SQL AND WORKING WITH DATABASES

GROWIA

WILLY RIZKIYAN

Mining company switch to data field

LET'S CONNECT:

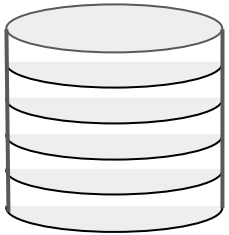
willyrizkiyan@gmail.com
https://linkedin.com/in/willyrizkiyan

SQL AND WORKING WITH DATABASES

- What is SQL and Databases -

GROWIA

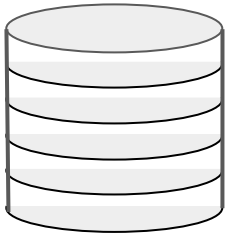
WHAT IS DATABASE?



- Suatu sistem dengan struktur dan metode yang jelas untuk menyimpan dan mengambil isi data
- Ada beberapa jenis sistem database di dunia saat ini, tapi yang paling populer adalah sistem database relasional atau Relational Database Management System (RDBMS). RDBMS menggunakan bahasa SQL untuk melakukan komunikasi
- Sistem database lain adalah network database, hierarchical database, object-oriented database, dan lain-lain.

GROWIA

WHAT IS SQL?



- SQL (Structured Query Language) adalah bahasa pemrograman khusus yang digunakan untuk mengelola dan mengakses database.
- SQL digunakan oleh Sebagian besar sistem database relasional seperti MySQL, PostgreSQL, SQL Server, Oracle, dan banyak lainnya.

Yang bisa dilakukan:
- Melihat data
- Filter data berdasarkan kriteria
- Mengurutkan hasil
- Menambahkan, mengubah, ataupun menghapus

SQL AND WORKING WITH DATABASES

- Why do we learn SQL -

GROWIA

SKILL SQL SELALU ADA PADA LOWONGAN UNTUK MENJADI PRAKTISI DATA

Data Analyst
Moladin · Jakarta, Jakarta, Indonesia (On-site)

- Minimum of 1-2 years of experience as a data
- Proficient in SQL and data visualization tools

Sr Data Scientist
JULO · Indonesia (Remote)

- Experience working with relational SQL databases

Senior Data Engineer
JULO · Jakarta Selatan, Jakarta, Indonesia (Hybrid)

- Able to write basic to advanced SQL queries

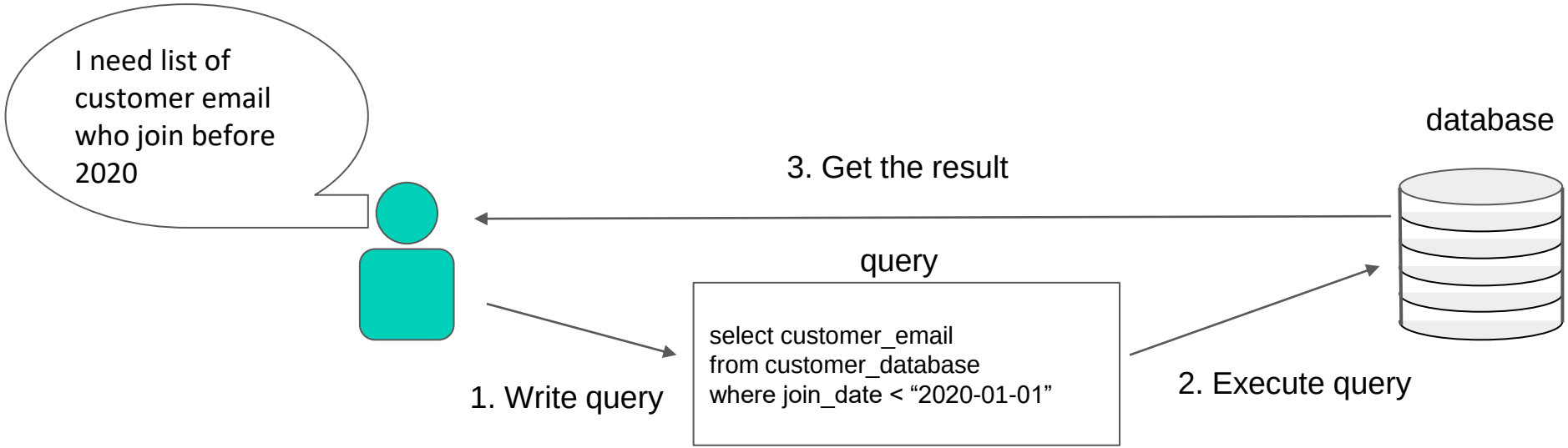
GROWIA

KETIKA ANALISA DATA, PERLU QUERY

- Berapa banyak orang yang bertransaksi kemarin?
- Berapa total pendapatan bulan ini?
- Siapa saja yang melakukan pembelian dalam 7 hari terakhir?
- Siapa saja peminjam yang bisa diklasifikasikan lancar atau tidak?
- Adakah customer yang tadinya aktif transaksi menjadi tidak pernah transaksi?

SQL

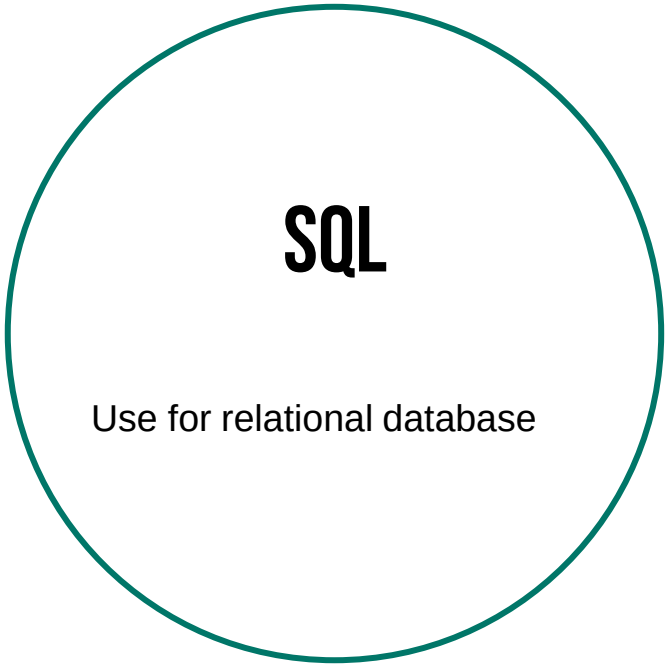
PROCESS


I need list of customer email who join before 2020
database
3. Get the result
query
select customer_email
from customer_database
where join_date < “2020-01-01”
1. Write query
2. Execute query

SQL AND WORKING WITH DATABASES

- SQL and Relational Database -

SQL VS NOSQL



NOSQL

Use for non-relational database

GROWIA

JENIS DATABASE

RDBMS / SQL

- Terdiri dari tabel-tabel (yang biasanya saling berhubungan menggunakan Primary key dan Foreign Key)
- Skema sudah ditetapkan sebelum data dimasukkan
- Cocok untuk data aplikasi dengan data yang strukturnya konsisten
- Bisa untuk query yang komplek

Non-RDBMS / NoSQL

- Terdiri dari dokumen - dokumen yang menggunakan key - value, bentuknya tidak teratur
- Skema dinamis, bisa dibilang tidak terstruktur
- Cocok untuk data log, atau data yang jumlah variabel tidak pasti
- Tidak cocok untuk query yang komplek

GROWIA

JENIS DATABASE

RDBMS / SQL

- MySQL
- PostgreSQL
- SQLite
- Oracle
- SQL-Server
- Google BigQuery
- Redshift
- Hive

Non-RDBMS / NoSQL

- MongoDB
- Redis
- Casandra
- Hbase
- Neo4j

POPULAR RELATIONAL DATABASES

GROWIA

TABULAR DATA

	A	B	C	D	E	F
1	Country	Salesperson	Order Date	OrderID	Units	Order Amoun
2	USA	Fuller	1/01/2011	10392	13	1,440.00
3	UK	Gloucester	2/01/2011	10397	17	716.72
4	UK	Bromley	2/01/2011	10771	18	344.00
5	USA	Finchley	3/01/2011	10393	16	2,556.95
6	USA	Finchley	3/01/2011	10394	10	442.00
7	UK	Gillingham	3/01/2011	10395	9	2,122.92
8	USA	Finchley	6/01/2011	10396	7	1,903.80
9	USA	Callahan	8/01/2011	10399	17	1,765.60
10	USA	Fuller	8/01/2011	10404	7	1,591.25
11	USA	Fuller	9/01/2011	10398	11	2,505.60
12	USA	Coghill	9/01/2011	10403	18	855.01
13	USA	Finchley	10/01/2011	10401	7	3,868.60
14	USA	Callahan	10/01/2011	10402	11	2,713.50
15	UK	Rayleigh	13/01/2011	10406	15	1,830.78
16	USA	Callahan	14/01/2011	10408	10	1,622.40
17	USA	Farnham	14/01/2011	10409	19	319.20
18	USA	Farnham	15/01/2011	10410	16	802.00

Tiap baris merepresentasikan 1 rekaman data
Setiap kolom merepresentasikan atribut dari rekaman

LONG FORMAT VS WIDE FORMAT

country	year	gdp
Indonesia	2021	1.186
Indonesia	2020	1.059
Indonesia	2019	1.119
Singapore	2021	0.397
Singapore	2020	0.343
Singapore	2019	0.375
Malaysia	2021	0.373
Malaysia	2020	0.337
Malaysia	2019	0.365

country	2019	2020	2021
Indonesia	1.186	1.059	1.119
Singapore	0.397	0.343	0.375
Malaysia	0.373	0.337	0.365

Tidak cocok untuk relational database

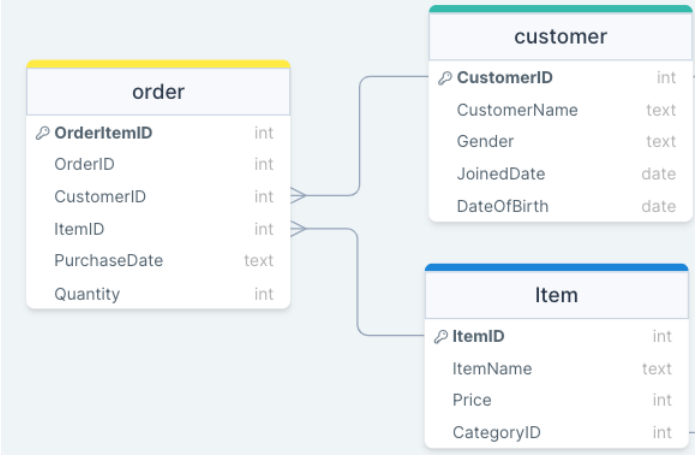
GROWIA

DATABASE SCHEMA

OrderItemID	OrderID	CustomerID	ItemID	PurchaseDate	Quantity
412312311	41231231	1	1	2001-12-03	2
412312313	41231231	1	3	2001-12-03	2
653634225	65363422	2	5	2001-12-04	1
840912406	84091240	3	6	2001-12-05	1

CustomerID	CustomerName	Gender	JoinedDate	DateOfBirth
1	Daniel	Male	2000-01-01	1979-12-04
2	Jackson	Male	2000-01-01	1980-03-03
3	Vivian	Female	2000-01-01	1985-02-01
4	Mikhail	Male	2000-01-02	1970-01-02

ItemID	ItemName	Price	CategoryID
1	Electric Fan	30000	199
2	Plastic Chair	20000	123
3	Gaming Laptop15"	12000000	145
4	Wet tissue 100 shee	2000	158



ONE TO MANY RELATIONSHIP

Jika customer punya banyak nomor hp, apakah menyimpan data seperti ini?

CustomerID	CustomerName	Gender	JoinedDate	DateOfBirth	PhoneNumber
1	Daniel	Male	2000-01-01	1979-12-04	08961234567
1	Daniel	Male	2000-01-01	1979-12-04	08964214213
1	Daniel	Male	2000-01-01	1979-12-04	08961241241
2	Jackson	Male	2000-01-01	1980-03-03	08121241223
3	Vivian	Female	2000-01-01	1985-02-01	07611321323
4	Mikhail	Male	2000-01-02	1970-01-02	08527986789

Tidak efisien

CustomerID	CustomerName	Gender	JoinedDate	DateOfBirth
1	Daniel	Male	2000-01-01	1979-12-04
2	Jackson	Male	2000-01-01	1980-03-03
3	Vivian	Female	2000-01-01	1985-02-01
4	Mikhail	Male	2000-01-02	1970-01-02

CustomerPhone	CustomerID	PhoneNumber
124142413	1	08961234567
542524554	1	08964214213
146243235	1	08961241241
346146326	2	08121241223
134612426	3	07611321323
146146364	4	08527986789

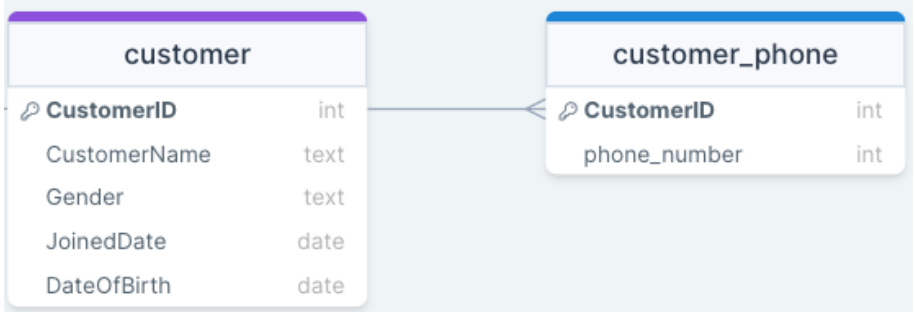
GROWIA

ONE TO MANY RELATIONSHIP

Nilai pada satu table berhubungan dengan beberapa value di table lainnya

CustomerID	CustomerName	Gender	JoinedDate	DateOfBirth
1	Daniel	Male	2000-01-01	1979-12-04
2	Jackson	Male	2000-01-01	1980-03-03
3	Vivian	Female	2000-01-01	1985-02-01
4	Mikhail	Male	2000-01-02	1970-01-02

CustomerPhone	CustomerID	PhoneNumber
124142413	1	08961234567
542524554	1	08964214213
146243235	1	08961241241
346146326	2	08121241223
134612426	3	07611321323
146146364	4	08527986789



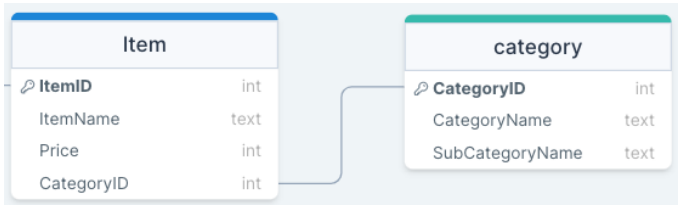
GROWIA

ONE TO ONE RELATIONSHIP

Tiap nilai pada 1 tabel hanya berhubungan dengan 1 data pada table lain

ItemID	ItemName	Price	CategoryID
1	Electric Fan	30000	199
2	Plastic Chair	20000	123
3	Gaming Laptop15"	12000000	145
4	Wet tissue 100 shee	2000	158

CategoryID	Category	SubCategory
199	Electronics	Fan
123	Household	Chair
145	Electronics	Laptop
158	Hygene	Tissue



SQL AND WORKING WITH DATABASES

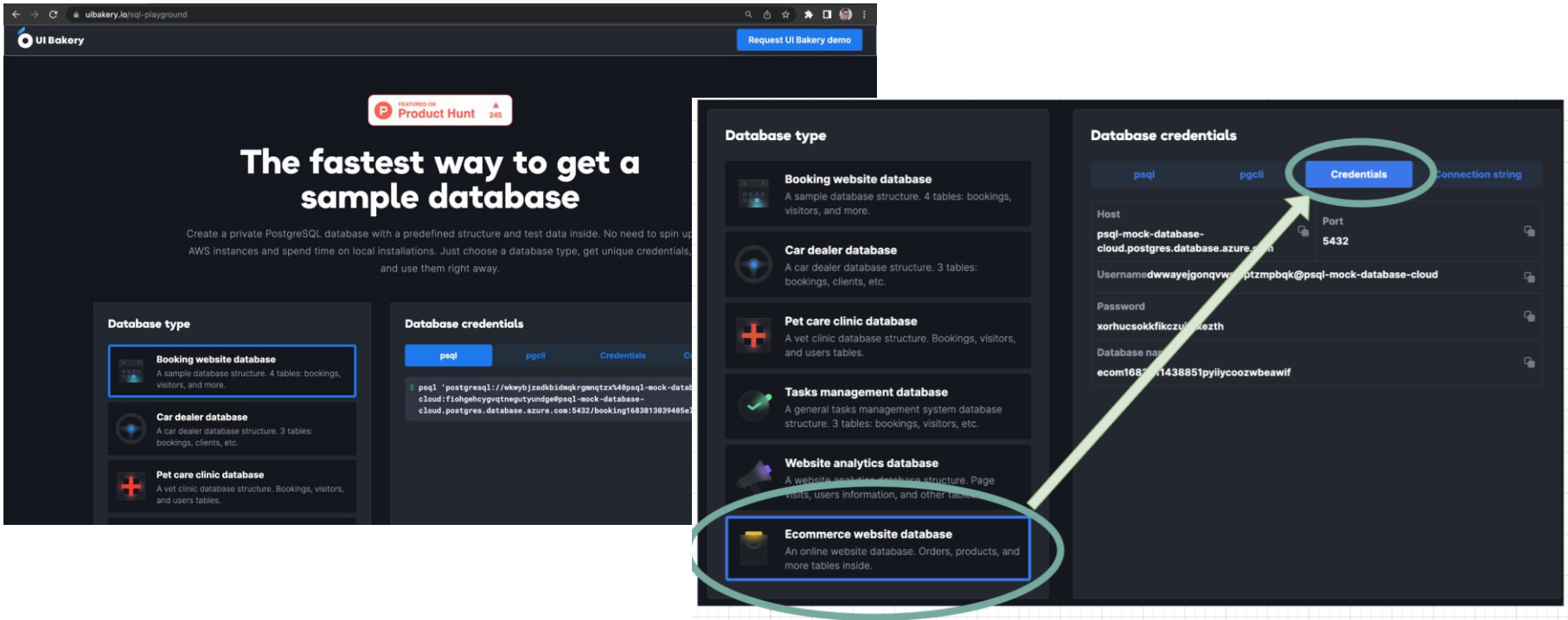
- Installation -

DBEAVER

Download link : https://dbeaver.io/download/

CONNECT TO DATABASE

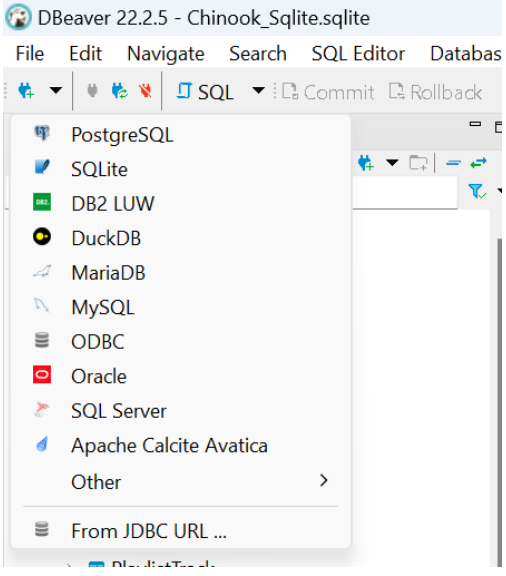
Buka website https://uibakery.io/sql-playground

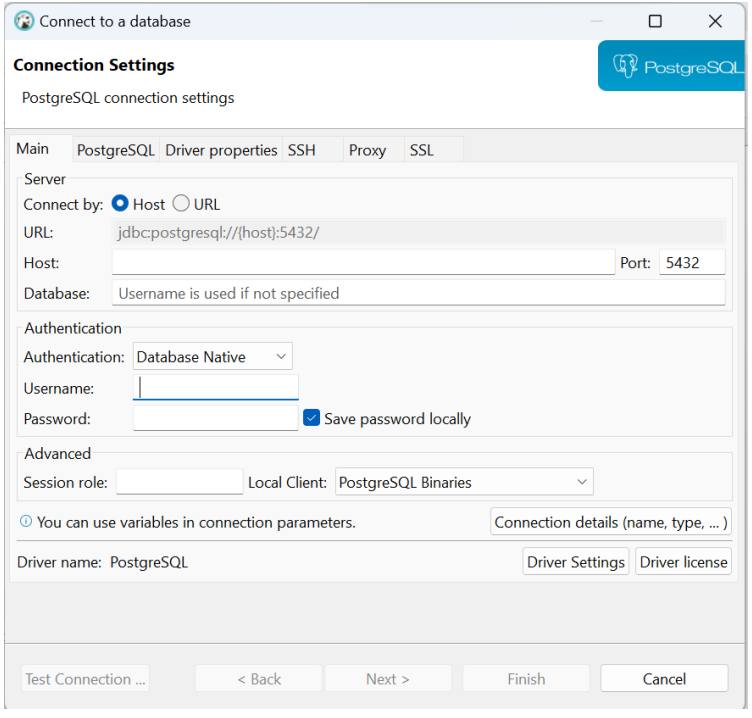


CONNECT TO DATABASE

Buat koneksi ke PostgreSQL

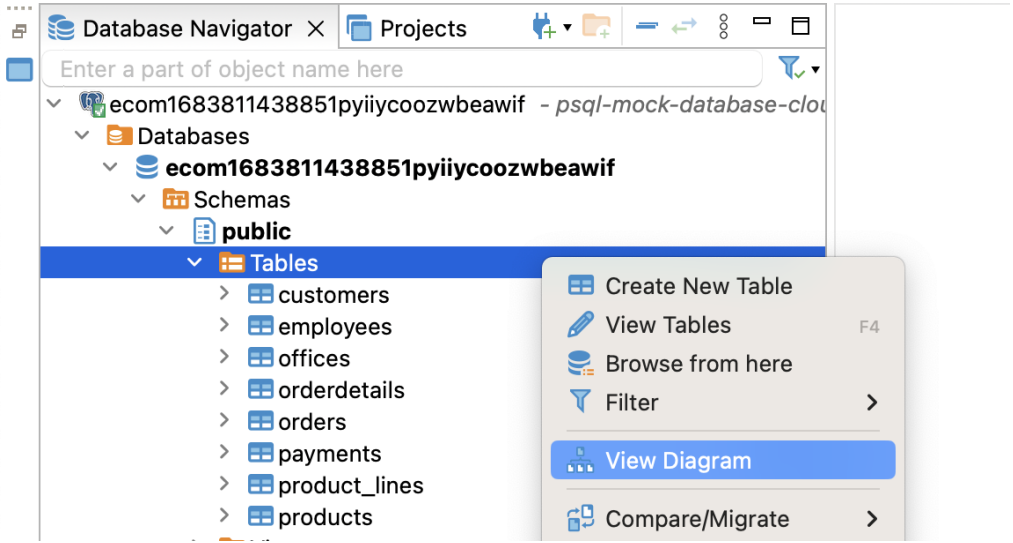
- Klik *dropdown* ikon “Connection” di kiri atas, pilih “PostgreSQL”
- Isi semua detail lalu klik “Finish”

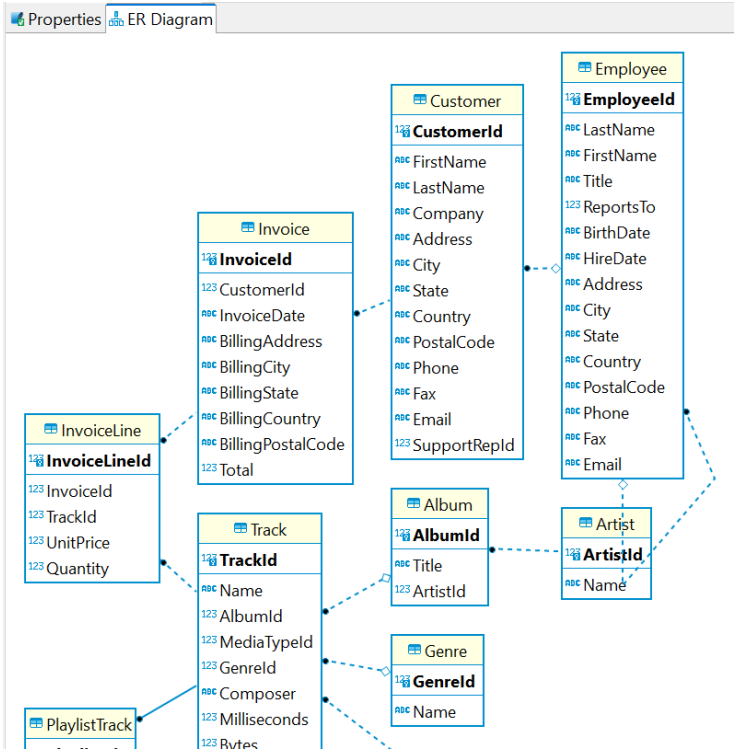




CONNECT TO DATABASE

Melihat diagram





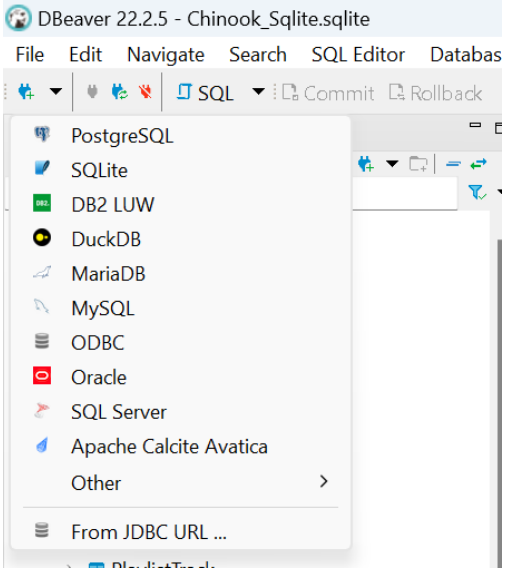
SQLITE

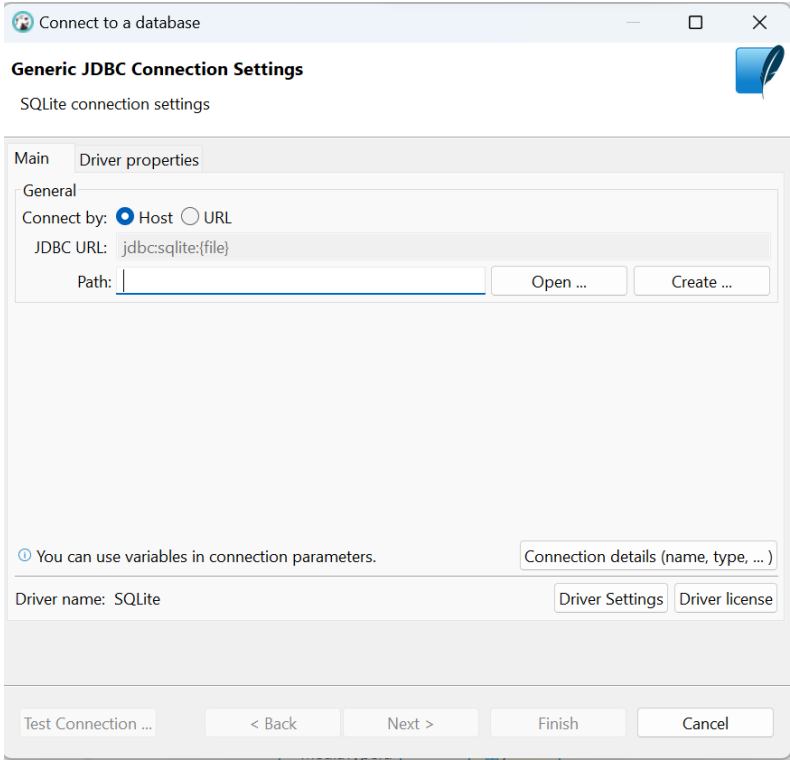
1. Serverless and self contained
2. Small and fast
3. Easy to setup

CONNECT TO DATABASE

Buat koneksi ke SQLite

- Klik *dropdown* ikon “Connection” di kiri atas, pilih “SQLite”
- Pilih Open di sebelah Path dan cari file SQLite di local





LET'S PLAY AROUND

- Data Types -

GROWIA

DATA TYPES

STRING
Rangkaian dari huruf atau karakter yang diapit olehkarakter kutip tunggal (‘).
Contoh : ‘Saya sangat senang belajar di Growia’

NUMERIC
Rangkaian digit yang mewakili angka rasional bulat dan pecahan, positif maupun negatif.
Contoh : 11, -2, 0.3

BOOLEAN
Nilai lateral Menunjukkan nilai kebenaran, dan hanya memiliki dua nilai.
Contoh : True dan False atau 1 dan 0

DATE AND TIME
Nilai literal dari date (tanggal) dan time (waktu) berupa string yang diawali dengan keyword DATE, TIME dan TIMESTAMP
Contoh : DATE '2020-12-18’, TIME '15:03:01’, TIMESTAMP '2020-12-18 15:03:01'

LET'S PLAY AROUND

- Query Dasar -

GROWIA

STRUKTUR QUERY

SELECT : Memilih kolom yang diinginkan
FROM : Referensi table yang ingin digunakan
WHERE : Kondisi pengambilan data
GROUP BY : Pengelompokkan berdasarkan kolom
ORDER BY : Data diurutkan berdasarkan kolom apa
HAVING : Kondisi pengambilan data untuk kolom agregasi
LIMIT : Berapa banyak data yang ingin diambil

```
SELECT nama_kolom, agregasi
FROM nama_tabel
WHERE filter_tabel
GROUP BY nama_kolom
HAVING filter_agregasi
ORDER BY nama_kolom
LIMIT angka
```

GROWIA

SELECT

Memilih semua kolom
Select *
from customer

Memilih kolom tertentu
Select FirstName, LastName, Country
from customer

Memilih unique value dari kolom tertentu
select DISTINCT Country
from customer

Tips:
Lihat hasil sampelnya terlebih dahulu
Select * from customer limit 10

GROWIA

FILTERING (WHERE)

=	sama dengan
<>, !=	tidak sama dengan
<	kurang dari
>	lebih dari
<=	kurang dari atau sama dengan
>=	lebih dari atau sama dengan

```
select FirstName, LastName
from Customer c
where Country = 'Brazil'
```

```
select CustomerId, InvoiceDate
from Invoice i
where InvoiceDate >= '2010-01-01'
```

WHERE AND OR

AND digunakan untuk menggabungkan 2 atau lebih kondisi untuk memfilter dengan semua kondisi harus bernilai benar

```
select FirstName, LastName
from Customer c
where Country = 'Portugal' and city = 'Porto'
```

OR digunakan untuk menggabungkan 2 atau lebih kondisi untuk memfilter dengan minimal 1 kondisi harus bernilai benar

```
select name, Composer
from Track t
where GenreId = 1 or GenreId = 2
```

WHERE BETWEEN

BETWEEN bisa digunakan untuk menggantikan **AND** yang mempunyai rentang

```
select CustomerId, InvoiceDate
from Invoice i
where InvoiceDate >= '2010-01-01' and InvoiceDate <= '2010-12-31'
```

menjadi

```
select CustomerId, InvoiceDate
from Invoice i
where InvoiceDate between '2010-01-01' and '2010-12-31'
```

WHERE IN

IN bisa digunakan untuk menggantikan **OR** pada kolom yang sama

```
select name, Composer
from Track t
where GenreId = 1 or GenreId = 2
```

menjadi

```
select name, Composer
from Track t
where GenreId in (1,2)
```

WHERE NULL

NULL berarti missing data

```
select name, GenreId, Composer
from Track t
where Composer is NULL
```

```
select name, GenreId, Composer
from Track t
where Composer is not NULL
```

GROWIA

WHERE LIKE DAN NOT LIKE

= digunakan untuk memfilter kata yang nilainya sama persis
LIKE digunakan untuk memfilter data berdasarkan sebagian kata saja menggunakan bantuan karakter %

g% : cari nilai diawali huruf ‘g’
%g : cari nilai diakhiri huruf ‘g’
%g% : cari nilai di posisi manapun ada huruf ‘g’
_g% : cari nilai berisi huruf ‘g’ di posisi kedua
g__% : cari nilai diawali huruf g dengan minimal 3 karakter
G%a : cari nilai yang diawali huruf ‘g’ dan diakhiri dengan huruf ‘a’

```
select name, Composer
from Track
where name like '%love%'
```

```
select name, Composer
from Track
where name like 'love%'
```

GROWIA

ORDER BY

ORDER BY digunakan untuk mengurutkan hasil
Secara default urutannya ASCENDING (dari kecil ke besar)
Untuk kebalikannya, pakai ORDER BY nama_kolom DESC

```
select name, Composer
from Track
where name like '%love%' and Composer is not null
order by Composer DESC
```

GROWIA

LATIHAN

Tampilkan nama track beserta composer dari table track yang tidak mengandung data yang hilang dimana huruf paling depan “G” pada nama track atau huruf paling belakang ‘A’ pada composer serta urutkan berdasarkan abjad pada nama composer

LET'S PLAY AROUND

- AGGREGATION and GROUPING -

COUNT

Menghitung jumlah baris dalam table
Select count(*) from track

Menghitung jumlah data tidak null pada kolom tertentu
Select count(composer) from track

Menghitung nilai unik dari kolom tertentu
Select count(distinct composer) from track

GROWIA

AVG, MIN, MAX, SUM

Melakukan kalkulasi terhadap kolom tertentu

```
Select
        avg(total),
        min(total),
        max(total),
        sum(total)
from Invoice
```

GROWIA

ALIAS (AS)

Jika menyadari, pada code sebelumnya akan menghasilkan nama kolom ‘avg(total)’.
Bisa membingungkan jika fungsinya panjang, sehingga bisa diubah nama kolom menggunakan AS

```
Select
        avg(total) as rata_penjualan,
        min(total) as minimum_penjualan,
        max(total) as maximum_penjualan,
        sum(total) as total_penjualan,
from Invoice
```

AGGREGATION

Pengumpulan sejumlah benda yang terpisah menjadi satu

GROUPING

Biasanya dilakukan bersama dengan proses agregasi berdasarkan kelompok

GROWIA

GROUP BY

Digunakan untuk agregasi tapi berdasarkan kelompok

```
Select
        BillingCountry,
        count(InvoiceId) as total_pembelian
from Invoice i
group by 1
```

GROWIA

HAVING

Mirip WHERE, tapi digunakan untuk data yang diagregasi

```
Select
        BillingCountry,
        count(*) as total_pembelian,
        round(avg(total)) as rata_pembelian
from Invoice i
group by 1
having rata_pembelian >= 6
```

GROWIA

LATIHAN

Tampilkan semua customer id, billing country, jumlah_pembelian, serta rata_pembelian pada tahun 2011 hanya untuk customer yang punya rata_pembelian lebih besar dari 5 pada table invoice

LET'S PLAY AROUND

- CASE WHEN -

GROWIA

CASE WHEN

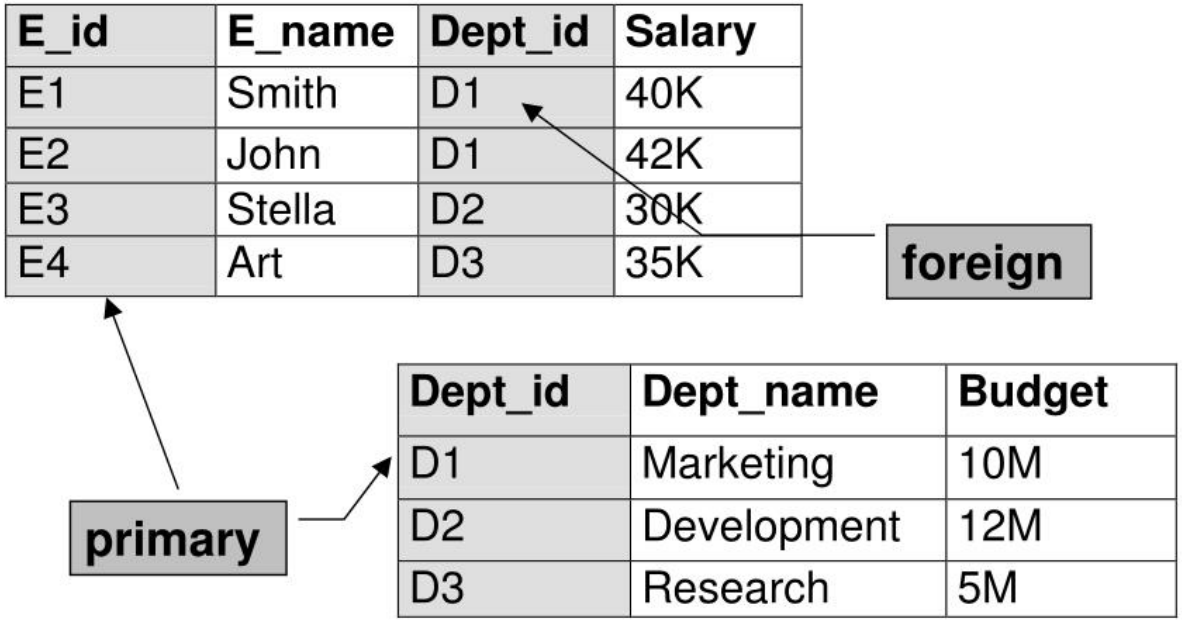
Digunakan untuk membuat / mengubah nilai dari field yang sudah ada
Misalnya untuk membuat kategori customer dari data yang sudah ada

```
select
        CustomerId,
        Country,
        case country
        when 'USA' then 'Domestic'
        else 'Foreign' end as grupcustomer
from Customer c
```

LET'S PLAY AROUND

- JOIN dan UNION -

PRIMARY KEY DAN FOREIGN KEY



Primary key harus unik di setiap table

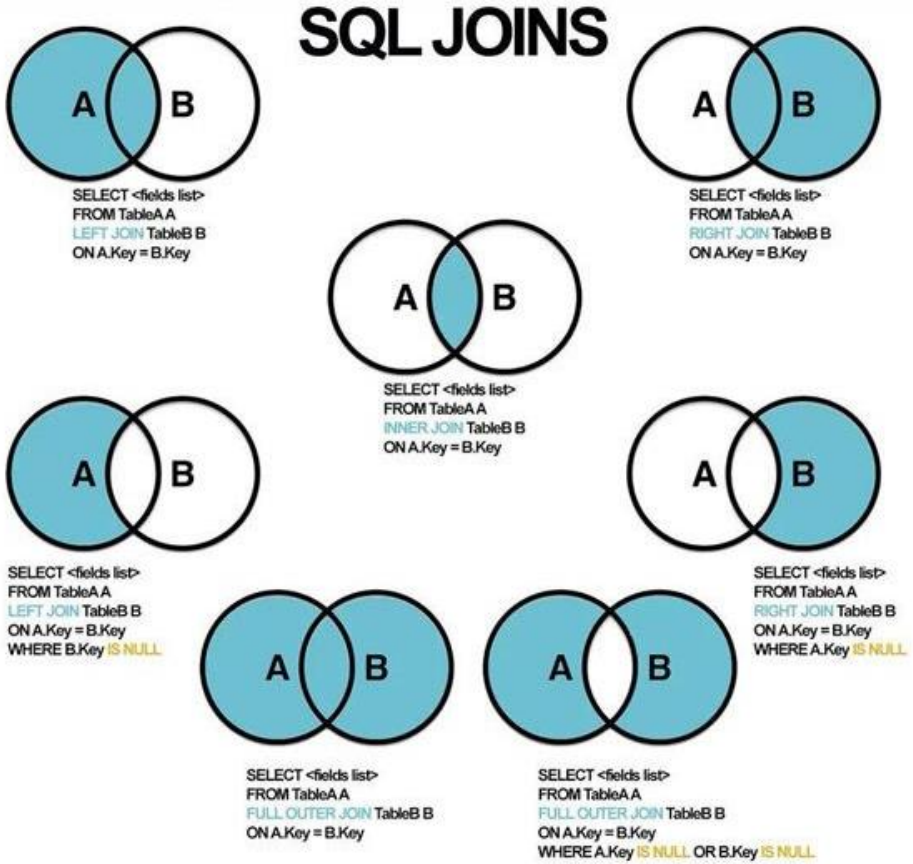
Foreign key merupakan kolom yang bisa dihubungkan dengan kolom pada table lain

JOIN

Pada prinsipnya, JOIN adalah menghubungkan suatu tabel ke tabel yang lain.

LEFT JOIN
Tampilkan semua data dari tabel kiri dan tampilkan value dari tabel kanan jika ada yang match

INNER JOIN
Hanya menampilkan data yang ada di kedua tabel



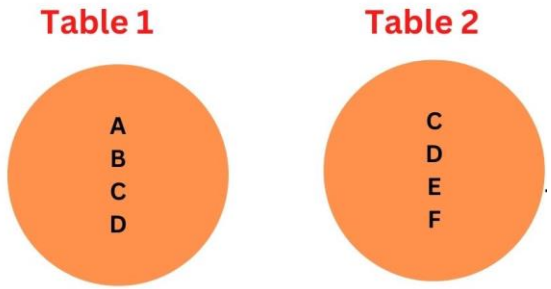
GROWIA

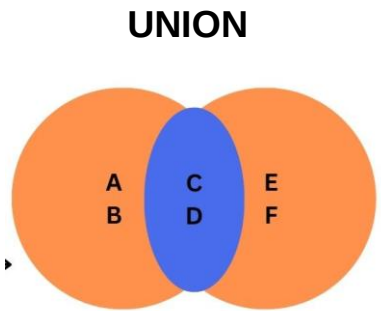
UNION

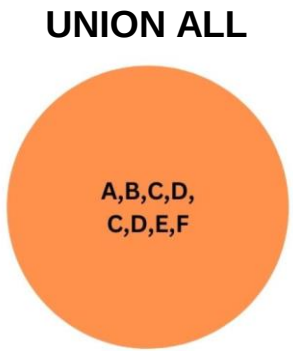
UNION
Untuk menggabungkan baris data dan hanya mengambil unik nya saja.
Jumlah dan posisi kolom harus sama persis

UNION ALL
Untuk menggabungkan baris data tanpa membuat menjadi unik, langsung gabungkan semuanya

Select * from tabelA
Union
Select * from tabelB







GROWIA

LATIHAN

Cari semua judul album yang dinyanyikan oleh Artist ‘Aerosmith’ dan ‘Nirvana’

LET'S PLAY AROUND

- **SUBQUERY dan CTE** -

SUBQUERY

Sebuah **subquery** pada SQL adalah perintah SELECT yang ditempatkan sebagai bagian dari query SELECT lain.

Ciri dari subquery adalah diapit oleh tanda kurung ().

Subquery digunakan untuk beberapa hal, yaitu:

- Filtering nilai
- Menambahkan nilai
- Tabel turunan

```
Select
        CustomerId,
        InvoiceDate,
        BillingCountry,
        Total
FROM invoice
WHERE Total > (select avg(Total) from invoice)
```

GROWIA

CTE

Common Table Expression (CTE) adalah satu hasil query atau subquery yang terdapat dalam satu statement query lain dan dapat direferensikan berulang kali.

Membuat Temp Table

```
WITH nama_tabel_baru as
(SELECT
        nama_kolom
        FROM nama_tabel)
SELECT nama_kolom, agregasi
FROM nama_tabel_baru
WHERE filter_tabel
GROUP BY nama_kolom
HAVING kondisi_agregasi
```

```
with top_spender as(
        select i.CustomerId, sum(Total) as spending
        from Invoice i
        group by CustomerId
        order by spending DESC
        limit 5)
select c.FirstName, c.LastName, ts.spending
from Customer c
join top_spender ts
on c.CustomerId = ts.customerID
```