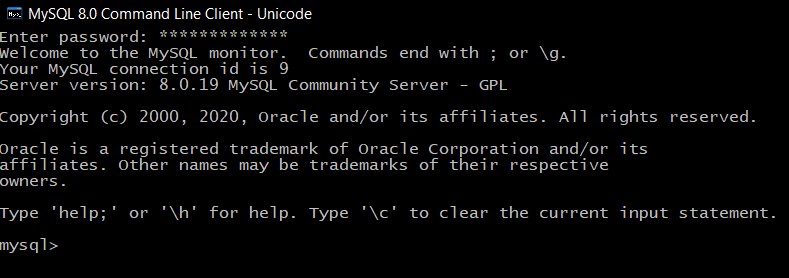
* 1. **Langkah pertama**

1. Buka mySQL Commandline Client dan masukkan password



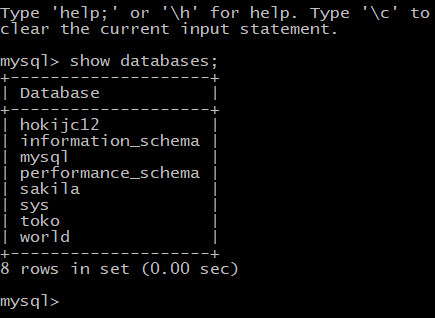
1. Selamat! Anda sudah masuk mysql command line!



Tempat untuk menulis syntax / query (dlm bahasa mysql)

* 1. **Create Database**

1. Tulis🡪 **show databases;** u/ menunjukkan database apa saja yg ada di dlm mysql



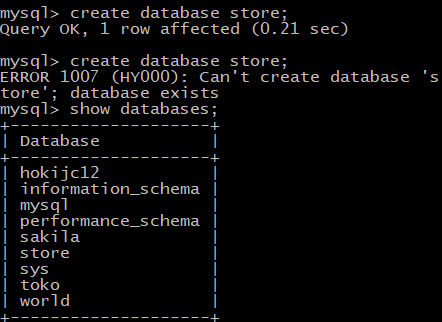
Jangan lupa titik koma (;). Query dlm btk B. ing jd lebih mudah.

Di tiap-tiap database akan ada table

Lsg muncul tempat u/ tulis query

1. Tulis🡪 **create database store;** untuk buat database baru /

Note: Bisa juga: **CREATE DATABASE IF NOT EXISTS STORE**; (Baca: kalau tak ada yg namanya *store*, akan dibuat database *store*)

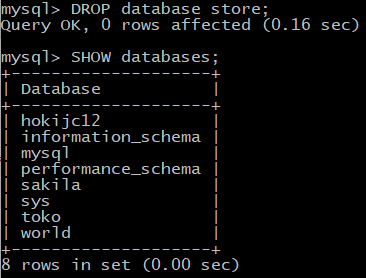


Kalau tulis: *show databases;* akan muncul database store.

Kalau ditulis ulang akan muncul error.

1. Tulis: **drop database store;** untuk hapus database yg ingin dihapus

Note: JARANG DIGUNAKAN krn bahaya kalau data dihapus



Jumlah row dlm database-nya

Kalau tulis: *show databases;* database *store* hilang

Bisa pakai **huruf kapital**, bs jg tidak

* 1. **Create Table**

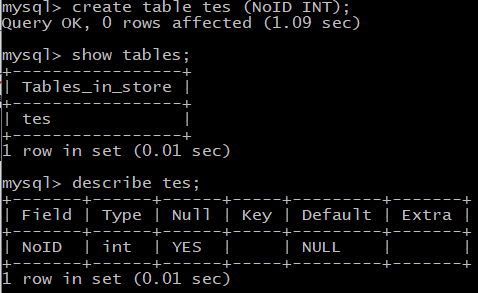
1. (Nanti akan) Ada banyak database dlm mysql**, untuk masuk ke database spesifik tersebut**, tulis: **USE** (nama database-nya)

Note: Buat lagi *database store*-nya.



1. Tulis 🡪 **CREATE TABLE TES** **(NOID INT)** u/ membuat table berisi 1 kolom

Baca: buat **TABLE,** yang namanya **tes** yang mempunyai **ID** dengan **TIPE** data **INTERGER** (angka).



Describe tes; ^^

Table minimal harus punya 1 kolum)

^^Baca:

Table: **nama** kolom-nya apa?(noID\*)

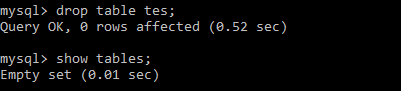
**type-nya** apa?:interger (angka)

Boleh **null** tak? Boleh\*\*

\*Tulisan hrs nyambung. Nanti di pisah dibagian front-end

\*\* Kok bisa langsung keluar *null*? (nanti belajar di bagian GUI)

1. Tulis🡪 **drop table (nama table-nya)** u/ hapus table-nya



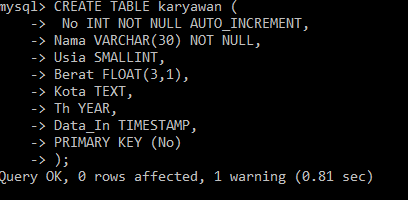
Sdh tak ada lagi *table tes*-nya

1. Di dalam table store, tulis 🡪 **create table karyawan** (isinya)

* Id ++ setiap tambah isi value
* String, max huruf 30
* Limited num (no way age = 1000^)
* Max 3 angka dibelakang koma
* Text = varchar tapi huruf lbh banyak
* Tahun/bulan/tgl tulis (sekarang)
* Nanti diajarin

Cara tulis pakai enter (biar lbh rapi)

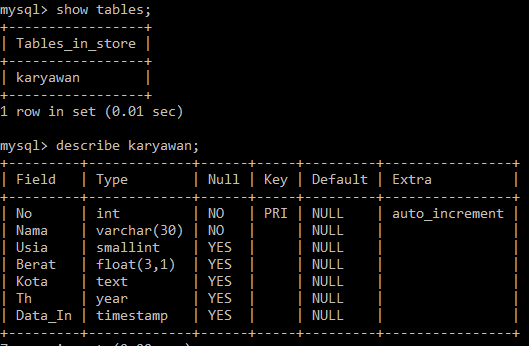
(bisa juga tulis satu baris tanpa enter)



\*Not null= tak boleh kosong

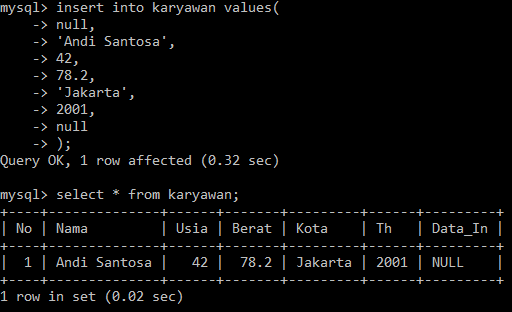
\*\* Penulisan di atas pakai

Hasil create tablenya:



* 1. **Insert Data**

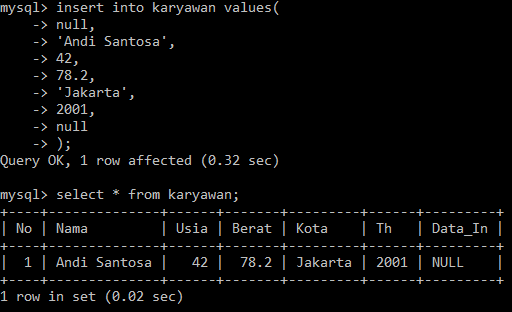
1. Tulis🡪 **INSERT INTO KARYAWAN VALUES** untuk isi data karyawan.



* Otomatis angka bertambah
* Var String harus pakai kutip
* Text juga pakai kutip
* Timestamp: Nanti diajarin

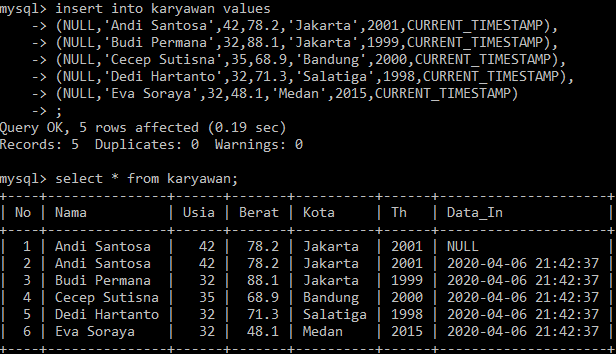
\*Urutan sangat berpengaruh (harus berurutan)

1. **Select \* from karyawan** (Baca: Lihat semua data dlm tabel karyawan)

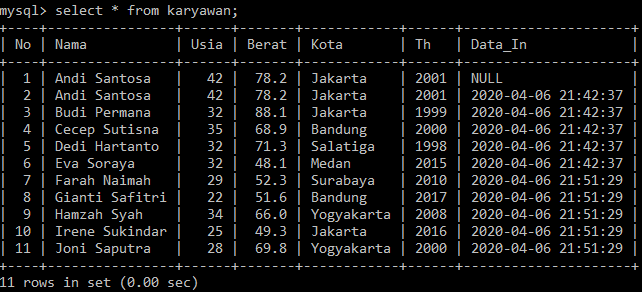


1. Tambah data kayawan-nya. Setelah ditambah akan menjadi seperti ini:

Agar tanggal masuk sesuai dg tanggal kalian masukin sekarang



.....



1. Pada tabel “karyawan” di database “store”, Insert data hanya ke kolom Nama & Usia:

**INSERT INTO karyawan (Nama, Usia) VALUES**

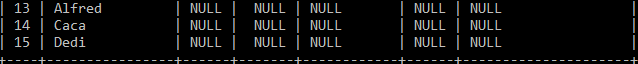
**-> (‘Budi Raharja’, 52);**



1. Insert multiple data ke kolom Nama:

**INSERT INTO karyawan (Nama) VALUES**

**-> (“Alfred”),(‘Caca’),(‘Dedi’);**

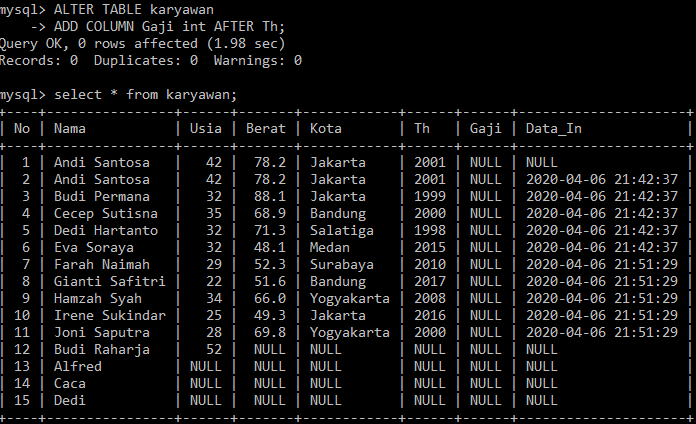


Note: Hasilnya berurutan.

* 1. **Insert Column**

1. Cara menambah kolom:

Tulis🡪**Alter table karyawan**-nya dulu



\* Cara tambah kolom gaji biasa**: ADD COLUMN GAJI INT**; (letaknya diujung)

\*\* Cara tambah kolom Gaji setelah kolom Tahun: **add column gaji int after th;**

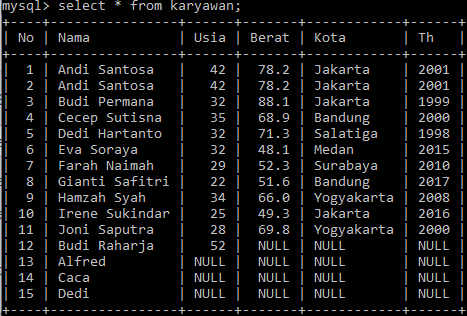
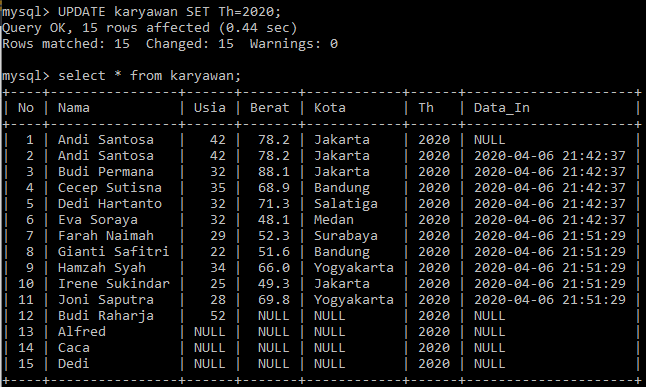
**\*\*\*** Cara **hapus**🡪 **ALTER TABLE karyawan DROP COLUMN gaji;**

**^^** Baca: tambah kolom pada table berupa kolom gaji yg tipe nya interger. Taruhnya setelah kolom th.

* 1. **Update Data**

1. **UPDATE Karyawan SET Th =2010;** (Sangat tidak disarankan karena semua tahunnya jadi berubah) pakai **where**. (no.2)

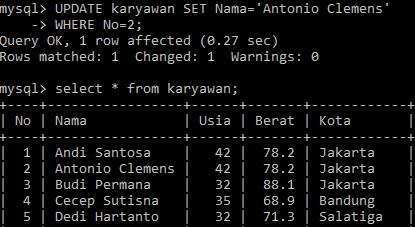
Sebelum perubahan Setelah perubahan

1. **UPDATE Karyawan SET Nama=’Antonio Clemens’**

🡪 **where No=2** (No/id ini berguna kalau dihubungkan ke node.js)

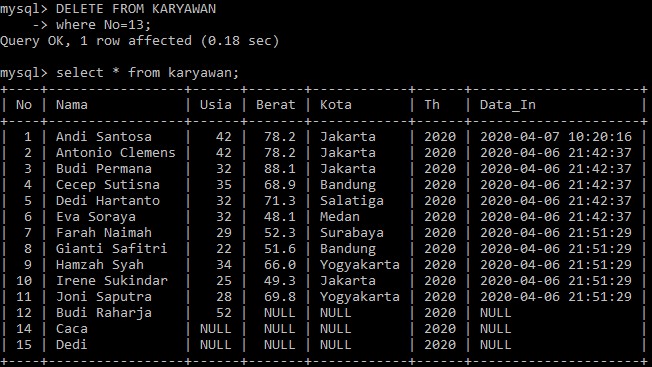
Baca: **update** (ganti) table **karyawan**, **SET** column **nama** jadi =’Antonio...’ **where**? di no.2

` 

1. Tulis🡺**DELETE from karyawan**

**Where No= 13;** (ingat, capslock tak berpengaruh)

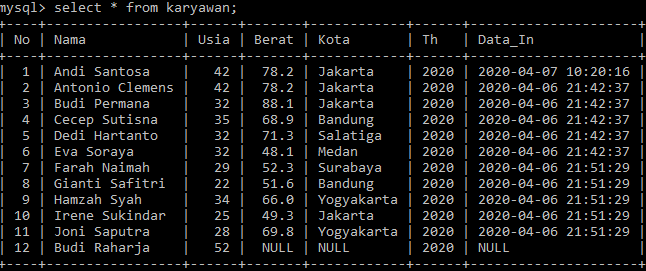
*Where* = mau yang (di)mana. Jangan langsung tulis: DELETE from karyawan krn akan hapus seluruhnya



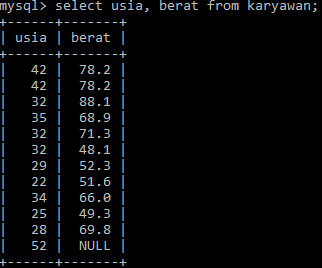
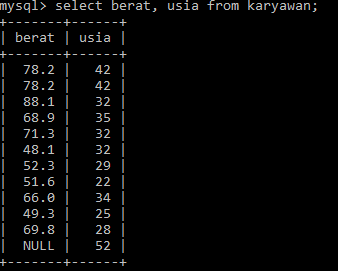
*Data no.13 sudah menghilang*

* 1. **Select Data**

1. **Select \* from karyawan;** memilih semua kolom di table karyawan (lihat 1.2 insert data)



1. **Select usia, berat from karyawan;** untuk pilih kolum spesifik dlm table karyawan

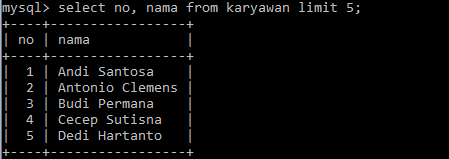
 

Note: urutan kolumn bisa kebalik (bukan permanen, tapi)

1. **Select no, nama from karyawan limit 5;** untuk lihat 5 data nama pertama

Membatasi hanya 5 data pertama.

Berguna untuk pagination.



Note: Posisi taruh limit **di paling akhir!!!**

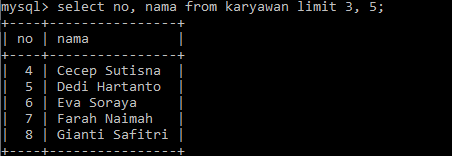
1. **Select no, nama from karyawan limit 3, 5;** untuk lihat data nama 5 data setelah data ke-3. (Mulai data ke-3, show 5 data berikutnya)

**Parameter 1:**

Skip sampai data ke berapa?

**Param 2:**

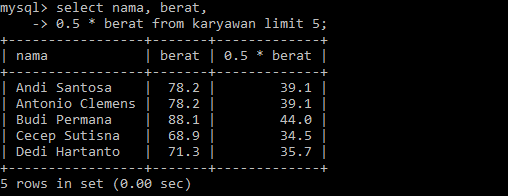
Mau unjukin berapa data-nya.



1. **Select nama, berat, 0.5\* Berat from karyawan;**

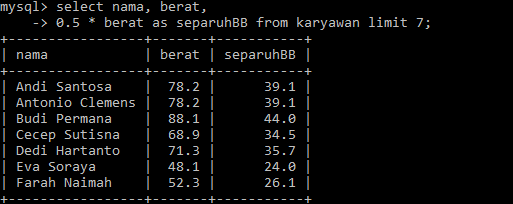
(untuk lihat setengah dari Berat badan masing-masing nama data-nya.

**Awas jgn jupa koma(,)**



Note: data tak dimasukkin ke table awal (harus di alter table karyawan-in terlebih dahulu)

Karena susah tulis 0.5 \* berat ,ditambahkan as (alias) setelah 0.5 \* berat



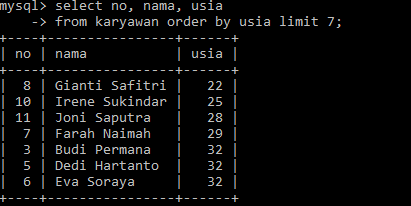
Kalau dikirim ke server, ini jadi array of object

Diubah Namanya jadi separuhBB

* 1. **Order By**

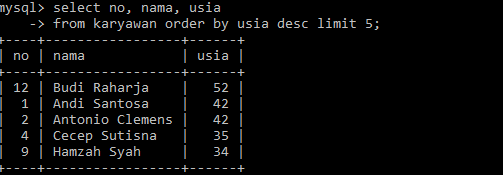
1. **Select nama, usia from karyawan ORDER BY usia;** diurutkan berdasarkan usia.

Ingat *limit* taruh paling terakhir

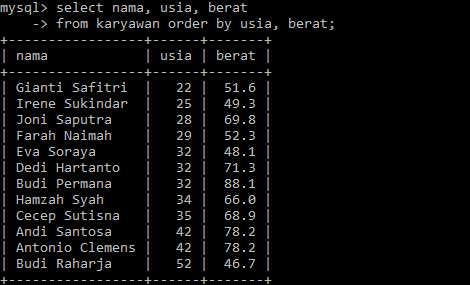


Default diurutkan secara *ascending* (dari terkecil)

1. Tulis-> **select nama, usia from karyawan ORDER BY usia desc;** untuk diurutkan berdasarkan usia paling tua terlebih dahulu.



1. Tulis-> **select nama, usia, berat from karyawan ORDER BY usia,berat;** untuk diurutkan berdasarkan usia & kalau ada usia yg sama, diurutkan (dari kecil ke besar) berdasarkan berat.



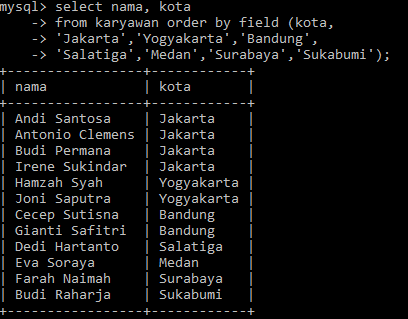
Kalau hanya berdasarkan usia, maka yang urutan nama: Budi, Dedi & Eva

Ada 2 parameter: usia, berat

Krn ada usia yg sama di urutkan ber-dsr-kan berat. Jd urutannya: E, D, B.

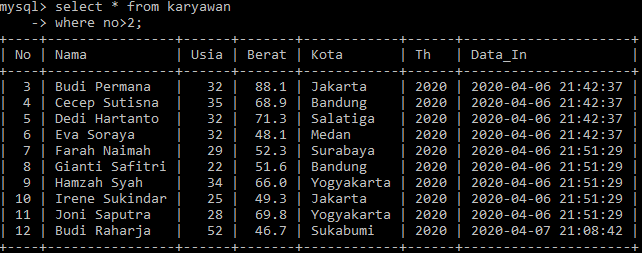
1. Order berdasarkan yg diinginkan (sedikir jarang dipakai)-> **select nama, kota, from karyawan ORDER BY FIELD**

Note: Kalau tak diurutkan semuanya, maka yg sudah di-order pindah ke paling bawah, yang belum di-order tetap pada posisi yg sama tapi pindah urutan.

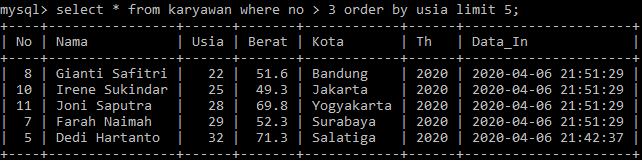


* 1. **Where**

1. **Select \* from karyawan WHERE no > 2** pilih semua tapi no lebih dari 2.



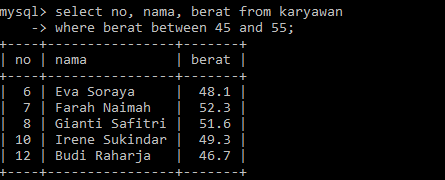
1. **Gabungan where, order by, limit**



Note: urutan pertama: where, order by, lalu limit.

1. **Select** (bebas) **from karyawan**

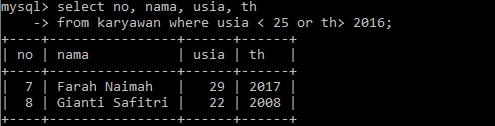
🡪**where berat BETWEEN 45 AND 55** untuk dapatkan data dengan berat antara 45 & 55



1. **Select** (bebas) **from karyawan**

🡪**where usia < 25 OR th>2016** dapatkan data dengan berat dibawah 25 / th lbh dr 2016.

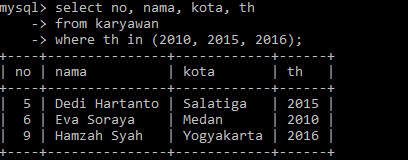
Kalau where usia < 25 saja, data yg keluar hanya 1: Gianti.



Note: ingat Js. OR kalau salah satu true, jadi sama-sama true;

1. **Select** (bebas) **from karyawan**

🡪**where th IN (2010, 2015, 2016)** dapatkan data banyak untuk 1 **kolumn**. (kalau **OR** bisa 2 param (kolom usia dan kolom tahun)

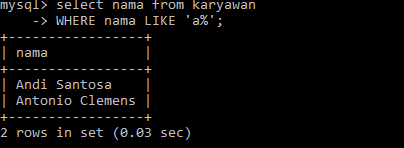


Query IN bisa lebih dr 1 param tapi hanya untuk 1 kolom saja.

* 1. **Where... like**

1. Tulis 🡪**Select nama from karyawan**

**Where nama LIKE ‘a%’ (**untuk cari namayang **DEPANNYA** huruf a**)**



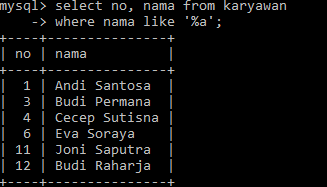
Baca **‘a%’**: huruf belakangny apa saja yang penting depannya huruf a (huruf kecil / capslock tak masalah)

Note: harus pakai kutip (baik string maupun angka)

1. Tulis 🡪**Select nama from karyawan**

**Where nama LIKE ‘%a’ (**untuk cari namayang **BELAKANGNYA** huruf a**)**

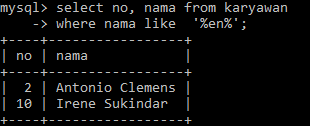
Baca **‘%a’**: depan huruf apa saja yang penting huruf BELAKANG a (huruf kecil / capslock tak masalah)



1. Tulis 🡪**Select nama from karyawan**

**Where nama LIKE ‘%e%’ (**untuk cari namayang **ADA** huruf a**)**

Baca **‘%en%’**: peduli amat depannya sama belakangnya huruf apa, yang penting ada huruf e.

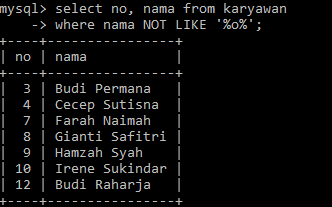


Note: Sering digunakan di *search* bagian front-end

1. Bisa juga tulis 🡪**Select nama from karyawan**

**Where nama NOT LIKE ‘%O%’ ;(**untuk cari namayang **TIDAK ADA** huruf a**)**

Baca **‘%O%’**: cari dlm kolom nama yang taka da huruf O-nya.



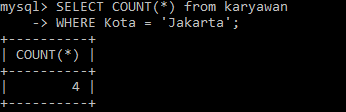
* 1. **Count, Min, Max**

1. Tulis 🡪**Select count(\*) from karyawan where kota = ‘Jakarta’;**

Posisi Count setelah SELECT

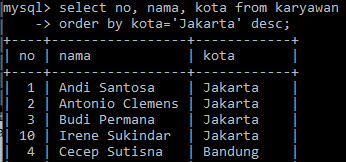
Tulis count(\*), bukan

Count (\*) (pakai spasi)

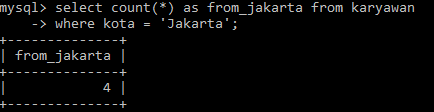


Note: Kalau count biasanya semuanya (\*)

^^ Baca: cari karyawan: yang tinggal di Jakarta **berapa** orang? (ini masih sedikit, nanti bisa jutaan)

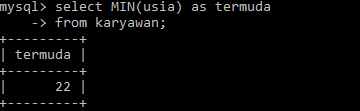
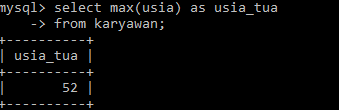


Karena jelek kalau judulnya count(\*), diubah jadi 🡪 AS from\_jakarta (pelajaran 1.7 no.5)



1. Tulis 🡪**Select MIN(usia)** atau **MAX(usia) from karyawan;**

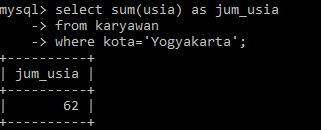
Untuk cari usia termuda untuk cari usia tertua

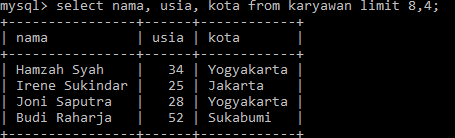
Note: tulisan **min(usia)/max(usia)** jangan dipisah

* 1. **Sum & Avg**

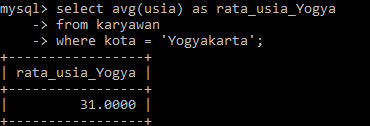
1. Tulis 🡪**Select SUM(usia) as jum\_usia** (langsung digabung dengan as) **from karyawan where kota = ‘Yogya’;** untuk tahu jumlah semua usia karyawan yang berasal dari Yogyakarta



Kalau dipastikan, jumlahnya sama

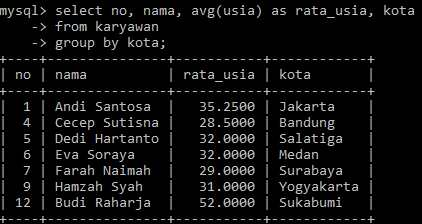


1. Tulis 🡪**Select AVG(usia) as rata\_usia\_Yogya from karyawan** untuk tahu jumlah semua usia karyawan yang berasal dari Yogyakarta



* 1. **Group By**

1. Tulis 🡪**Select AVG(usia) from karyawan group by kota;** untuk tahu rata-rata usia karyawan dari tiap-tiap kota.



Biasa tulis:

*select usia, kota from karyawan*

Bisa juga tulis:

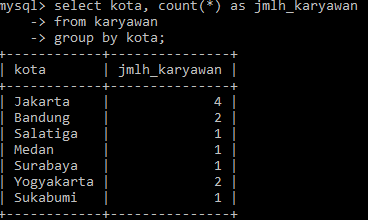
*Select nama,* ***avg(usia)****, kota from karyawan* (Bisa diselip)

**Note: Count, Min, max, sum, avg** (yang barusan aja dipelajarin) bisa dikelompokkan rata-ratanya berdasarkan tempat.

\*Real life situation: cari rata-rata pendapatan di kota A / B pakai query ini.

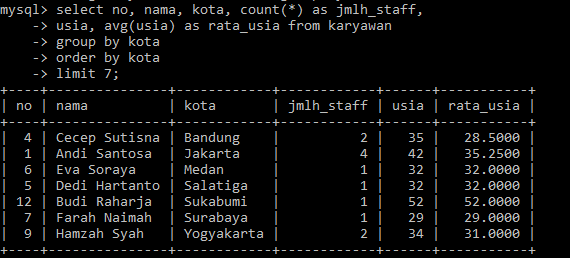
1. Tulis 🡪**Select count(\*) as jmlh\_karyawan from karyawan**

**group by kota;** untuk tahu jumlah karyawan di tiap-tiap kota



Note: Kalau count(\*) sendiri bingung jadi aku tambahkan data kota disampinnya.

1. **Gabungan Group By, Order By, Limit (Versi Nyusahin)**



Urutan:

Select (bebas)

Count(\*)

From karyawan

Where

Group by...

having

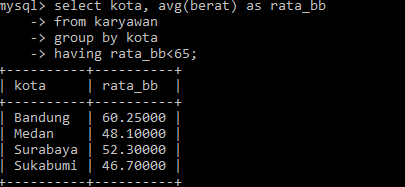
Order by...

Limit ...

* 1. **Having**

Error krn jmlh\_staff tdk tertera di table aslinya (msh ingat isi create table karyawan?)



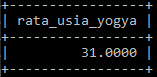
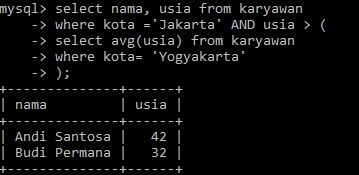


**Note:** sama seperti where, bedanya kalau where tak bisa pakai nama alias. Kalau patokannya alias harus pakai having.

**\*** Gak semua harus count(\*), min, max, dll dipakai.

* 1. **Subquery (Query dalam Query)**

Yg dlm kurung dikerjakan terlebih dalhulu)

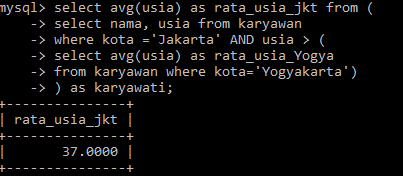


**Baca:** Tampilkan nama & usia karyawan di Jakarta, yang usianya > rata-rata usia karyawan di Yogyakarta.

**\*Query** hanya return yg rata-rata usia di Jakarta.

Yang lebih nyusahin lagi (buat rata-rata dari hasil dari atas)

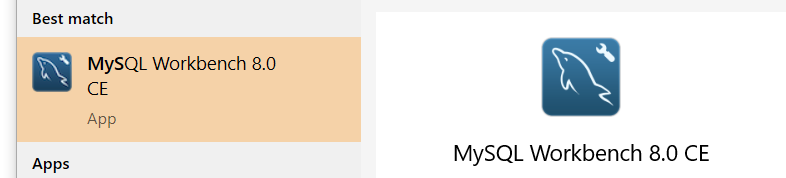
Kurang ngerti kenapa harus pakai *as karyawati* (tapi kalau tak pakai bakal keluar error hrs pakai alias)



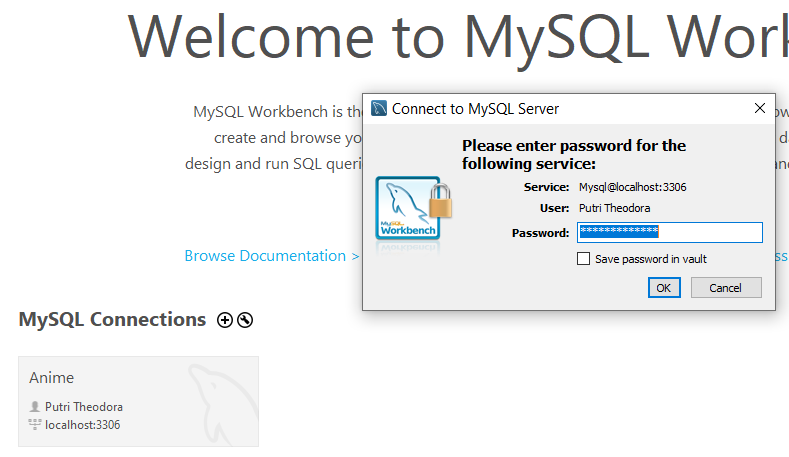
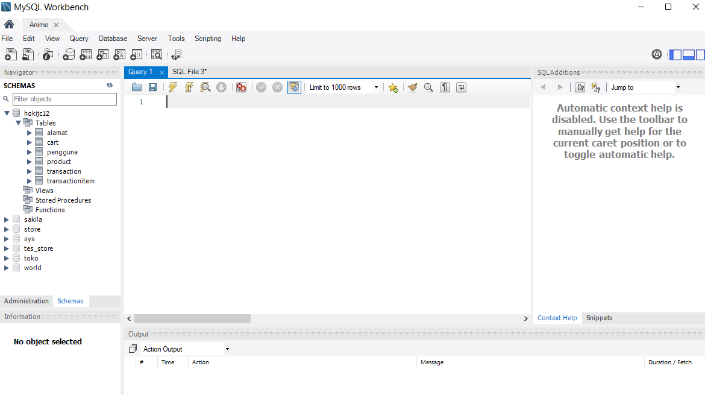
* 1. **GUI SQL**

Ambil database kak Dino dari github dan sambungkan ke mysql

1. **Buka MySQL**



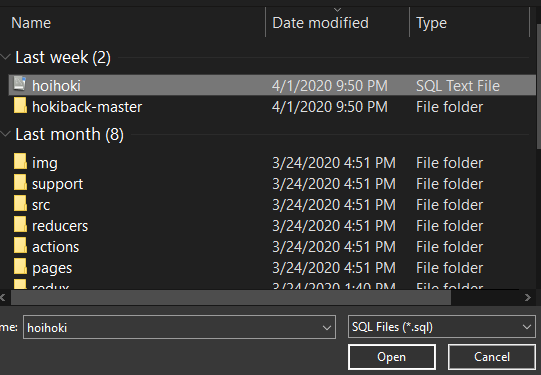
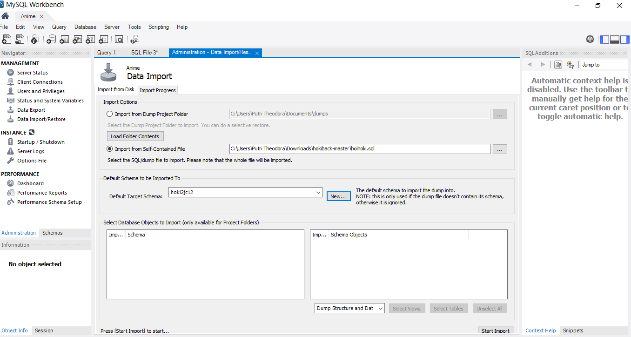
1. **Masuk ke MySQL Connections-nya Isi password, & tunggu sampai workspace keluar.**



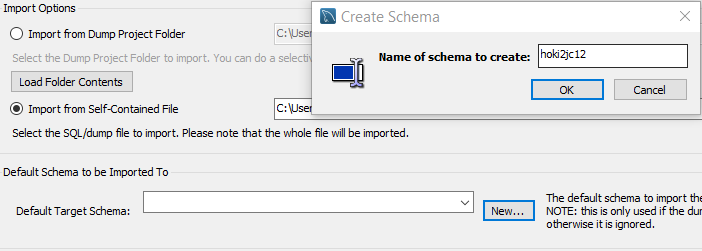
Di klik

Di isi

1. Di bagian Navigator bar pilih Administration 🡪 data, import/restore
2. Di sampping navigator ->title import from disk ->pilih yang import from self-contained File
3. Pilih data yg mau diimport (dlm btk SQL.files)🡪 Open
4. Di bawah import from self-contained File, buat nama baru.
5. Start import (posisi paling bawah kanan., kalau tak ada, di bagian kanan atas, klik tengahnya.



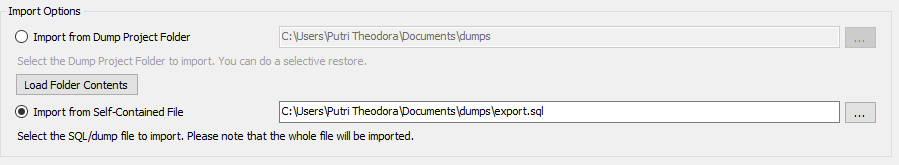
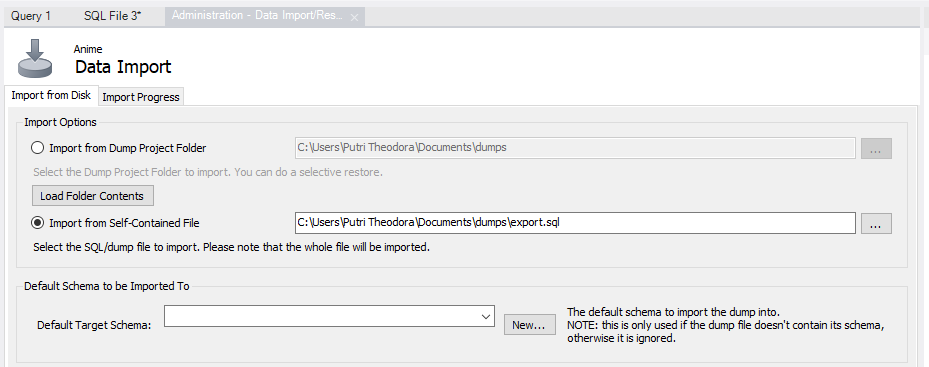
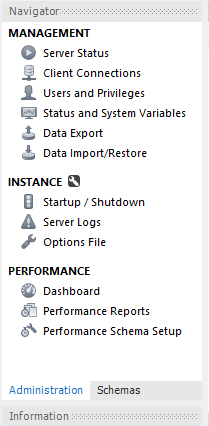
5



6

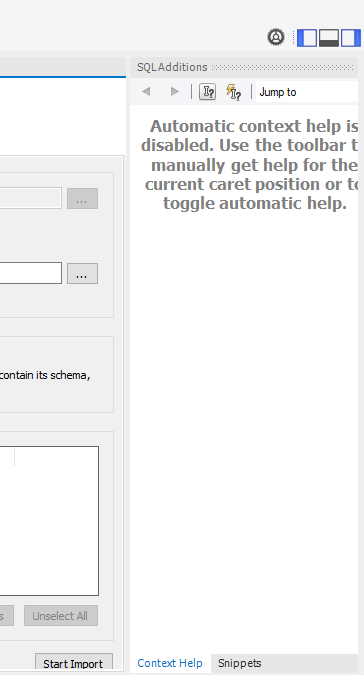
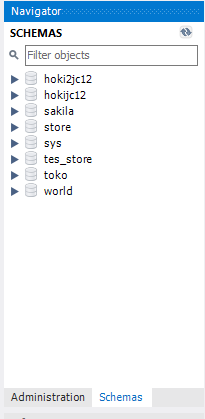
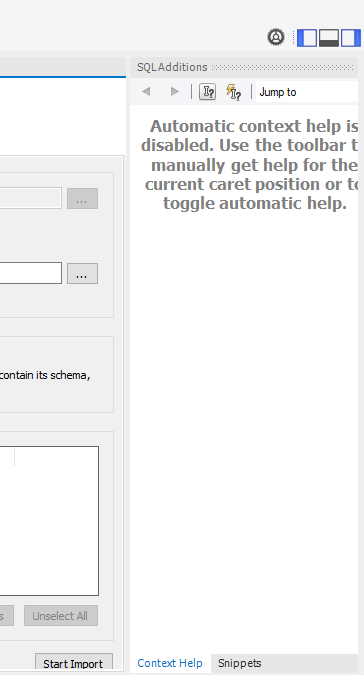
Inserted written name here

7



3

4



Di refresh dulu baru muncul

8

1. Balik ke bagian Navigator, refresh, akan muncul hoki2jc12 tadi.