah unik (belum pernah diinputkan).
* Nilai dengan tipe karakter(char, varchar) ditulis dengan tanda petik. Contoh : 'Meri'
* Nilai dengan tipe numerik(int, bigint) tidak perlu ditulis dengan tanda petik. Contoh : 1250
* Nilai dengan tipe NULL (boleh kosong), dituliskan dengan NULL tanpa tanda petik.

**Contoh1 :**

Inputkan data barang dengan kode B001 dan nama barang adalah Laptop Acer, satuan Unit.

INSERT INTO tbrg

VALUES('B001','Laptop Acer','Unit');

SELECT \* FROM tbrg;

**Contoh2 :**

Inputkan data barang dengan kode B001 nama barang adalah Mouse Logitech dan satuan adalah Pieces.

Duplicate entry --> karena kode barangnya sama.

**Contoh3 :**

Inputkan data barang dengan kode B002. nama barang adalah Monitor LG, tidak ada nilai satuan

INSERT INTO tbrg

VALUES ('B002','Monitor LG',NULL);

NULL --> tidak dalam tanda petik

jika ditulis 'NULL' --> bernilai tulisan NULL

1. Mengisi beberapa baris sekaligus

INSERT INTO namatabel VALUES

(nilai1, nilai2, .....),

(nilai1, nilai2, .....),

.....;

**Contoh :**

Isikan data sbb:

Kode barang B003, nama barang Flashdisk HP, satuan buah

Kode barang B004, nama barang keyboard LG, satuan pieces

Kode barang B005, nama barang Mouse ECO, satuan box

1. Mengisikan data ke beberapa kolom/field

INSERT INTO namatabel(namafield1, namafield2)

VALUES (nilaifield1, nilaifield2),

(nilaifield1, nilaifield2),

...............;

Yang diisikan hanya 2 kolom saja dan pastikan nilai PK ada, kecuali jika PK diseting auto increment.

**Contoh1 :**

Isikan hanya kode dan nama barang saja untuk barang dengan kode B007 dan nama barang adalah Headshet JBL. Kode B008 nama barang adalah CMOS Battery.

INSERT INTO tbrg(kdbrg, nmbrg)

VALUES

('B007','Headshet JBL'),

('B008','CMOS Battery');

**Contoh2 :**

Isikan kdlgn, nama pelanggan dan kota saja, dengan data sbb :

kdlgn P001, nama Raisa, Kota Jakarta

kdlgn P002, nama Rossa, kota tidak ada isinya.

1. INSERT... SET

=> bentuk lain dari insert. Disini nilai untuk setiap kolom diisikan satu persatu.

INSERT INTO namatabel

SET namafield1 = nilai1,

namafield2 = nilai2,

......;

**Contoh :**

Isikan kode barang B010, nama barang adalah speaker Logitech dan satuan adalah unit.

INSERT INTO tbrg

SET kdbrg = 'B010',

nmbrg = 'Speaker Logitech',

satuan = 'Unit';

Membuat tabel yang mengandung nilai auto increment dan nilai default

auto increment => melakukan penambahan data secara otomatis. Contoh jika kdbrg diseting auto increment, maka akan menjad 1, 2, 3, ....

default => membuat data memiliki nilai default.

Contoh misalnya tanggal diseting date default 20210922

Artinya : ketika tanggal ini tidak isi, maka secara otomatis sistem akan mengisikan data 20210922

Buatlah sebuah tabel tkaryawan dengan komposisi :

kode int auto\_increment primary key

nama\_kary varchar(30) not null

kota varchar(30) default 'Padang'

tanggal date default 20210922

1. Mengisi tabel dengan tipe data auto increment

**Contoh1 :**

Masukkan data ke tabel tkaryawan dengan data sbb;

kode null

nama Budi

kota Bukittinggi

tanggal 1 September 2021

INSERT INTO tkaryawan

VALUES(NULL,'Budi','Bukittinggi','2021-09-01');

**Contoh2 :**

kode tidak diisi

nama susi

kota bandung

tanggal 2 September 2021

INSERT INTO tkaryawan (nama\_kary, kota, tanggal)

VALUES('Susi','Bandung','2021-09-02');

1. Mengisi data dengan nilai default

**Contoh :**

Masukkan data berikut

kode NULL

nama Joko

field yang lain adalah nilai default

INSERT INTO tkaryawan

VALUES(NULL, 'Joko', default, default);

**SQL => Structure Query Language**

1. DDL => Data Definition Language

Bahasa untuk mendefinisikan struktur dalam database. Yang berhubungan dgn database, table, user, procedure, dll

1. DML --> Data Manipulation Language

Bahasa yang digunakan untuk memanipulasi data. Insert, Update, Delete, Select.

1. DCL --> Data Control Language

User/privilege

Database dbjual :

1. Tabel tbrg

2. Tabel tpelanggan

3. Tabel ttrans

2 buah tabel dari database dbjual

1. tabel tbrg => halaman 105

2. tabel tpelanggan => halaman 122

* Isi tabel tbrg dengan data yang ada pada halaman 116
* Isi tabel tpelanggan dengan data yang ada pada halaman 123

**DML**

1. **INSERT**
2. INSERT INTO namatabel VALUES

INSERT INTO tbrg VALUES

('.....','.....','.....'),

('.....','.....','.....'),

...... ;

1. INSERT ... SET
2. **UPDATE** => mengubah nilai data pada sebuah tabel

**Struktur :**

UPDATE namatabel

SET namafield = nilaibaru

[WHERE kondisi];

**Contoh :**

Ubahlah nilai satuan menjadi unit.

UPDATE tbrg

SET satuan = 'Unit';

Ubahlah nilai satuan menjadi unit dengan kode barang B001.

Tabel ?? --> tbrg

Update ?? --> satuan = unit

field ?? --> kdbrg = B001

Kondisi => pernyataan logis (benar atau salah)

UPDATE namatabel

SET namafield = nilaibaru

[WHERE kondisi];

**Operator dalam kondisi :**

1. **Operator relasi : <, <=, >, >=, <>/!=**

Contoh :

1. Jadikan satuan menjadi unit untuk barang dengan kode B005.

UPDATE tbrg

SET satuan = 'Unit'

WHERE kdbrg = 'B005';

1. Ubah nama barang disket menjadi flashdisk sandisk

UPDATE tbrg

SET nmbrg = 'Flashdisk Sandisk'

WHERE nmbrg = 'Disket';

1. Untuk satuan selain Unit, ubahlah menjadi Box

UPDATE tbrg

SET satuan = 'Box'

WHERE satuan <> 'Unit';

atau :

UPDATE tbrg

SET satuan = 'Box'

WHERE satuan != 'Unit'

atau :

UPDATE tbrg

SET satuan = 'Box'

WHERE NOT(satuan = 'Unit');

1. **Operator Logika :**

AND

OR

NOT

**Contoh :**

1. Ubahlah satuan pada tabel barang menjadi Pieces Untuk barang dengan kode B001 atau B002.

UPDATE tbrg

SET satuan = 'Pieces'

WHERE kdbrg = 'B001' OR kdbrg = 'B002';

1. **DELETE =**> Perintah untuk menghapus nilai yang ada pada sebuah tabel.

**Struktur :**

DELETE FROM namatabel

[WHERE kondisi];

**Contoh :**

Hapuslah data barang dengan kode B005

DELETE FROM tbrg

WHERE kdbrg = 'B005';

1. **TRUNCATE =**> perintah untuk mengosongkan tabel

**Struktur :**

TRUNCATE TABLE namatabel;

DELETE FROM namatabel;

**DML --> Data Manipulation Language**

SELECTION => Menampilkan semua data/beberapa field dari satu atau lebih tabel dengan kondisi tertentu.

1. Select dengan ekspresi => select ini tidak menggunakan tabel tetapi menggunakan ekspresi sederhana.

**Struktur :**

SELECT .... select ekspression;

**Contoh :**

SELECT 2;

SELECT 10 + 5;

SELECT 'Teknologi Informasi';

SELECT 2\*3, 2+3;

1. Select (query) yang berasal dari 1 tabel
   * Menampilkan isi seluruh tabel

SELECT \* FROM namatabel;

contoh : SELECT \* FROM tpelanggan;

* + Menampilkan hanya beberapa kolom/field yang dibutuhkan

**struktur :**

SELECT namafield1, namafield2, ....

FROM namatabel;

**Contoh :**

Tampilkan nama dan alamat pelanggan.

Berasal dari tabel apa? tabel tpelanggan

Field apa yang ditampilkan? nama, alamat

Kondisi/kriteria bagaimana? tidak ada kriteria

SELECT nmlgn, almlgn

FROM tpelanggan;

Tampilkan nama barang dari tabel barang

1. Menggunakan fungsi alias (AS) => digunakan untuk mengganti tampilan kolom/field

**Struktur :**

SELECT namafield AS namafieldbaru

FROM namatabel;

**Contoh**

Tampilkan nama dan alamat pelanggan.

SELECT nmlgn AS nama\_pelanggan,

almlgn AS 'alamat pelanggan'

FROM tpelanggan;

jika didalam tanda kutip, maka boleh ada spasi

jika tidak didalam tanda kutip, boleh menggunakan spasi

SELECT nmlgn nama\_pelanggan,

almlgn 'alamat pelanggan'

FROM tpelanggan;

1. Menampilkan isi tabel dengan kondisi/kriteria tertentu

**struktur :**

SELECT namafield1, namafield2, .....

FROM namatabel

[WHERE kondisi];

**Macam-macam kondisi :**

**Operator perbandingan : <, <=, >, >=, =, <> atau !=**

**Contoh :**

* Tampilkan data barang yang satuannya selain dari unit

SELECT \* FROM tbrg

WHERE satuan != 'Unit';

* Tampilkan nama pelanggan, alamat dan kota untuk pelanggan yang tinggal di kota Padang

SELECT nmlgn 'nama pelanggan',

almlgn alamat,kota

FROM tpelanggan

WHERE kota = 'Padang';

1. Menampilkan data secara unik

**Struktur :**

SELECT DISTINCT namafield from namatabel;

**Contoh :**

Tampilkan data barang dengan kode B001 dan B003.

SELECT \* FROM tbrg

WHERE kdbrg = 'B001' AND kdbrg = 'B003';

=> empty set karena tidak ada barang yang memiliki kode B001 sekaligus B003

SELECT \* FROM tbrg

WHERE kdbrg = 'B001' OR kdbrg = 'B003';

Tampilkan data barang yang satuannya unit tetapi bukan barang dengan kode B001

kondisi 1 --> satuan = 'unit'

kondisi 2 --> kdbrg <> 'B001'

SELECT \* FROM tbrg

WHERE satuan = 'Unit' AND kdbrg <> 'B001';

**DATA MANIPULATION LANGUAGE**

**INSERT, UPDATE dan DELETE, SELECT**

SELECTION => perintah untuk menampilkan semua data / beberapa data dari satu atau lebih tabel

1. **SELECT dengan ekspresi** => seleksi tidak berkaitan dengan tabel tetapi menggunakan ekspresi sederhana (berasal dari sebuah fungsi)

**contoh :**

SELECT 2;

SELECT 10+2;

SELECT 10+3, 10-3, 10\*3, 10/3;

SELECT 'Jurusan Teknologi Informasi';

SELECT SQRT(81)as akar\_kuadrat;

SELECT POW(5,4) as 'pangkat 5';

SELECT 'Teknik Komputer' FROM dual;

dual => tabel dummy (tabel yang tidak ada maknanya, tapi digunakan seolah-olah seperti tabel)

1. **Operator alias => as** => digunakan untuk pemberian nama tampilan kolom

SELECT namafield as namafieldbaru

FROM namatabel;

penamaan field baru mengikuti aturan penamaan variabel, kalau ingin menggunakan spasi, nama field baru diapit oleh tanda petik

Penulisan as --> dihilangkan

SELECT 10+3 as tambah;

bisa disederhanakan

SELECT 10+3 tambah;

1. **QUERY 1 tabel**

A. Menampilkan isi seluruh tabel (seluruh kolom)

SELECT \* FROM namatabel

**Contoh :**

SELECT \* FROM tbrg;

B. Menampilkan hanya beberapa kolom

SELECT namafield1, namafield2

FROM namatabel;

**Contoh :**

Tampilkan nama dan alamat pelanggan

SELECT nmlgn, almlgn

FROM tpelanggan;

atau

SELECT nmlgn 'nama pelanggan',

almlgn alamat

FROM tpelanggan;

C. Menampilkan data secara terurut

Ada 2 proses urut / sort :

* ascending/asc

=> pengurutan naik (A-Z) atau kecil --besar

* descending/desc

=> turun (Z-A) atau besar - kecil

**Struktur :**

SELECT namafield1,namafield2

FROM namatabel

ORDER BY namafield ASC/DESC;

**ASc redah ke tinggi secara urut**

**DESc tinggi ke rendah**

Pengurutan default --> ascending. Jadi kalau anda ingin menampilkan field terurut ascending, maka tidak perlu dituliskan asc nya.

order by nama asc

cukup ditulis

order by nama

**Contoh :**

1. Tampilkan nama barang dan satuan terurut berdasarkan nama barang

SELECT nmbrg nama\_barang, satuan

FROM tbrg

ORDER BY nmbrg;

1. Tampilkan nama pelanggan, alamat dan kota, terurut secara terbalik berdasarkan nama pelanggan

D. Menampilkan nilai field dengan membatasi jumlah baris/record yang akan ditampilkan

**Struktur :**

SELECT namafield1, namafield2

FROM namatabel

LIMIT jumlahrecord;

jumlahrecord --> banyak baris (integer)

**Struktur gabungan :**

SELECT namafield1, namafield2

FROM namatabel

ORDER BY namafield ASC/DESC

LIMIT jumlahrecord;

**Contoh:**

Tampilkan 5 nama siswa dengan IPK tertinggi (misalkan nama dan IPK berada pada tabel siswa)

SELECT nama, IPK

FROM siswa

ORDER BY IPK desc

LIMIT 5;

**Contoh :**

Tampilkan 4 data barang secara terurut berdasarkan nama

SELECT \* FROM tbrg

ORDER BY nmbrg

LIMIT 4;

Bentuk lain dari klausa limit

SELECT namafield1,namafield2

FROM namatabel

LIMIT barissetelah, jumlahrecord;

**Contoh :**

Tampilkan 3 data barang setelah baris kedua

SELECT \* FROM tbrg

LIMIT 2, 3;

E. Menampilkan data secara unik => data yang nilainya ganda/sama akan ditampilkan sekali saja

**Struktur :**

SELECT DISTINCT namafield

FROM namatabel;

**Contoh :**

Tampilkan data nilai satuan pada tabel barang

SELECT DISTINCT satuan

FROM tbrg;

Tampilkan data kota asal pelanggan

**FOREIGN KEY**

dbjual :

- tabel tbrg

- tabel tpelanggan

- tabel ttrans

FOREIGN KEY : fieldrelasi (field/kolom yang menghubungkan antar tabel yang saling berelasi)

Contoh antara tabel tbrg dengan ttrans

kdbrg adalah primary key (PK) di tabel tbrg

kdbrg adalah foreign key (FK) di tabel ttrans

penulisan FK boleh saja berbeda dengan PK, tetapi tipe data dan constraint harus sama jadi kalau kdbrg di tbrg varchar(5), maka di ttrans harus varchar(5) walaupun nama kolom kode\_barang

**Struktur PK**

cara 1 :

CREATE TABLE namatabel

(

.....,

PRIMARY KEY(namafield)

);

cara 2 :

CREATE TABLE namatabel

(

namafield tipedata not null primary key,

...

);

cara 3 :

ALTER TABLE namatabel

ADD Primary key(namafield);

INGAT :

PK dalam sebuah tabel cuma ada 1 --> bisa dari 1 kolom atau gabungan beberapa kolom (composite field)

FK dalam sebuah tabel bisa lebih dari 1

**contoh :**

tabel ttrans ada 2 FK

* kdbrg --> merelasikan dengan tabel tbrg
* kdlgn --> merelasikan dengan tabel tpelanggan

Sebuah tabel bisa punya PK dan FK

**Untuk membuat tabel yang memiliki foreign key**

**cara 1 :**

dideklarasikan sewaktu pembuatan tabel :

CREATE TABLE namatabel

(

.........,

......,

FOREIGN KEY(namafield)REFERENCES namatabel(namafield)

);

**contoh :**

CREATE TABLE ttrans

(

....,

kode\_brg varchar(5),

.....,

foreign key(kode\_brg) references tbrg(kdbrg)

);

**cara 2 :**

ALTER TABLE namatabel

ADD FOREIGN KEY(namafield)REFERENCES namatabel(namafield);

Buatlah tabel ttrans di dbjual

CREATE TABLE ttrans

(

nonota varchar(5),

tgl date,

kdlgn varchar(5),

kdbrg varchar(5),

jml int,

hrg bigint(20),

foreign key(kdlgn) references tpelanggan(kdlgn),

foreign key(kdbrg) references tbrg(kdbrg)

);

Isi tabel ttrans sesuai dengan halaman 123 jobsheet

**BACKUP dan RESTORE**

**A. backup / export :**

filedatabase ----> backupfile.sql

**Cara backup :**

- Keluar dari root, dengan perintah exit

- ketikkan perintah backup yang struktur sbb :

**mysqldump -u root namadatabase > lokasi\_dan\_namafile.sql**

**B. Restore / Import**

file.sql ----> database yang ada di mysql

**cara restore :**

Contoh :

database penampung : dbjual

file : dbpenjualanTK2A.sql

- pastikan database penampung ada

CREATE DATABASE dbjual;

- keluar dari root

EXIT

- perintah RESTORE

**mysql -u root namadatabase < lokasi\_dan\_namafile.sql**

Jadi :

**mysql -u root dbjual < E:\dbpenjualanTK2A.sql**

**Berbagai macam operator kondisi :**

**Operator relasi / perbandingan : <, >, <=, >=, =, != / <>**

**Contoh :**

1. Tampilkan seluruh data transaksi untuk nonota F001 (nonota --> tabel ttrans)

SELECT \* FROM ttrans

WHERE nonota = 'F001';

SELECT \* FROM ttrans

WHERE jml > 10;

**'...'--> tipe data string**

**tidak berkutip --> tipe data numerik**

1. Tampilkan tanggal transaksi, kode barang, harga untuk transaksi dengan harga minimal 100.000. tgl, kdbrg, hrg --> dari tabel ttrans

SELECT tgl, kdbrg, hrg

FROM ttrans

WHERE hrg >= 100000;

1. Tampilkan data pelanggan yang tidak tinggal di kota Bukittinggi.

SELECT \* FROM tpelanggan

WHERE kota <> 'Bukittinggi';

SELECT \* FROM tpelanggan

WHERE kota != 'Bukittinggi';

**Operator Logika**

**AND --> &&**

**OR --> ||**

**NOT --> !**

**Contoh:**

1. Tampilkan data transaksi yang melibatkan pelanggan dengan kode P002 yang membeli barang dengan kode B001

Kondisi 1 --> kode pelanggan P002

Kondisi 2 --> kode barang B001

SELECT \* FROM ttrans

WHERE kdlgn = 'P002' AND kdbrg = 'B001';

1. Tampilkan data transaksi untuk pelanggan P001, P002 dan P003

SELECT \* FROM ttrans

WHERE kdlgn = 'P001' OR kdlgn='P002' OR kdlgn='P003';

atau

SELECT \* FROM ttrans

WHERE kdlgn = 'P001' || kdlgn='P002' || kdlgn='P003';

1. Tampilkan barang yang dibeli oleh pelanggan dengan kode P001 yang harganya dibawah 500.000

kondisi 1 --> kode pelanggan P001

kondisi 2 --> hrg dibawah 500.000

SELECT kdbrg, kdlgn, hrg

FROM ttrans

WHERE kdlgn = 'P001' AND hrg < 500000;

**Operator Aritmatika**

**\*, /, +, -, div, mod, sqrt, pow**

**Contoh :**

1. Tampilkan tanggal, kode pelanggan, kode barang, harga, jumlah dan total. Total diperoleh dari hasil perkalian antara hrg dan jml.

SELECT tgl, kdlgn, kdbrg, hrg, jml, hrg\*jml AS total

FROM ttrans;

1. Tampilkan 5 data transaksi total tertinggi yang terdiri dari tanggal, kode pelanggan, kode barang, hrg, jml dan total.

SELECT tgl, kdlgn,kdbrg, hrg,jml, hrg\*jml total

FROM ttrans

ORDER BY total desc

LIMIT 5;

1. Tampilkan data transaksi yang totalnya diatas 500000

SELECT \*, hrg \* jml as total

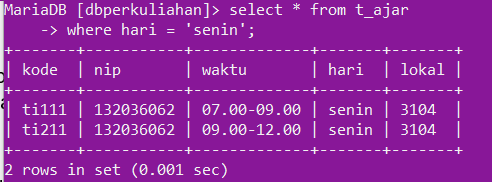
FROM ttrans

WHERE hrg\*jml >500000;

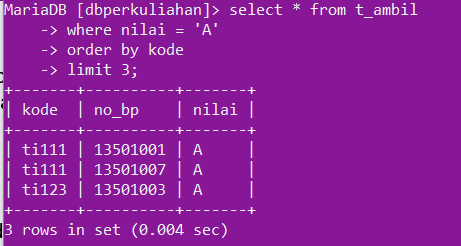
====================================================================

**DBPERKULIAHAN**

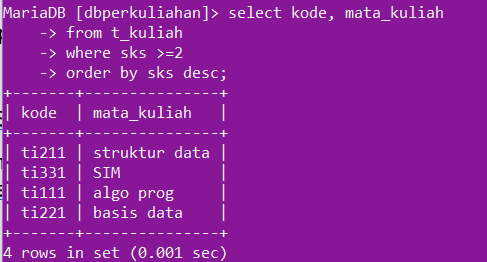
1. Tampilkan semua data pengajaran yang dilakukan pada hari senin



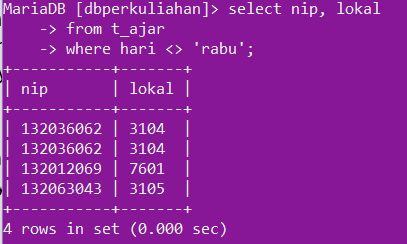
1. Tampilkan no bp mahasiswa yang mendapatkan nilai A. Tampilkan 3 data saja, urutkan berdasarkan kode mata kuliah



3. Tampilkan kode kuliah, nama mata kuliah yang jumlah SKS nya minimal 2. Urutkan berdasarkan jumlah SKS tertinggi

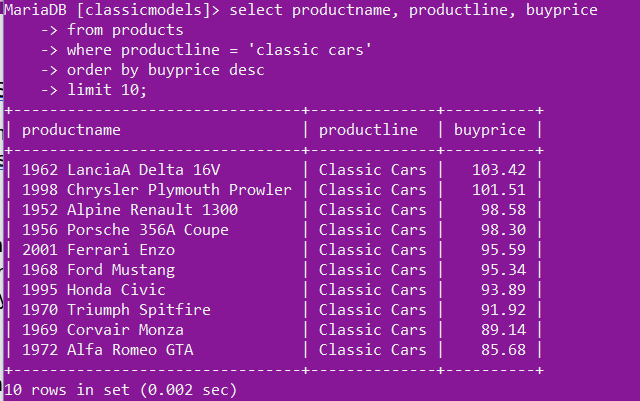


4. Tampilkan NIP dan lokal dosen yang mengajar selain hari Rabu

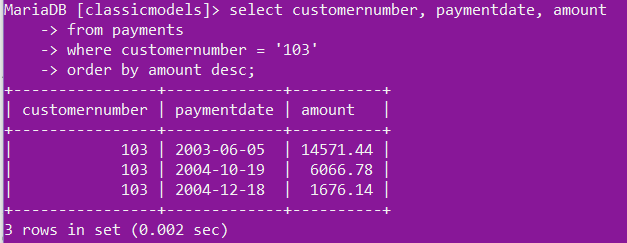


**CLASSICMODELS**

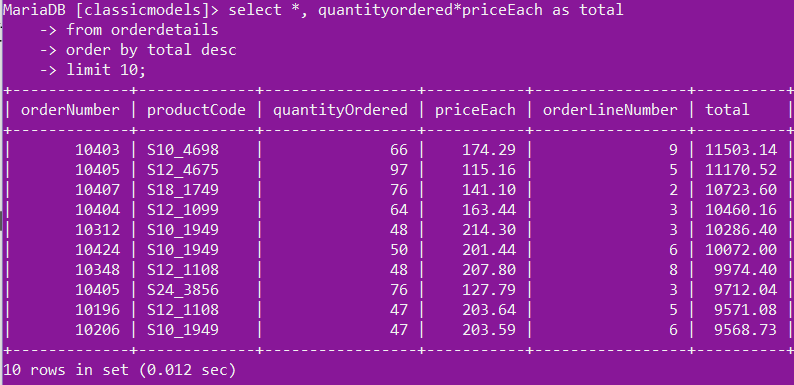
1. Tampilkan 10 data productname, productline, buyprice untuk seluruh produk berjenis(product line) classic cars. Urutkan berdasarkan harga pembelian tertinggi



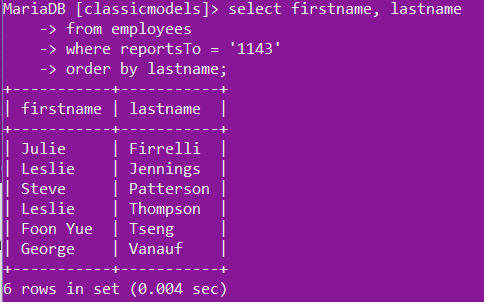
1. Tampilkan customernumber, tanggal pembayaran dan jumlah/amount dari tabel payments yang dilakukan oleh customernumber 103. urutkan berdasarkan jumlah pembayaran tertinggi



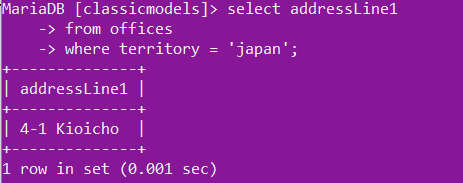
1. tampilkan 10 data detailorder dengan total tertinggi. Total diperoleh dari quantityordered \*priceeach



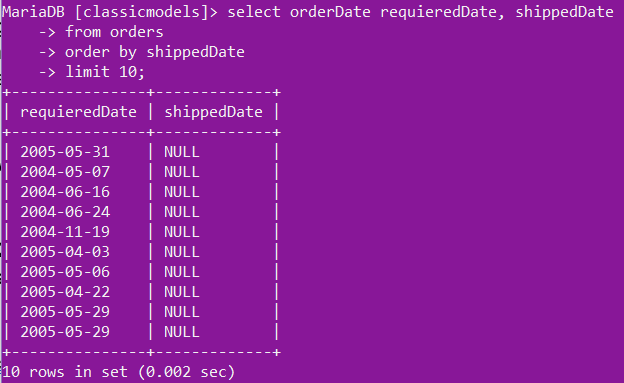
1. Siapa saja yang memberikan laporan kepada manajer dengan kode 1143? Tampilkan firstname dan lastname, urutkan berdasarkan lastname



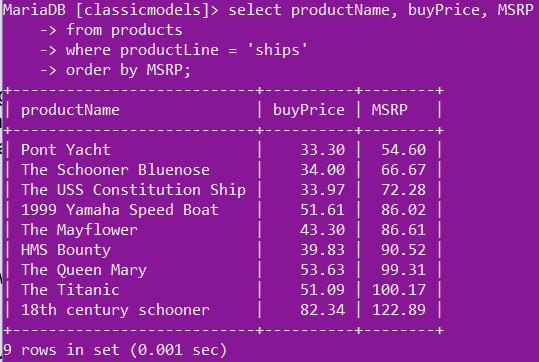
1. Tampilkan alamat kantor yang berada di daerah territory Jepang



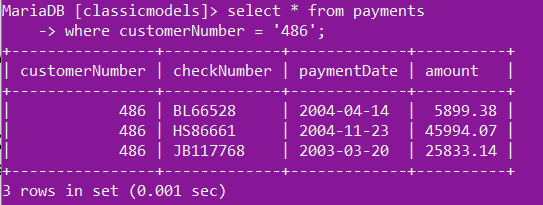
1. Tampilkan tanggal order, tanggal disetujui(requiredate) dan tanggal pengiriman yang memiliki statusnya telah dikirim (shipped), tampilkan 10 data saja dan urutkan berdasarkan tanggal pengiriman



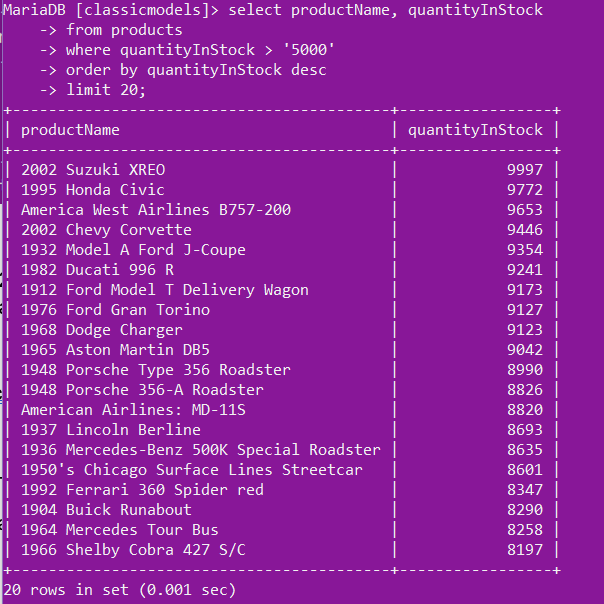
1. MSRP pada tabel products merupakan singkatan Manufacturer Suggested Retail Price, yaitu harga minimum penjualan yang dianjurkan oleh perusahaan. Tampilkan nama produk, harga pembelian, dan harga MSRP untuk produk dengan jenis ships. Urutkan berdasarkan MSRP.



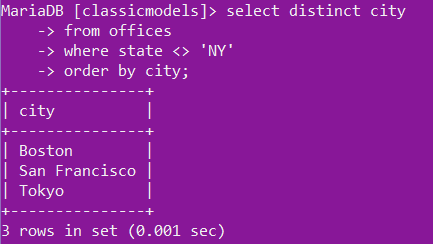
1. Tampilkan riwayat order untuk customer dengan nomor 486



1. Tampilkan 20 data nama produk, jumlah stok tersedia dari tabel produk. Tampilkan berdasarkan stok yang paling banyak yang stoknya diatas 5000

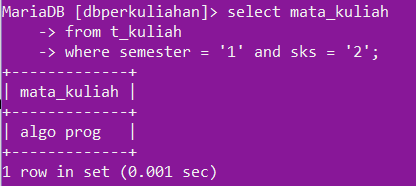


1. Tampilkan city dari tabel office yang statenya selain NY secara unik.urutkan

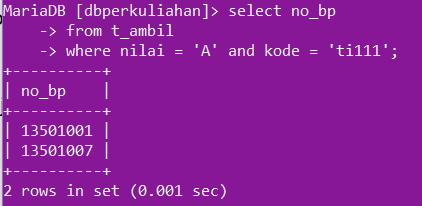


**DB PERKULIAHAN**

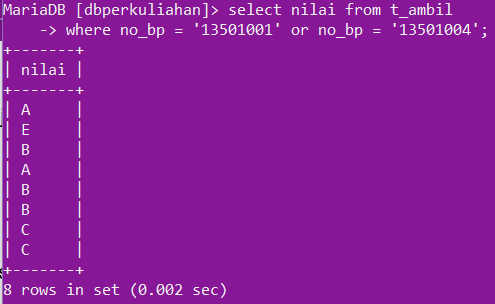
1. Tampilkan mata kuliah yang diajarkan di semester 1 yang sksnya 2



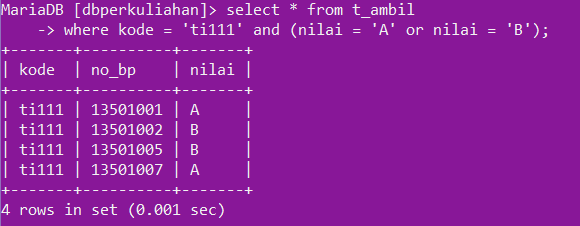
1. Tampilkan no bp mahasiswa yang mendapatkan nilai A pada kode kuliah TI111



1. Nilai apa saja yang didapatkan oleh mahasiswa dengan no bp 13501001 dan 13501004



1. Tampilkan siapa saja mahasiswa yang mendapatkan nilai A dan B pada kodekuliah Ti111



1. Tampilkan data siapa saja yang mendapatkan nilai A pada kuliah Ti111 dan Ti123

