

# SQL

## SQL Fundamentals



# Outline



Konsep  
Database



Pengenalan  
PostgreSQL



Konsep  
SQL



Perintah  
Dasar SQL



# Konsep Database

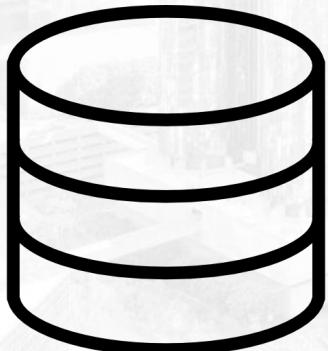
Apa itu Database? Kenapa diperlukan?

# Apa itu Database?

Sebelum database.. Apa sih **data** itu?

**Data adalah informasi.** Segala bentuk fakta yang berkaitan dengan objek tertentu bisa disebut sebagai data. Misalnya, **nama** kamu, **umur** kamu, **tinggi badan** kamu, itu semua **DATA!**

Jadi, kalau **database**?



**“A collection of data that is organized systematically so that it can be easily accessed, managed and updated.”**

# Kalau Database Management System (DBMS)?

*Software atau perangkat lunak yang dirancang untuk membuat dan mengelola database serta menjalankan operasi-operasi pada database.*



Pada umumnya, **DBMS juga biasa disebut sebagai Database saja.**  
Jadi, jangan bingung!

# Dua Tipe Umum DBMS:

- **Relational DBMS (RDBMS)**

Tipe DBMS yang menyimpan data dalam bentuk **tabular** dan diketahui **hubungan** antar tabel satu dengan tabel lainnya.

- **NoSQL DBMS**

Disebut juga NonRelational DBMS, merupakan tipe DBMS yang tidak menyimpan data dalam bentuk tabular. Biasanya berguna untuk menyimpan unstructured data.

Sesuai namanya, NoSQL **tidak bisa** menggunakan Bahasa SQL

# Contoh DBMS

RDBMS:



yang akan kita gunakan selama module

NoSQL:



# Terminologi dalam RDBMS

- **Table**

Sekumpulan data yang disimpan secara terstruktur dalam baris dan kolom

- **Record / Tuple / Row**

Single entry (satu baris data) pada suatu tabel

- **Field / Attribute / Column**

Informasi spesifik yang dimiliki oleh masing-masing record dalam tabel

- **Query**

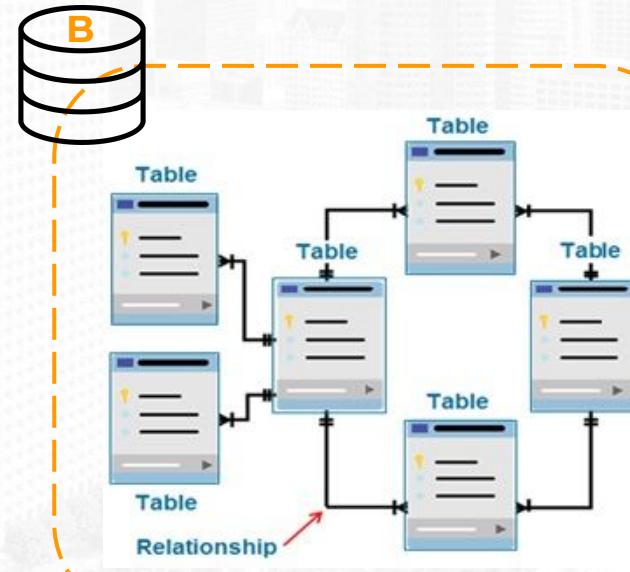
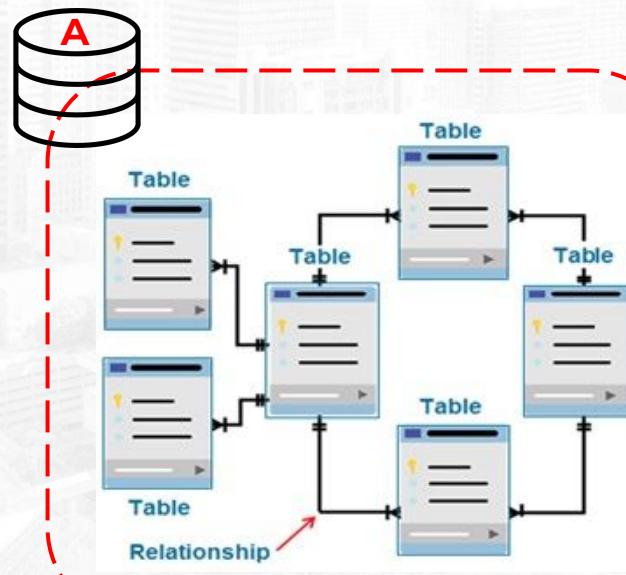
Suatu permintaan/request untuk bagian data tertentu dalam database

# Skema RDBMS

Di dalam satu server, bisa terdapat beberapa database.

Di dalam satu database, bisa terdapat beberapa tabel.

**SERVER**



# Skema Tabel

					Kolom		
Baris	order_id	customer_id	merchant_name	kuantitas	harga	tanggal_pembelian	bayar_cash
	1	1	KFC	1	30,000	6/10/2020	1
	2	2	McDonald's	3	100,000	6/11/2020	0
	3	2	Pizza Hut Box	200	2,000,000	6/11/2020	1

integer      integer      varchar      integer      integer      date      boolean

Setiap nilai pada satu kolom memiliki tipe data yang sama.

**FOREIGN KEY**, merupakan **primary key** dari tabel lain.

**PRIMARY KEY**, kolom yang digunakan sebagai **identifier** baris pada suatu tabel.

# Skema RDBMS - RANGKUMAN

- **Database Server** dapat menyimpan beberapa **Database**
- Suatu **Database** dapat memiliki beberapa **Tabel**
- Suatu **Tabel** terdiri dari susunan **Baris (Records)** dan **Kolom (Fields)**
- Suatu **Tabel** umumnya memiliki satu kolom **identifier** unik
- Kolom **Identifier** pada suatu tabel disebut sebagai **Primary Key**

# Kenapa sih pakai Database?

*Coba bayangkan sebentar...*

Sering pakai ojek online???

Kira-kira ...

Berapa banyak data yang dimiliki oleh perusahaan ojek online favorit kamu?



# Contohnya, GOJEK

## Perusahaan GOJEK



Bagaimana kalau semua data ini disimpan di EXCEL? Google Spreadsheet?

# Excel? NOPE



**SAY NO  
TO EXCEL !!!**

- 1 tabel terbatas 1 juta baris,
- Pengolahan data yang lambat untuk data besar,
- Sulit untuk digunakan secara bersamaan dalam satu perusahaan.

# Database? Yes, Please.



## Say **HELLO** to Database !

**Menyimpan seluruh data secara sistematisk.**

- Penyimpanan data dalam jumlah besar dan bisa diakses oleh semua user secara bersamaan,
- Format data yang terstandarisasi dan konsisten,
- Cepat dan mudah untuk memanipulasi data (menambah data baru, delete, edit dan lainnya).



# Pengenalan PostgreSQL

Apa itu PostgreSQL? Bagaimana mengaksesnya?

# Apa itu PostgreSQL?



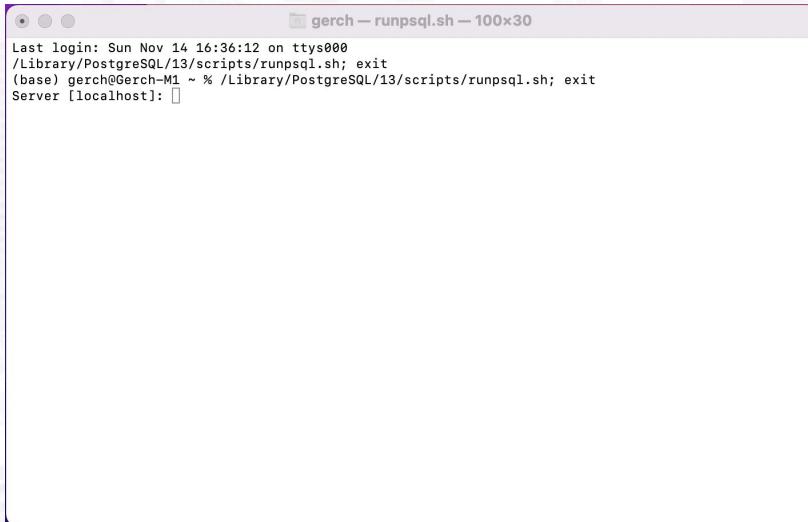
**PostgreSQL** merupakan salah satu **Relational Database Management System (RDBMS)** yang populer dipakai oleh berbagai macam perusahaan/enterprise untuk kelancaran bisnis.

**PostgreSQL** bersifat *open source* yang membuat RDBMS ini juga populer di kalangan developer ataupun pelajar (termasuk kita!).

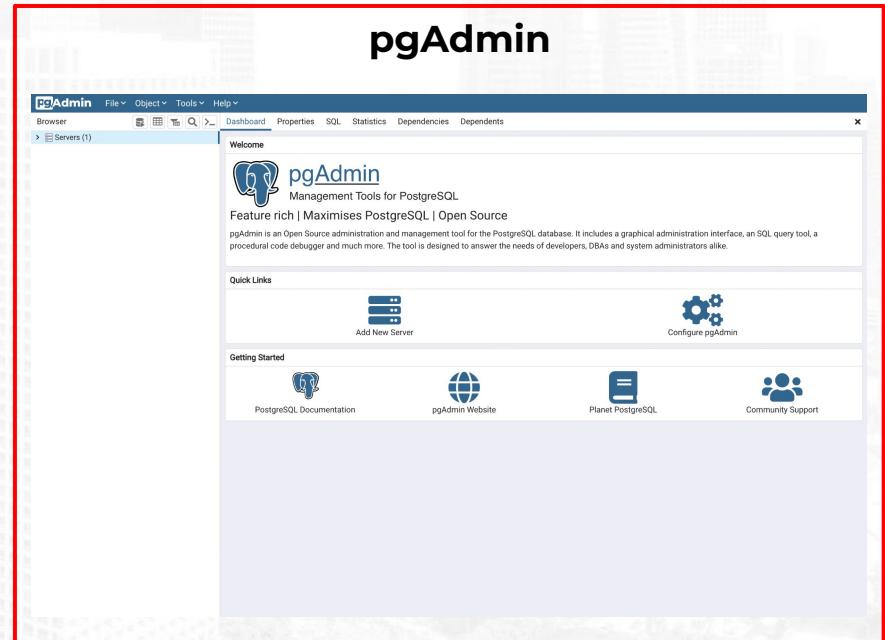
# Mengakses PostgreSQL

Selama modul kita akan menggunakan pgAdmin

## SQL Shell (psql)

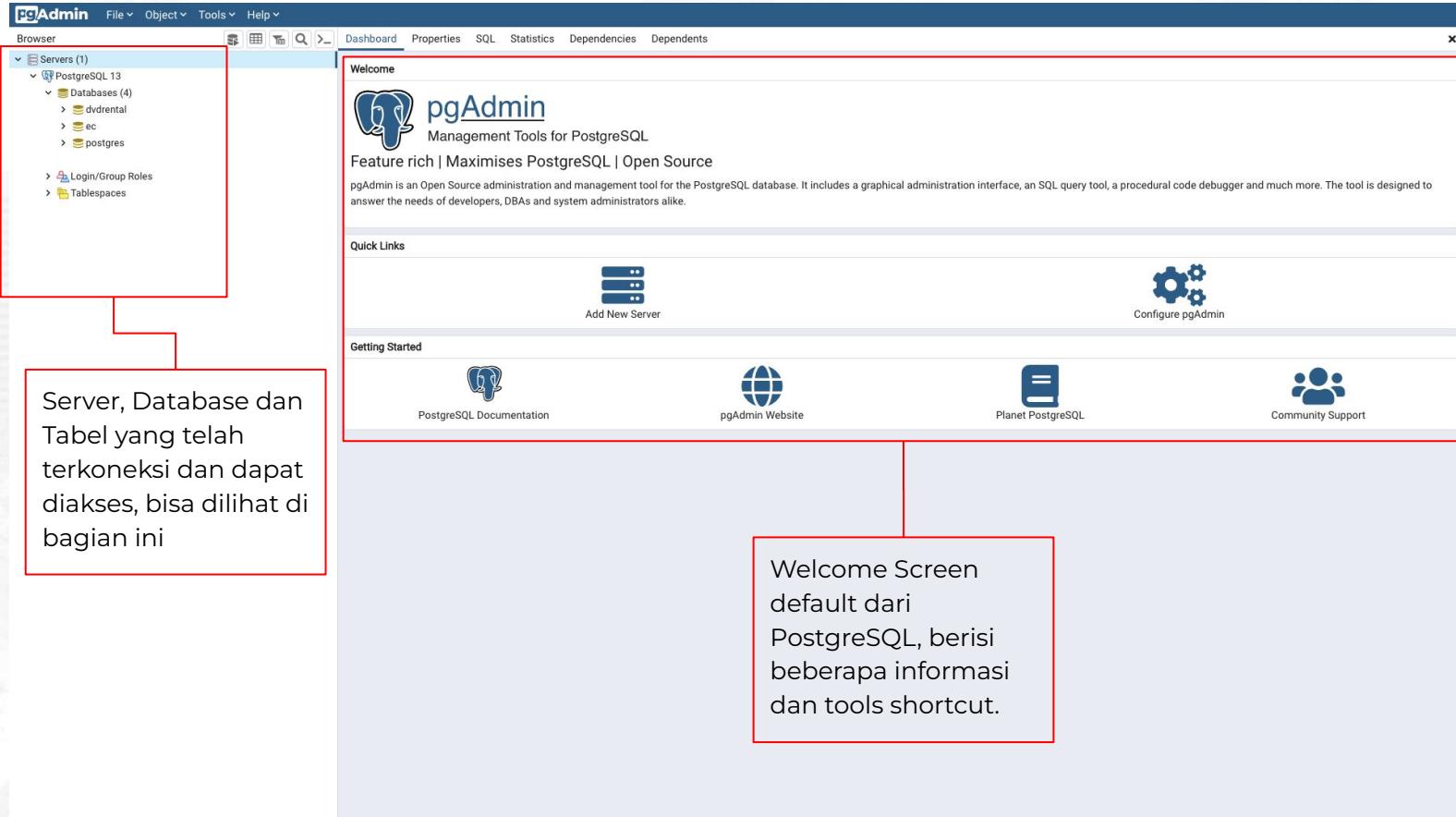


```
gerch - runpgsql.sh - 100x30
Last login: Sun Nov 14 16:36:12 on ttys000
/ Library/PostgreSQL/13/scripts/runpgsql.sh; exit
(base) gerch@Gerch-M1 ~ % / Library/PostgreSQL/13/scripts/runpgsql.sh; exit
Server [localhost]:
```



pgAdmin merupakan suatu **management tool** untuk PostgreSQL yang dapat dijalankan dalam bentuk Web Application maupun Desktop Application.

# pgAdmin - Welcome Screen



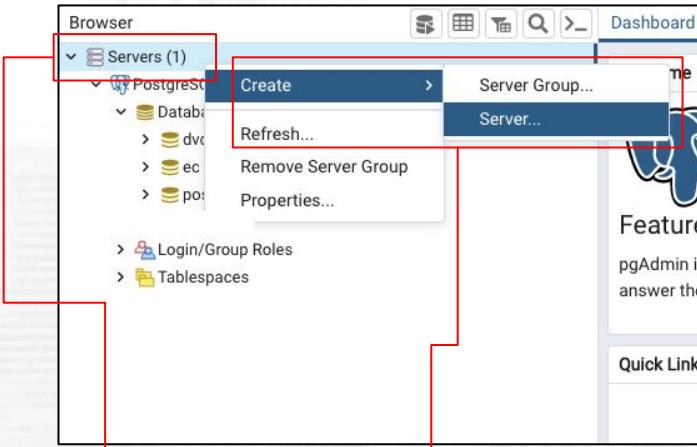
The screenshot shows the pgAdmin interface with a red box highlighting the left sidebar under the heading "Servers (1)". Inside this box, another red box highlights the "PostgreSQL 13" entry, which contains four database entries: "dvrental", "ec", and "postgres". A callout box points from this area to the text: "Server, Database dan Tabel yang telah terkoneksi dan dapat diakses, bisa dilihat di bagian ini".

The main content area is also highlighted with a large red box. It features the pgAdmin logo and the text "Management Tools for PostgreSQL". Below this is a "Feature rich | Maximises PostgreSQL | Open Source" section. A detailed description follows: "pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike." A "Quick Links" section contains links to "Add New Server" (with a server icon) and "Configure pgAdmin" (with a gear icon). The "Getting Started" section includes links to "PostgreSQL Documentation" (with a PostgreSQL icon), "pgAdmin Website" (with a globe icon), "Planet PostgreSQL" (with a document icon), and "Community Support" (with a people icon).

Server, Database dan  
Tabel yang telah  
terkoneksi dan dapat  
diakses, bisa dilihat di  
bagian ini

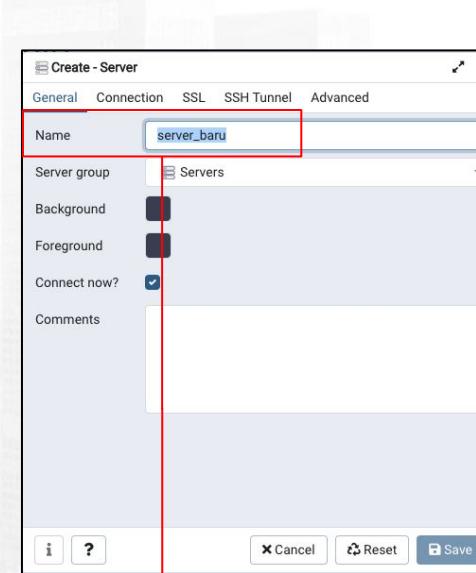
Welcome Screen  
default dari  
PostgreSQL, berisi  
beberapa informasi  
dan tools shortcut.

# pgAdmin - Membuat Koneksi Server Baru

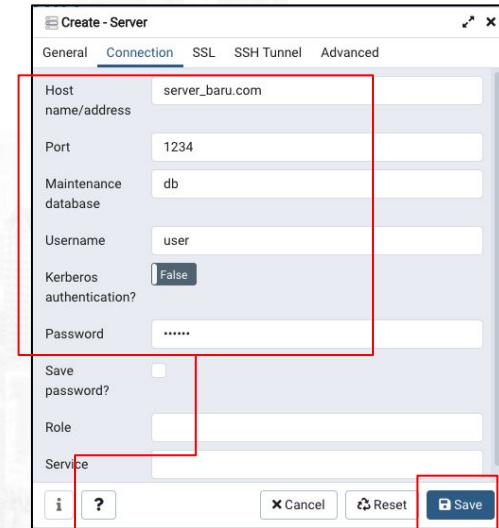


1. Klik Kanan di Servers

2. Pilih Create > Server



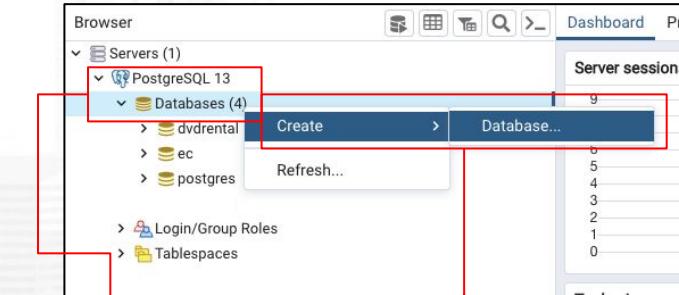
3. Pada Window yang muncul, masukkan nama server yang diinginkan (custom) pada tab General.



4. Pada tab Connection, masukkan credentials lainnya: alamat database, port, nama database yang dituju dan password.

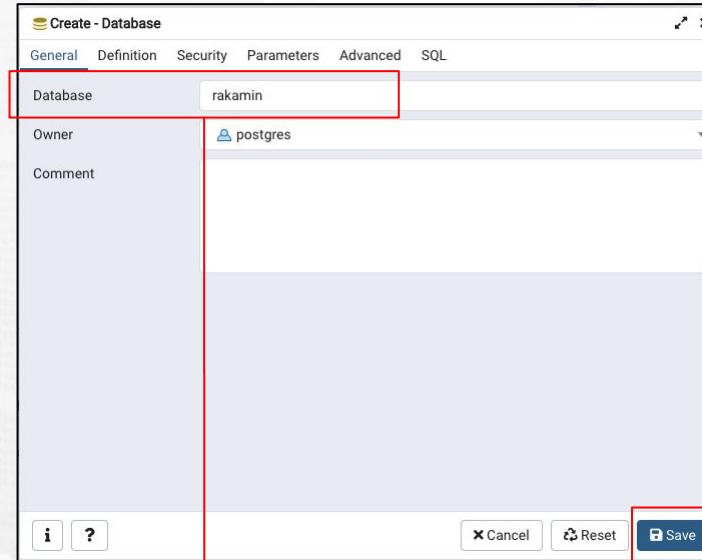
5. Klik Save

# pgAdmin - Membuat Database Baru



1. Buka dropdown di Server yang diinginkan, lalu klik kanan di Databases

2. Pilih Create > Database

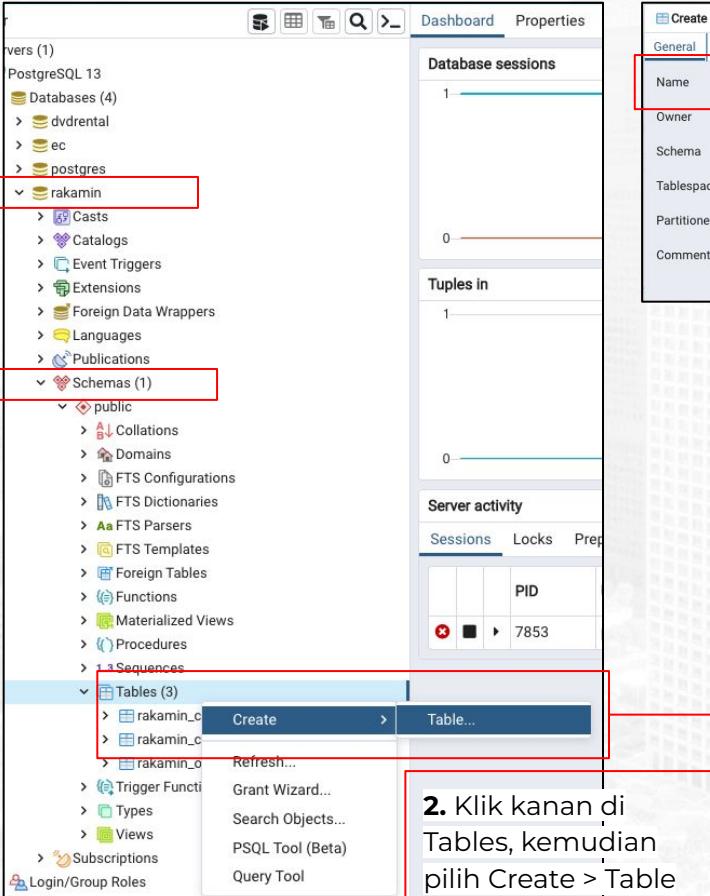


3. Pada Window yang muncul, masukkan nama Database yang diinginkan (custom) pada tab General.

4. Klik Save

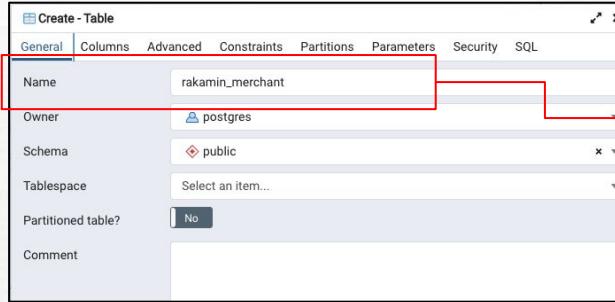
# pgAdmin - Membuat Tabel Baru

**1.** Buka dropdown Database yang diinginkan, lalu buka dropdown Schemas

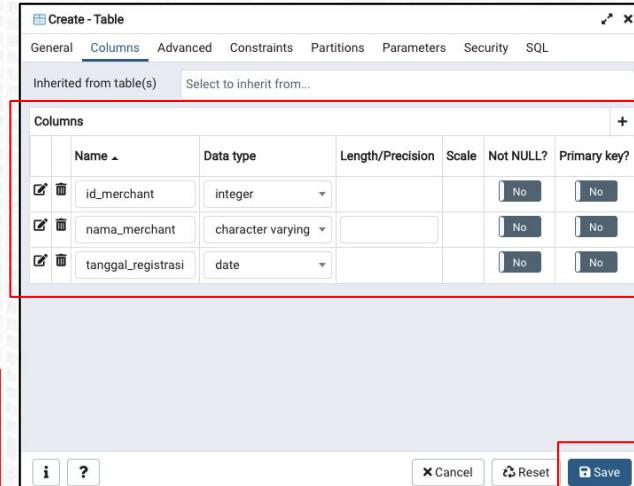


1. Buka dropdown Database yang diinginkan, lalu buka dropdown Schemas

2. Klik kanan di Tables, kemudian pilih Create > Table



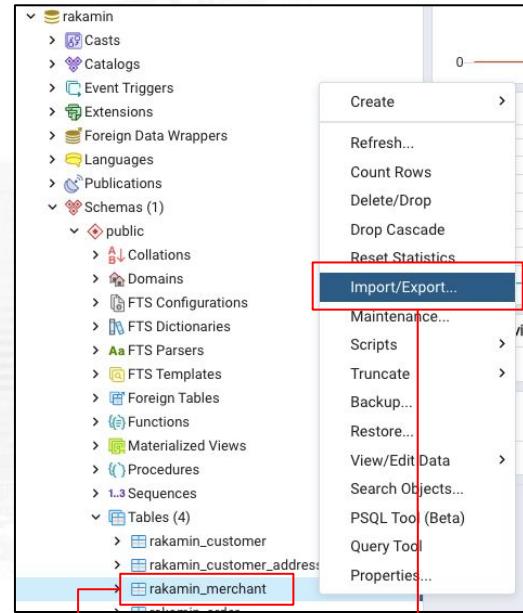
3. Masukkan nama tabel yang diinginkan (custom) pada tab General



4. Masukkan nama-nama kolom dan tipe data yang diinginkan.

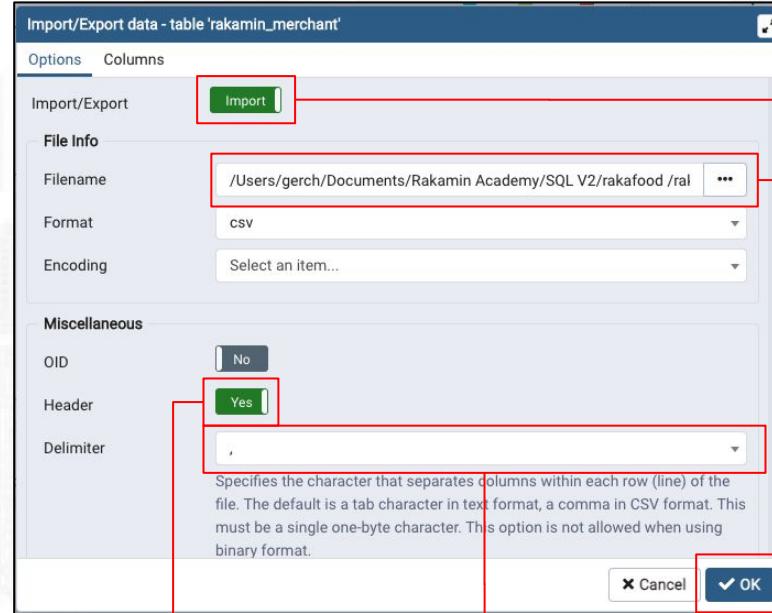
5. Klik Save

# pgAdmin - Mengisi Tabel dari File CSV



1. Klik kanan pada tabel yang ingin diisi.

2. Pilih Import/Export



3. Pilih Import

4. Pilih file CSV yang sudah ada di komputer Anda

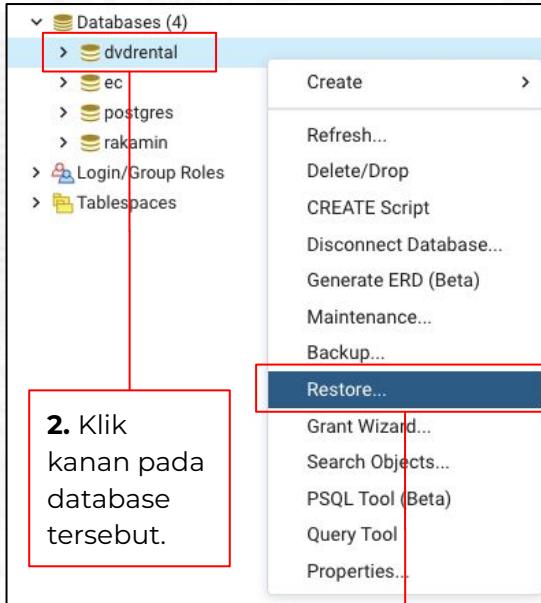
5. Pilih Header = Yes jika baris pertama CSV merupakan nama kolom

6. Pilih delimiter yang sesuai dengan file CSV

7. Klik OK

# pgAdmin - Restore Database

1. Buat database baru



2. Klik kanan pada database tersebut.

3. Pilih Restore

Restore (Database: dvdrental)

General    Restore options

Format

Directory

Filename

/Users/gerch/Downloads/dvdrental/

Number of jobs

Role name

Select an item...

Cancel

Restore

4. Pilih Format yang sesuai. Directory: jika dalam bentuk folder. Custom or tar: jika dalam bentuk .tar

5. Pilih file/folder yang relevan.

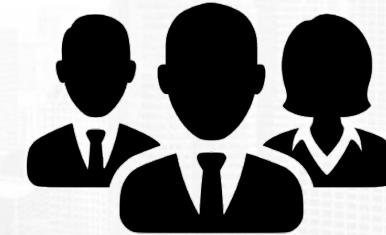
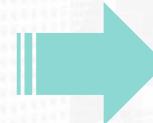
6. Klik Restore



# Konsep SQL

Database dan SQL, bagaimana kaitannya?

# General Data Pipeline

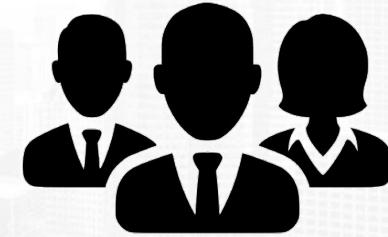
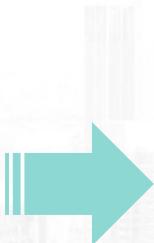


**Kumpulan data yang dimiliki tiap perusahaan**  
(contoh: data transaksi, data user yang baru login, dsb).

**Data disimpan ke dalam sebuah database** yang diatur secara sistematis oleh DBMS.

**Data diakses oleh Business Intelligence** dan/atau posisi lain yang berhubungan (Engineering).

# General Data Pipeline



**Kumpulan data yang dimiliki tiap perusahaan**  
(contoh: data transaksi, data user yang baru login, dsb).

**Data disimpan ke dalam sebuah database** yang diatur secara sistematis oleh DBMS.

**Data diakses oleh Business Intelligence** dan/atau posisi lain yang berhubungan (Engineering).

**Bagaimana caranya data dalam DB dapat diakses oleh pengguna??**

# SQL adalah jawabannya!



**SQL (Structured Query Language)**  
adalah bahasa standar untuk  
mengakses **RDBMS** dan mengolah  
data dalam *database*.

# SQL is a Standard

SQL became a standard of the **American National Standards Institute (ANSI)** in 1986, and of the **International Organization for Standardization (ISO)** in 1987.



International  
Organization for  
Standardization

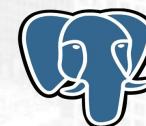
# SQL is a Standard, BUT ....

Walaupun SQL sudah menjadi standar yang tercatat di ANSI/ISO, namun ada beberapa versi **DIALEK SQL**.

**Tenang!** Karena sudah diatur standarnya, setiap versi / dialek SQL yang ada tidak akan berbeda jauh satu sama lain!



SQL/PLM



PostgreSQL

PL/pgSQL

# Tipe-Tipe Perintah SQL

- **DDL (Data Definition Language)**

Perintah yang berkaitan dengan skema dan deskripsi database.

**CREATE, ALTER, DROP, ...**

- **DML (Data Manipulation Language)**

Perintah yang berguna untuk memanggil data, melakukan perubahan, menghapus, dsb.

**SELECT, INSERT, UPDATE**

- **DCL (Data Control Language)**

Perintah yang berkaitan dengan pengaturan akses user (permissions)

**GRANT, REVOKE**



# Perintah Dasar SQL

# Perintah Dasar SQL

- **Select**, untuk memilih kolom yang ingin diambil
- **Limit**, membatasi jumlah row pada tabel yang diambil
- **Where**, membatasi/filter/mengambil spesifik data yang diambil
- **Order By**, mengurutkan data setelah data diambil

# Perintah Dasar 1 - SELECT

Digunakan untuk memilih kolom apa saja yang ingin diambil dari tabel

Deskripsi	Perintah dalam SQL
Ambil semua kolom	<code>SELECT *</code> <code>FROM &lt;nama tabel&gt;</code>
Ambil beberapa kolom saja	<code>SELECT kolom_1, kolom_2, ... kolom_10</code> <code>FROM &lt;nama tabel&gt;</code>

# Perintah Dasar 1 - SELECT

Contoh 1: Ambil semua kolom pada tabel **film**

Query Editor    Query History

```
1 select *
2 from film
3
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	description text
1	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria
2	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia
3	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India
4	98	Bright Encounters	A Fateful Yarn of a Lumberjack And a Feminist who must Conquer a Student in A Jet Boat
5	1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies
6	2	Ace Goldfinger	A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China
7	3	Adaptation Holes	A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory
8	4	Affair Prejudice	A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank
9	5	African Egg	A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico

# Perintah Dasar 1 - SELECT

Contoh 2 : Ambil kolom **title** pada tabel **film**

Query Editor    Query History

```
1 select title
2 from film
3
4
5
```

Data Output    Explain    Messages    Notifications

	title	
1	character varying (255)	🔒
1	Chamber Italian	
2	Grosse Wonderful	
3	Airport Pollock	
4	Bright Encounters	
5	Academy Dinosaur	

# Perintah Dasar 1 - SELECT

Contoh 3 : Ambil 3 kolom pada tabel **film**

Query Editor    Query History

```
1 select film_id, title, release_year
2 from film
3
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	release_year integer
1	133	Chamber Italian	2006
2	384	Grosse Wonderful	2006
3	8	Airport Pollock	2006
4	98	Bright Encounters	2006
5	1	Academy Dinosaur	2006
6	2	Ace Goldfinger	2006
7	3	Adaptation Holes	2006
8	4	Affair Prejudice	2006
9	5	African Egg	2006

## Perintah Dasar 2 - LIMIT

Digunakan untuk membatasi jumlah baris/row yang ingin diambil dari tabel (Perintah diletakan di paling bawah)

Deskripsi	Perintah dalam SQL
Ambil semua kolom dan hanya 10 baris / row	<code>SELECT * FROM &lt;nama tabel&gt; LIMIT 10</code>
Ambil beberapa kolom saja Dan hanya 5 baris / row	<code>SELECT kolom_1, kolom_2, ... kolom_10 FROM &lt;nama tabel&gt; LIMIT 5</code>

# Perintah Dasar 2 - LIMIT

Contoh 1:

Ambil semua kolom pada tabel **film**, tetapi hanya 5 baris / row saja.

Query Editor    Query History

```
1 select *
2 from film
3 limit 5
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	description text	release_year integer	langua smallir
1	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria	2006	
2	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia	2006	
3	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	2006	
4	98	Bright Encounters	A Fateful Yarn of a Lumberjack And a Feminist who must Conquer a Student in A Jet Boat	2006	
5	1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	2006	

# Perintah Dasar 2 - LIMIT

Contoh 2 :

Ambil 3 kolom pada tabel **film**, tetapi hanya 10 baris / row saja

Query Editor    Query History

```
1 select film_id, title, release_year
2 from film
3 limit 10
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	release_year integer
1	133	Chamber Italian	2006
2	384	Grosse Wonderful	2006
3	8	Airport Pollock	2006
4	98	Bright Encounters	2006
5	1	Academy Dinosaur	2006
6	2	Ace Goldfinger	2006
7	3	Adaptation Holes	2006
8	4	Affair Prejudice	2006
9	5	African Egg	2006
10	6	Agent Truman	2006

# Perintah Dasar 3 - WHERE

Digunakan untuk membatasi / filter pada spesifik kolom tertentu saja (Perintah diletakan tepat setelah *FROM (tabel)* )

Deskripsi	Perintah dalam SQL
Ambil semua kolom dan kolom_1 = xxxx	SELECT * FROM <nama tabel> WHERE kolom_1 = 'xxxx'
Ambil beberapa kolom saja dan kolom_2 = yyyy	SELECT kolom_1, kolom_2, ... kolom_10 FROM <nama tabel> WHERE kolom_2 = 'yyyy'

# Perintah Dasar 3 - WHERE

Contoh 1:

Ambil beberapa kolom pada tabel **film**, dimana kolom **rental\_duration** = 3.

Query Editor    Query History

```
1 select film_id, title, rental_duration
2 from film
3 where rental_duration = 3
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	rental_duration smallint
1	2	Ace Goldfinger	3
2	6	Agent Truman	3
3	9	Alabama Devil	3
4	17	Alone Trip	3
5	21	American Circus	3
6	23	Anaconda Confessions	3
7	25	Angels Life	3
8	26	Annie Identity	3
9	37	Arizona Bang	3

## Perintah Dasar 3 - WHERE

Kita juga bisa menggunakan tipe data lain selain integer

Tipe data	Perintah dalam SQL
string	SELECT * FROM <nama tabel> WHERE kolom_1 = 'xxxx'
date	SELECT kolom_1, kolom_2, ... kolom_10 FROM <nama tabel> WHERE kolom_2 = 'yyyy-mm-dd'
null	SELECT kolom_1, kolom_2, ... kolom_10 FROM <nama tabel> WHERE kolom_3 IS NULL

# Perintah Dasar 3 - WHERE

Contoh 2 : Ambil beberapa kolom pada tabel **film**, dimana kolom **rating** = NC-17

Query Editor    Query History

```
1 select film_id, title, release_year, rating
2 from film
3 where rating = 'NC-17'
4
5
```

Data Output   Explain   Messages   Notifications

	film_id [PK] integer	title character varying (255)	release_year integer	rating mpaa_rating	
1	133	Chamber Italian	2006	NC-17	
2	3	Adaptation Holes	2006	NC-17	
3	10	Aladdin Calendar	2006	NC-17	
4	14	Alice Fantasia	2006	NC-17	
5	15	Alien Center	2006	NC-17	
6	16	Alley Evolution	2006	NC-17	
7	27	Anonymous Human	2006	NC-17	
8	29	Antitrust Tomatoes	2006	NC-17	
9	31	Apache Divine	2006	NC-17	
-	-	-	-	-	-

# Perintah Dasar 3 - WHERE

Contoh 3 : Ambil semua kolom pada tabel **rental**, dimana kolom **rental\_date** = 2005-07-29 08:40:36

Query Editor    Query History

```
1 select *
2 from rental
3 where rental_date = '2005-07-29 08:40:36'
4
5
```

Data Output   Explain   Messages   Notifications

	rental_id [PK] integer	rental_date timestamp without time zone	inventory_id integer	customer_id smallint	return_date timestamp without time zone	staff_id smallint	last_update timestamp without time zone
1	8477	2005-07-29 08:40:36	3571	354	2005-08-06 08:28:36	2	2006-02-16 02:30:53
2	8478	2005-07-29 08:40:36	3742	162	2005-08-01 10:23:36	1	2006-02-16 02:30:53

## Perintah Dasar 3 - WHERE

Contoh 4 : Ambil semua kolom pada tabel **address**, dengan kolom **address2** sama dengan null.

```
Query Editor Query History  
1 select *  
2 from address  
3 where address2 is null
```

Detailed Address Information								
	address_id [PK] integer	address character varying (50)	address2 character varying (50)	district character varying (20)	city_id smallint	postal_code character varying (10)	phone character varying (20)	last_update timestamp
1	1	47 MySakila Drive	[null]	Alberta	300			2023-10-01 12:00:00
2	2	28 MySQL Boulevard	[null]	QLD	576			2023-10-01 12:00:00
3	3	23 Workhaven Lane	[null]	Alberta	300		14033335568	2023-10-01 12:00:00
4	4	1411 Lillydale Drive	[null]	QLD	576		6172235589	2023-10-01 12:00:00

# Penggunaan Operator Aritmatika

Tidak hanya menggunakan angka saja, namun kita juga bisa menggunakan operator aritmatika di dalam query.

Contoh 5 : Ambil kolom **customer\_id**, **rental\_id**, **amount** pada tabel **payment**, kali kolom **amount** dengan 14000

Query Editor    Query History

```
1 select customer_id, rental_id, amount, amount * 14000
2 from payment
```

Data Output   Explain   Messages   Notifications

	customer_id smallint	rental_id integer	amount numeric (5,2)	?column? numeric	
1	341	1520	7.99	111860.00	
2	341	1778	1.99	27860.00	
3	341	1849	7.99	111860.00	
4	341	2829	2.99	41860.00	
5	341	3130	7.99	111860.00	
6	341	3382	5.99	83860.00	
7	342	2190	5.99	83860.00	
8	342	2914	5.99	83860.00	
9	342	3081	2.99	41860.00	
10	249	1547	4.00	56960.00	

## Perintah Dasar 4 - ORDER BY

Digunakan untuk mengurutkan data dari besar ke kecil (Descending) atau kecil ke besar (Ascending) pada kolom tertentu setelah data diambil.

Deskripsi	Perintah dalam SQL
Ambil semua kolom dan diurutkan berdasarkan kolom_1 dari kecil ke besar (ascending).	SELECT * FROM <nama tabel> ORDER BY kolom_1 ASC
Ambil beberapa kolom saja dan diurutkan berdasarkan kolom_1 dari besar ke kecil (descending)	SELECT kolom_1, kolom_2, ... kolom_10 FROM <nama tabel> ORDER BY kolom_1 DESC

# Perintah Dasar 4 - ORDER BY

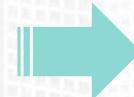
Query Editor    Query History

```

1 select *
2 from customer
  
```

Data Output   Explain   Messages   Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)
1	524	1	Jared	Ely
2	1	1	Mary	Smith
3	2	1	Patricia	Johnson
4	3	1	Linda	Williams
5	4	2	Barbara	Jones
6	5	1	Elizabeth	Brown
7	6	2	Jennifer	Davis
8	7	1	Maria	Miller
9	8	2	Susan	Wilson
10	9	2	Margaret	Moore



Query Editor    Query History

```

1 select *
2 from customer
3 order by first_name
  
```

Data Output   Explain   Messages   Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)
1	375	2	Aaron	Selby
2	367	1	Adam	Gooch
3	525	2	Adrian	Clary
4	217	2	Agnes	Bishop
5	389	1	Alan	Kahn
6	352	1	Albert	Crouse
7	568	2	Alberto	Henning
8	454	2	Alex	Gresham
9	439	2	Alexander	Fennell
10	429	2	Alfred	Gacillan

# Perintah Dasar 4 - ORDER BY

Query Editor    Query History

```
1 select *
2 from customer
```

Data Output   Explain   Messages   Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)
1	524	1	Jared	Ely
2	1	1	Mary	Smith
3	2	1	Patricia	Johnson
4	3	1	Linda	Williams
5	4	2	Barbara	Jones
6	5	1	Elizabeth	Brown
7	6	2	Jennifer	Davis
8	7	1	Maria	Miller
9	8	2	Susan	Wilson
10	n	3	Margaret	Moore

Query Editor    Query History

```
1 select *
2 from customer
3 order by first_name desc
```

Data Output   Explain   Messages   Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)
1	479	1	Zachary	Hite
2	174	2	Yvonne	Watkins
3	190	2	Yolanda	Weaver
4	212	2	Wilma	Richards
5	359	2	Willie	Markham
6	219	2	Willie	Howell
7	303	2	William	Satterfield
8	578	2	Willard	Lumpkin
9	469	2	Wesley	Bull
10	115	1	Mandy	Warren



# F.A.Q

- **Apakah SQL case sensitive? Huruf kecil / besar sama tidak?**  
Keyword di SQL tidak case sensitive. Namun untuk nama tabel / kolom mengikuti aturan RDBMS yang digunakan, ada yang case sensitive dan ada juga yang tidak.
- **Apakah harus menggunakan semicolon (;) di akhir query?**  
Semicolon menandakan akhir dari satu query. Penggunaan semicolon ini wajib jika ingin menjalankan lebih dari satu query sekaligus. Jika hanya ingin menjalankan satu query, tergantung dari aturan RDBMS yang digunakan, biasanya tidak perlu.
- **Apakah ada ketentuan tertentu untuk indentasi?**  
Tidak ada. Indentasi biasanya hanya untuk keperluan kosmetik/estetika agar query lebih mudah dipahami.



**Terima Kasih**



# Extras

# Downloadables

Download PostgreSQL di sini

Download Superstore CSVs di sini

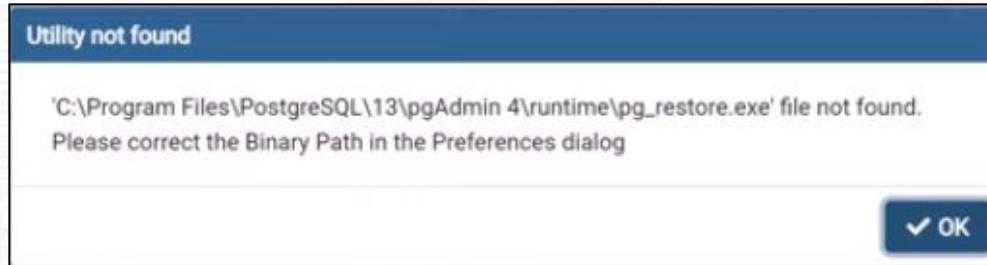
Download DVDRental di sini

# Membuat Tabel Baru dengan SQL

Query Editor    Query History

```
1 CREATE TABLE superstore_customer (
2     customer_id varchar,
3     customer_name varchar,
4     segment varchar,
5     country varchar,
6     city varchar,
7     state varchar,
8     postal_code varchar,
9     region varchar
10 )
```

# Muncul Error saat Restore Database?



**Coba salah satu dari solusi berikut!**

Solusi 1

Solusi 2