DigitalSkola

LEARNING PROGRESS REVIEW
"WEEK 3"

BY : DREAM (KELOMPOK 1)

Data Realm Engineers And Maestros

Anggota Kelompok

- Afroh Fauziah
- Althaf Taqiyyah
- Andrew Bintang Pratama
- Andrew Suadnya
- Andi Rosilala

Database

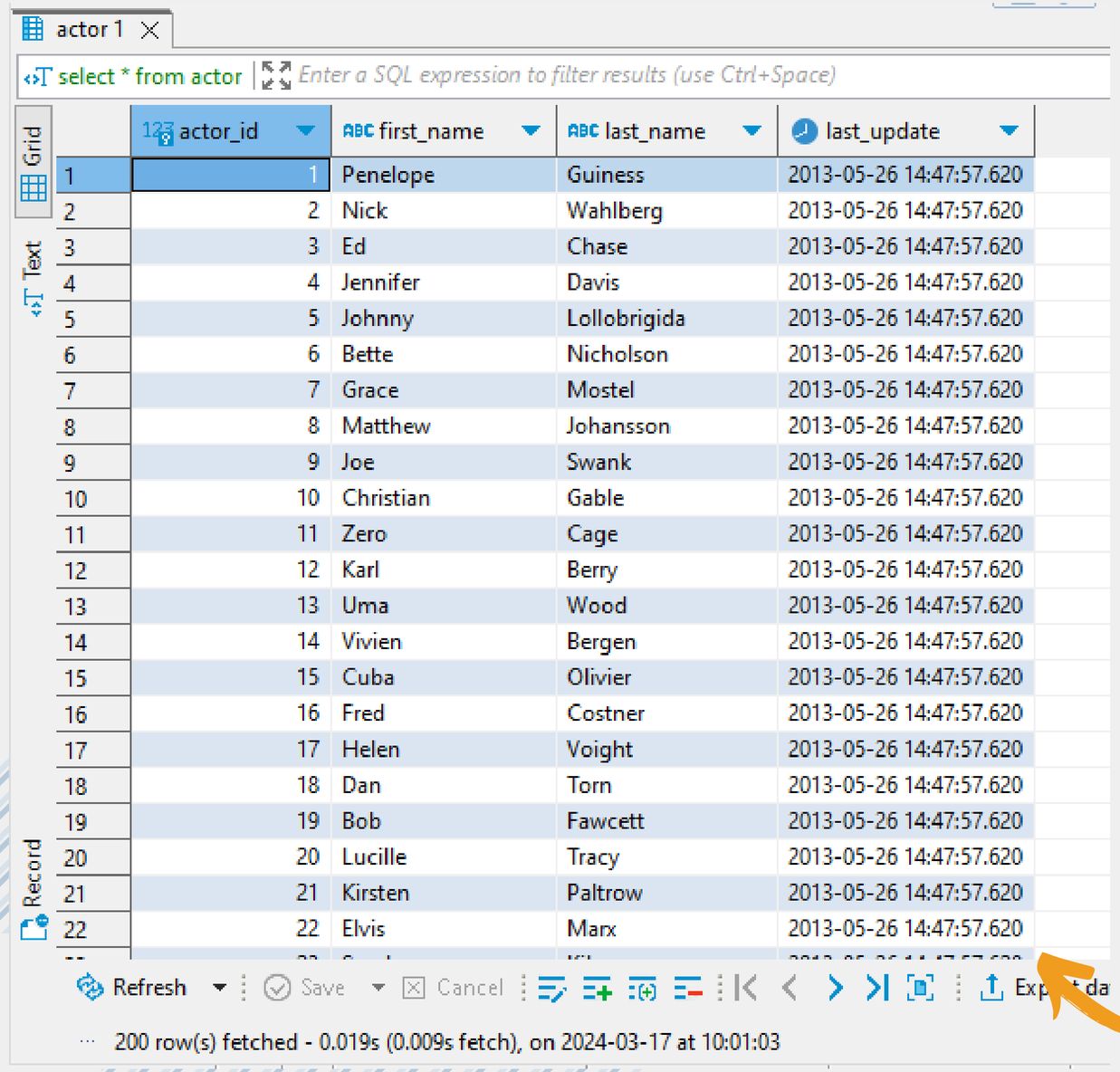
DigitalSkola
Uncover The World of Digital Skills with Us

Definisi Database

Sekumpulan informasi yang terorganisasi dan terstruktur. Umumnya tersimpan secara elektronik pada sebuah sistem komputer.

Database dirancang untuk menyimpan, mengambil. dan memanipulasi data ukuran besar secara efisien

Table, Row, & Column



	actor_id	first_name	last_name	last_update
1	1	Penelope	Guiness	2013-05-26 14:47:57.620
2	2	Nick	Wahlberg	2013-05-26 14:47:57.620
3	3	Ed	Chase	2013-05-26 14:47:57.620
4	4	Jennifer	Davis	2013-05-26 14:47:57.620
5	5	Johnny	Lollobrigida	2013-05-26 14:47:57.620
6	6	Bette	Nicholson	2013-05-26 14:47:57.620
7	7	Grace	Mostel	2013-05-26 14:47:57.620
8	8	Matthew	Johansson	2013-05-26 14:47:57.620
9	9	Joe	Swank	2013-05-26 14:47:57.620
10	10	Christian	Gable	2013-05-26 14:47:57.620
11	11	Zero	Cage	2013-05-26 14:47:57.620
12	12	Karl	Berry	2013-05-26 14:47:57.620
13	13	Uma	Wood	2013-05-26 14:47:57.620
14	14	Vivien	Bergen	2013-05-26 14:47:57.620
15	15	Cuba	Olivier	2013-05-26 14:47:57.620
16	16	Fred	Costner	2013-05-26 14:47:57.620
17	17	Helen	Voight	2013-05-26 14:47:57.620
18	18	Dan	Torn	2013-05-26 14:47:57.620
19	19	Bob	Fawcett	2013-05-26 14:47:57.620
20	20	Lucille	Tracy	2013-05-26 14:47:57.620
21	21	Kirsten	Paltrow	2013-05-26 14:47:57.620
22	22	Elvis	Marx	2013-05-26 14:47:57.620

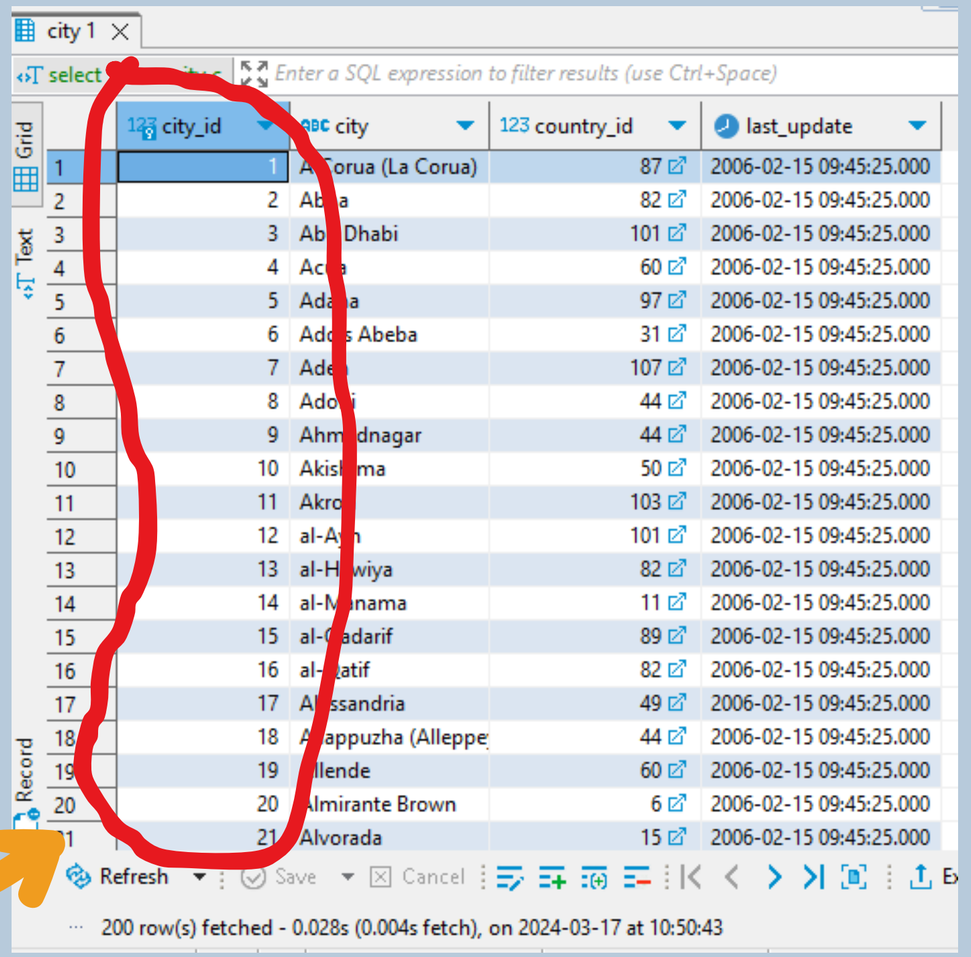
Table: objek fundamental dalam sebuah database yang berfungsi untuk menyimpan dan mengorganisasi data dalam bentuk baris (row) dan kolom (column).

Baris (row): data yang tersimpan dalam sebuah tabel. Setiap baris merupakan gabungan nilai dari setiap kolom.

Kolom (column): biasa diebut field. Elemen data berbentuk vertikal yang mewakili sifat data yang disimpan.

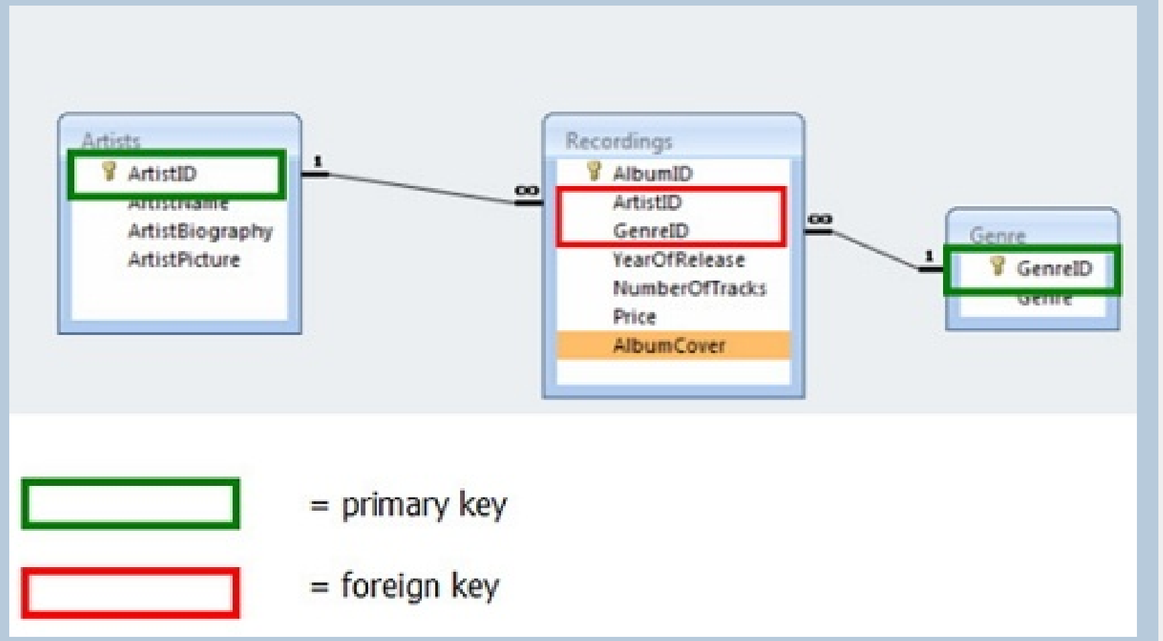
Primary Key

- PK merupakan ‘tanda pengenal’ yang ditetapkan untuk suatu tabel. PK harus merupakan atribut yang paling cocok dan paling dapat membedakan data-data yang ada di dalam tabel tersebut.
- Nilai yang digunakan dalam primary key tidak boleh NULL.
- Merupakan kombinasi antara UNIQUE dan NOT NULL. Hal itu menyebabkan tidak adanya nilai duplikat.



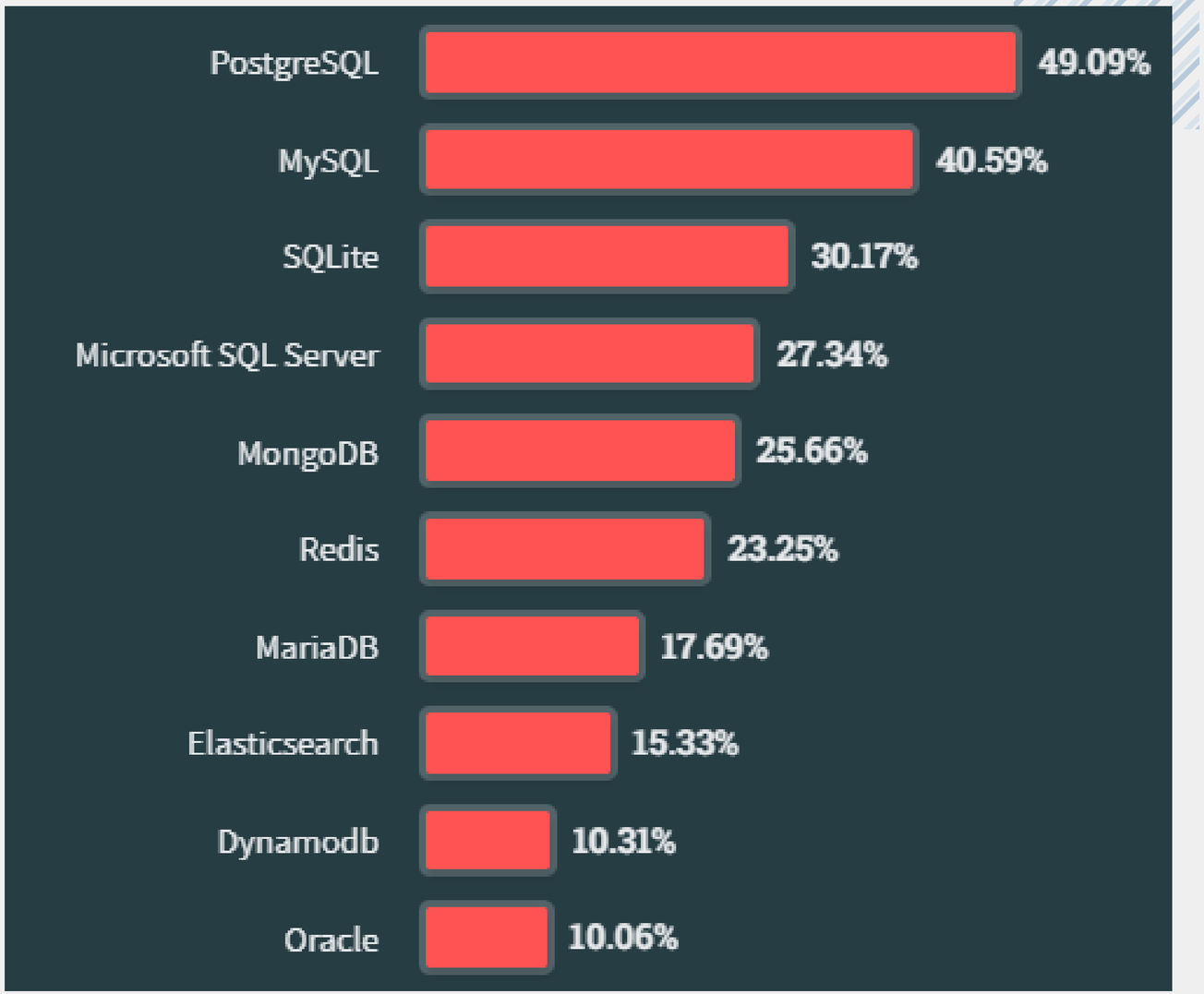
Foreign Key

- FK berfungsi sebagai penghubung antartabel.
- FK digunakan untuk menandakan hubungan tabel yang satu dengan yang lainnya. Yang dimana istilah ini dikenal sebagai parent dan child.
- FK dapat menerima nilai NUL dan tidak terdapat clustered index.
- Pada FK dapat menghapus suatu nilai yang berasal dari kolom foreign key tanpa mengganggu record yang lain,



DigitalSkola

Beberapa software RDBMS yang populer



https://survey.stackoverflow.co/2023/#most-popular-technologies-database-prof

DigitalSkol

SQL Statement

Data Definiation Languate

CREATE TABLE
ALTER TABLE
DROP TABLE

Data Query Language

SELECT, FROM
WHERE, ORDER BY,
GROUP BY, HAVING,
LIMIT, OFFSET

Data Manipulation

INSERT
UPDATE
DELETE
TRUNCATE

Data Control Language

GRANT
REVOKE

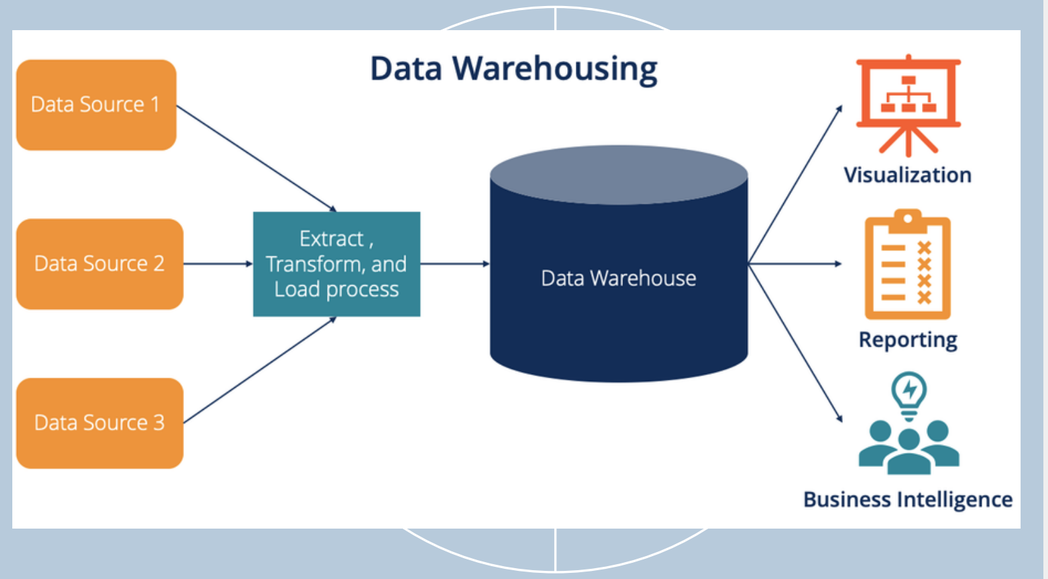
DigitalSkol

Datawarehouse and Data Modelling I

DigitalSkola
Uncover The World of Digital Skills with Us

Datawarehouse

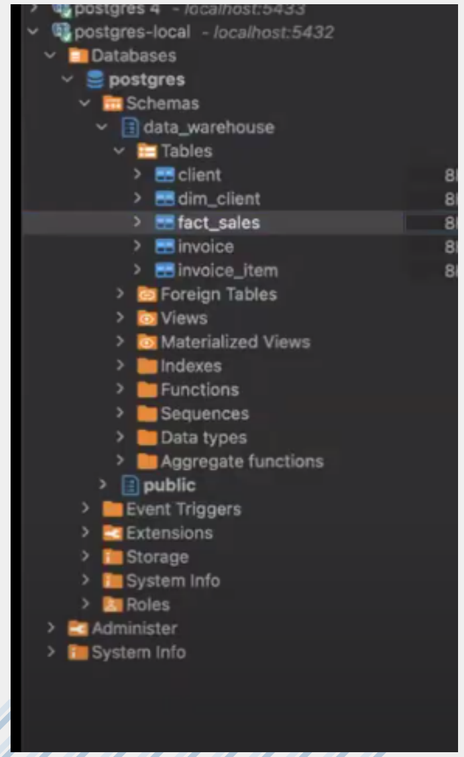
- Data source merupakan lokasi asli tempat data disimpan atau dihasilkan. Sumber data dapat berupa database, file, API, atau sistem lain yang menghasilkan data. Dalam konteks ini, ada tiga sumber data yang disebutkan, tetapi rincian spesifiknya tidak disediakan.
- Datawarehouse adalah teknologi yang digunakan untuk mengekstraksi, membersihkan, mengubah, memuat, menyimpan, dan mengelola data dari berbagai sumber sehingga dapat dianalisis dan diubah menjadi wawasan. Gudang data adalah pusat penyimpanan data yang dibuat dengan mengintegrasikan data dari satu atau lebih sumber terpisah.
- ETL adalah proses dalam penggunaan database dan khususnya dalam data warehousing. Proses ETL mengekstrak data mentah dari berbagai sumber, mengubahnya menjadi format yang dapat dimuat ke dalam gudang data, dan kemudian memuatnya ke dalam gudang data.



DigitalSkola

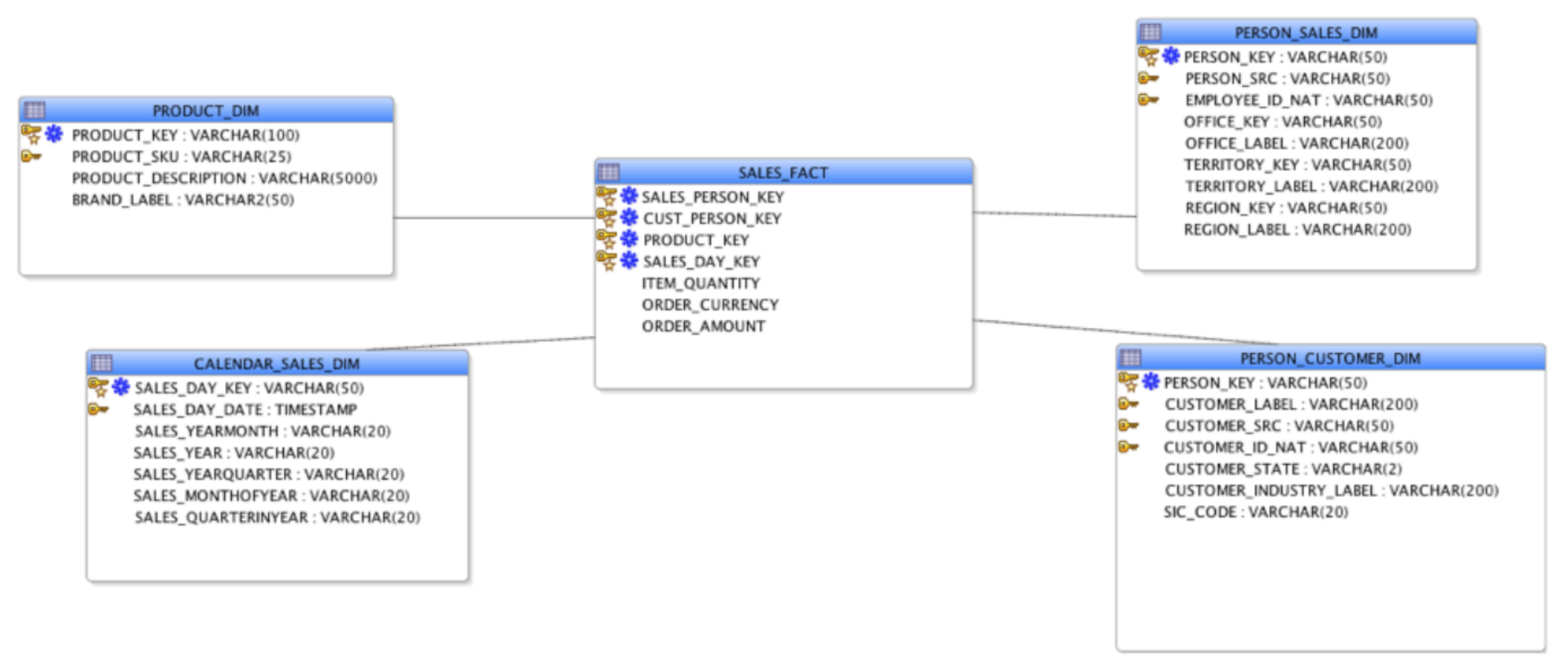
Building Datawarehouse

PostgreSQL



Membangun datawarehouse dengan PostgreSQL. PostgreSQL sendiri merupakan sistem manajemen database relasional (RDBMS) yang bersifat open source. Dikembangkan oleh Berkeley Computer Science Department, Sistem manajemen database ini dapat mengolah data dalam tabel yang memiliki relasi satu sama lain

Dimensional model



Dimensional model

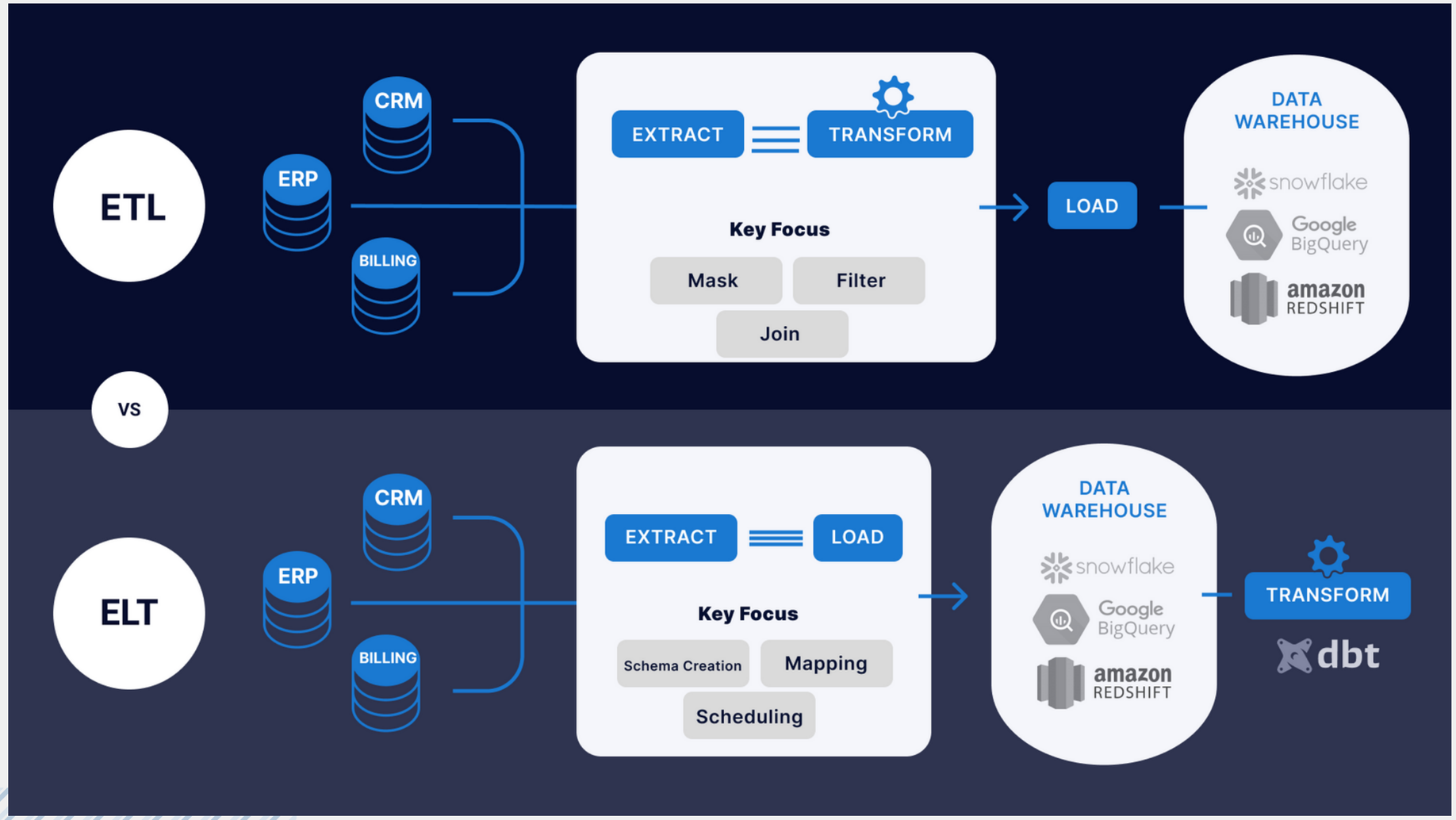
- **PRODUCT_DIM: Ini adalah tabel dimensi yang menyimpan informasi tentang produk.Kunci utamanya adalah PRODUCT_KEY dan atributnya PRODUCT_SKU, PRODUCT DESCRIPTION,dan BRAND LABEL.**

- **PERSON_SALES_DIM: Ini adalah tabel dimensi yang menyimpan informasi tentang tenaga penjualan. Kunci utama adalah PERSON_KEY dan memiliki atribut seperti EMPLOYEE_ID_NAT, OFFICE KEY, TERRITORY_KEY, REGION_KEY, dan labelnya masing-masing.**

- **CALENDAR_SALES_DIM: Ini adalah tabel dimensi yang menyimpan informasi tentang tanggal. Kunci utama adalah SALES_DAY_KEY dan memiliki atribut seperti SALES_DAY_DATE, SALES_YEARMONTH, SALES_YEAR, SALES_YEARQUARTER, SALES MONTHOFYEAR, dan SALES QUARTERINYEAR.**

- **PERSON CUSTOMER_DIM: Ini adalah tabel dimensi yang menyimpan informasi tentang pelanggan. Kunci utama adalah PERSON_KEY dan memiliki atribut seperti CUSTOMER_LABEL, CUSTOMER_SRC, CUSTOMER_ID_NAT, CUSTOMER_STATE, CUSTOMER INDUSTRY_LABEL, dan SIC CODE.**

ETL dan ELT


ETL
CRM
ERP
BILLING
EXTRACT
TRANSFORM
Key Focus
Mask
Filter
Join
LOAD
DATA WAREHOUSE
snowflake
Google BigQuery
amazon REDSHIFT
VS
ELT
CRM
ERP
BILLING
EXTRACT
LOAD
Key Focus
Schema Creation
Mapping
Scheduling
DATA WAREHOUSE
snowflake
Google BigQuery
amazon REDSHIFT
TRANSFORM
dbt

ETL dan ELT

ELT (Extract, Load, Transform) adalah metode data integration yang telah berkembang sebagai metode yang lebih efisien dengan perkembangan teknologi cloud. Dalam metode ELT, data ditambahkan langsung ke sistem target sebelum diterapkan transformasi. Transformasi dilakukan secara online menggunakan SQL pada sistem target, dan data tidak perlu diteruskan ke staging area sebelum ditransformasi.

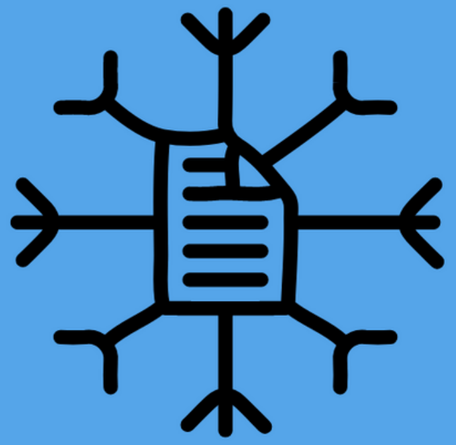
ETL (Extract, Transform, Load), pada sisi lain, adalah metode data integration yang lebih tradisional. Dalam metode ETL, data ditambahkan ke staging area sebelum diterapkan transformasi. Transformasi dilakukan di server pengolahan data terpisah, dan data perlu diteruskan ke staging area sebelum ditransformasi.

Datawarehouse and Data Modelling II

DigitalSkola
Uncover The World of Digital Skills with Us

Star Schema vs Snowflake Schema

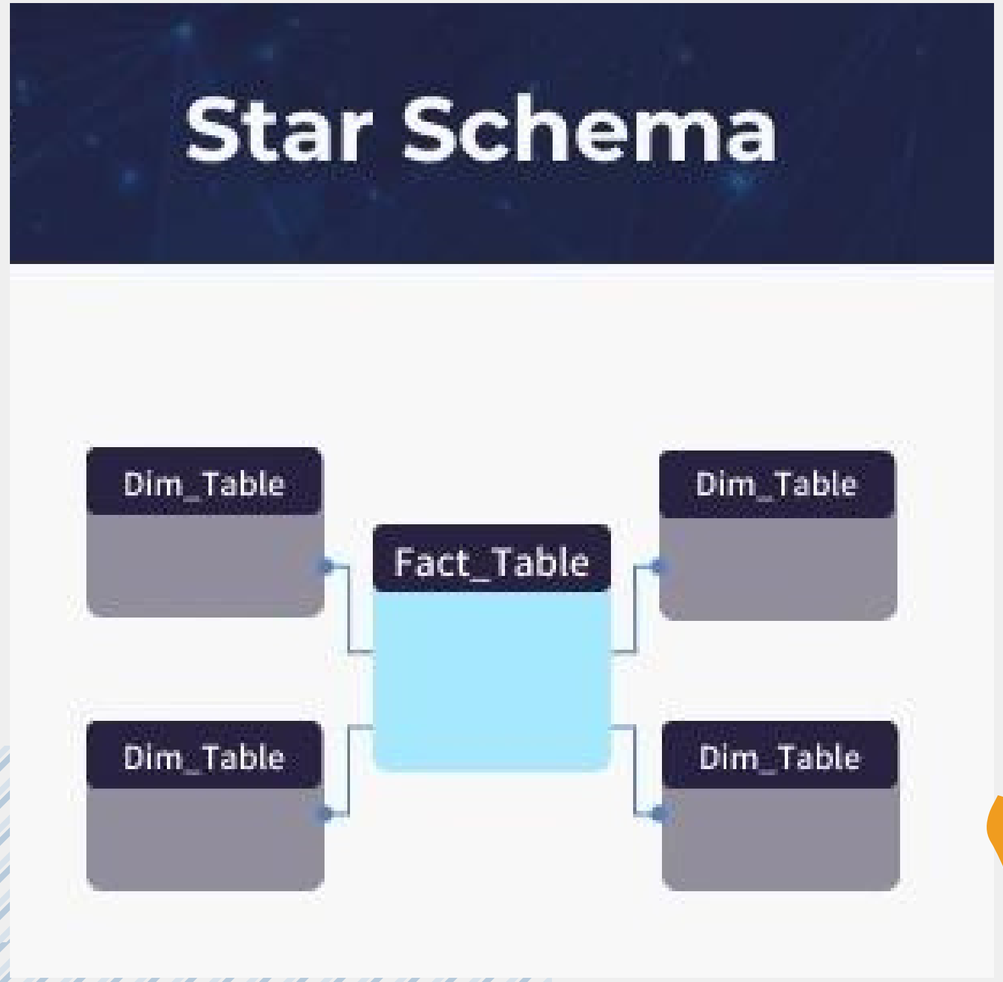
VS



Star Schema

Snowflake Schema

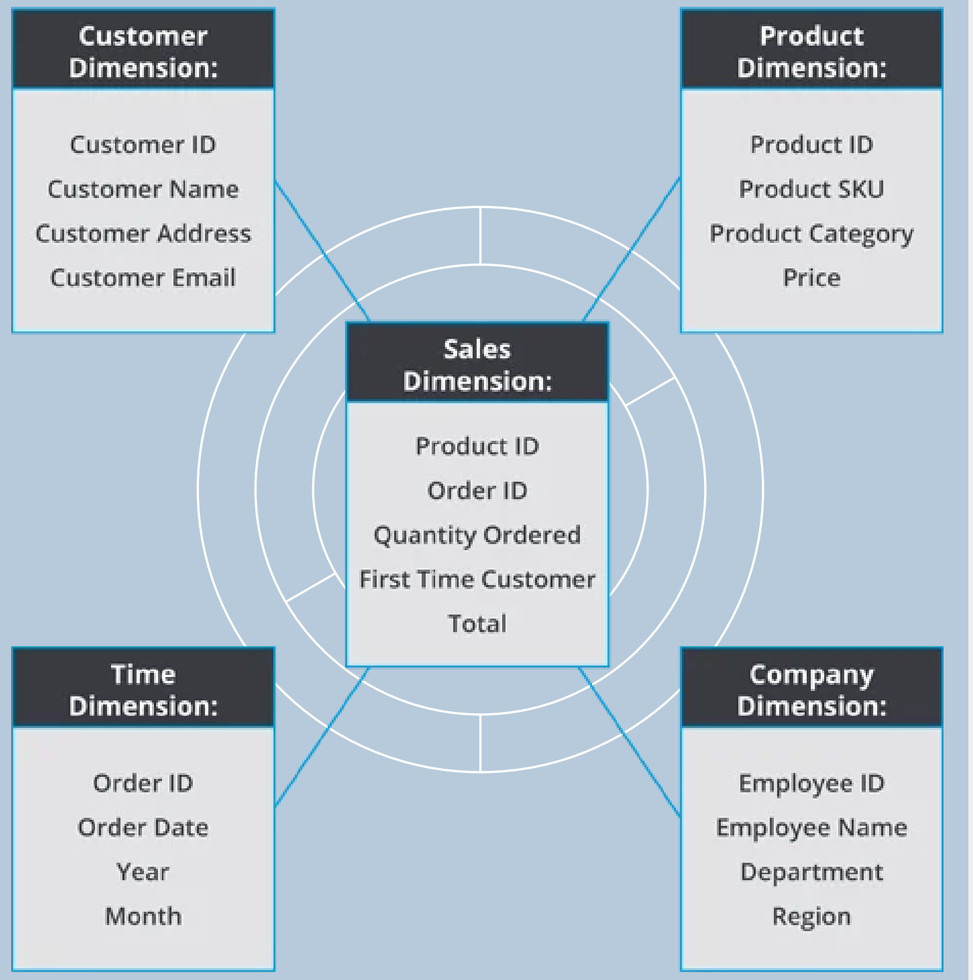
Star Schema



Star Schema adalah skema data warehouse yang menyerupai bintang, dengan tabel fakta di tengah dan tabel dimensi di sekitarnya. Skema ini dioptimalkan untuk query cepat dan analisis data, tetapi mungkin tidak ideal untuk hub[illegible]n data yang kompleks.

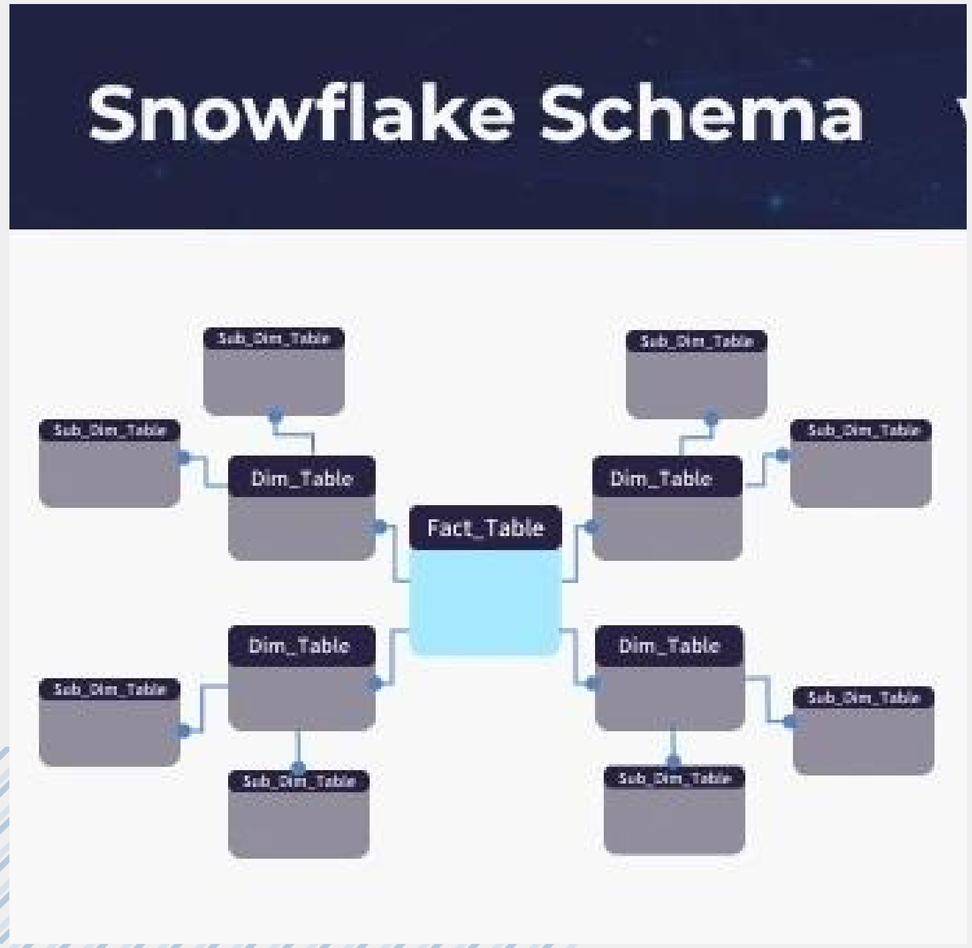
Karakteristik Star Schema

- **Struktur Sederhana:** Mirip bintang, tabel fakta di pusat, tabel dimensi di sekeliling.
- **Jumlah Tabel Sedikit:** Mempermudah desain dan pengelolaan data warehouse.
- **Tabel Dimensi Dinormalisasi:** Mempercepat query.
- **Foreign Key pada Tabel Dimensi:** Memudahkan penggabungan antar tabel.
- **Query Cepat:** Ideal untuk analisis data.
- **Fokus Agregasi:** Mendukung agregasi data.



DigitalSkola

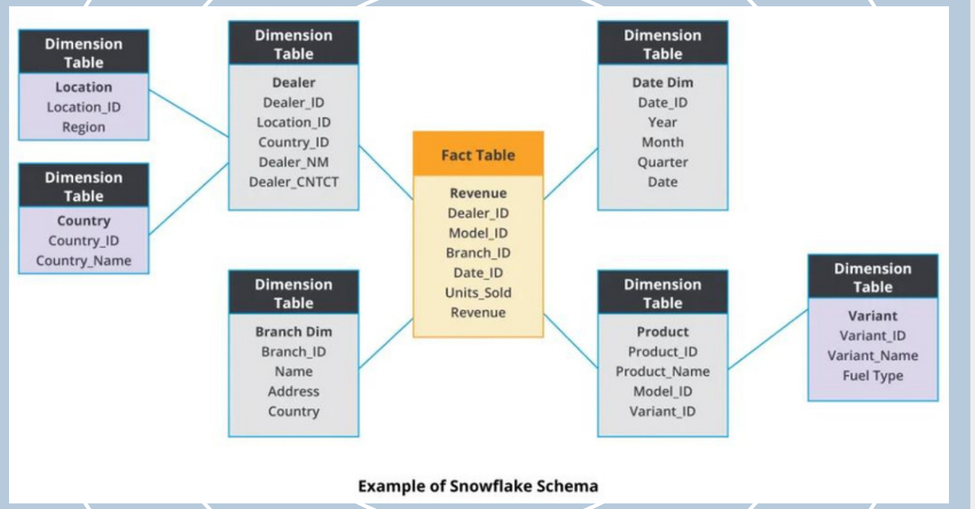
Snowflake Schema



Snowflake Schema merupakan versi normalized dari Star Schema, di mana tabel dimensi dipecah lebih lanjut untuk meningkatkan integritas data. Skema ini lebih cocok untuk hubungan data yang kompleks, tetapi membutuhkan query yang lebih kompleks.

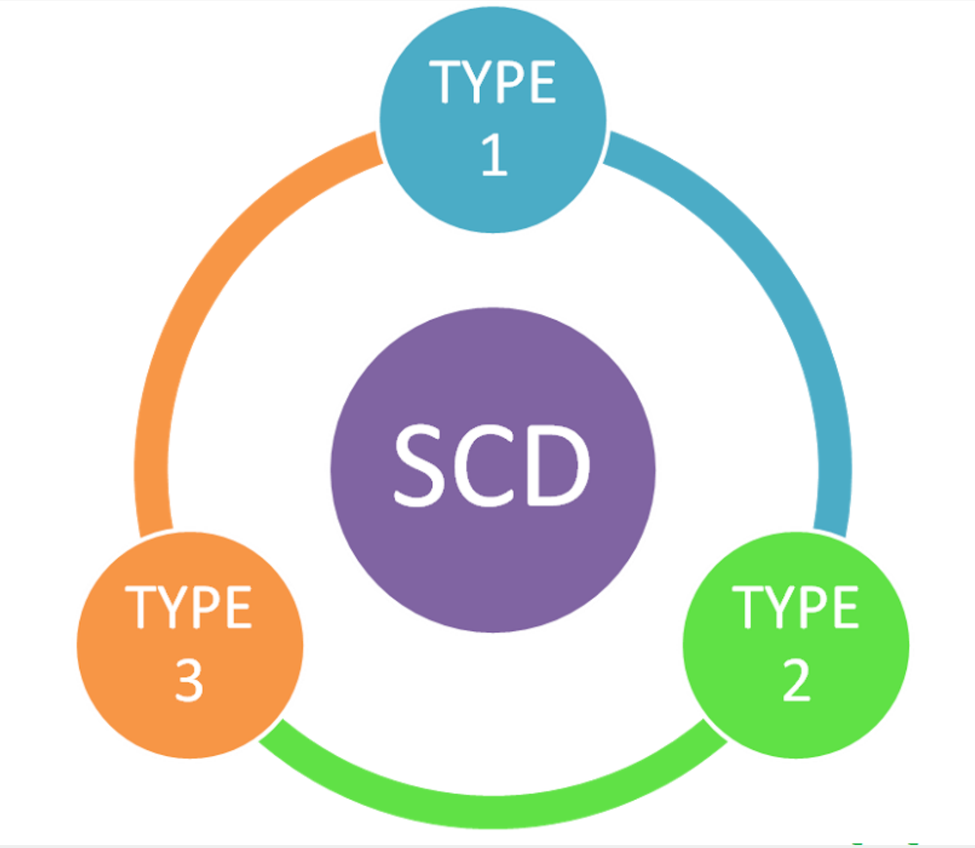
Karakteristik Snowflake Schema

- **Struktur Berbentuk Snowflake:** Mirip kepingan salju, tabel fakta di pusat, dikelilingi oleh sub-tabel dimensi yang dinormalisasi.
- **Tabel Dimensi Dinormalisasi Tinggi:** Meningkatkan integritas data dan granularitas.
- **Lebih Banyak Tabel:** Meningkatkan kompleksitas desain dan pengelolaan.
- **Query Lebih Kompleks:** Membutuhkan join yang lebih banyak.
- **Fleksibilitas:** Cocok untuk hubungan data kompleks.
- **Keamanan Data Lebih Baik:** Mengurangi redundansi dan meningkatkan kontrol akses.



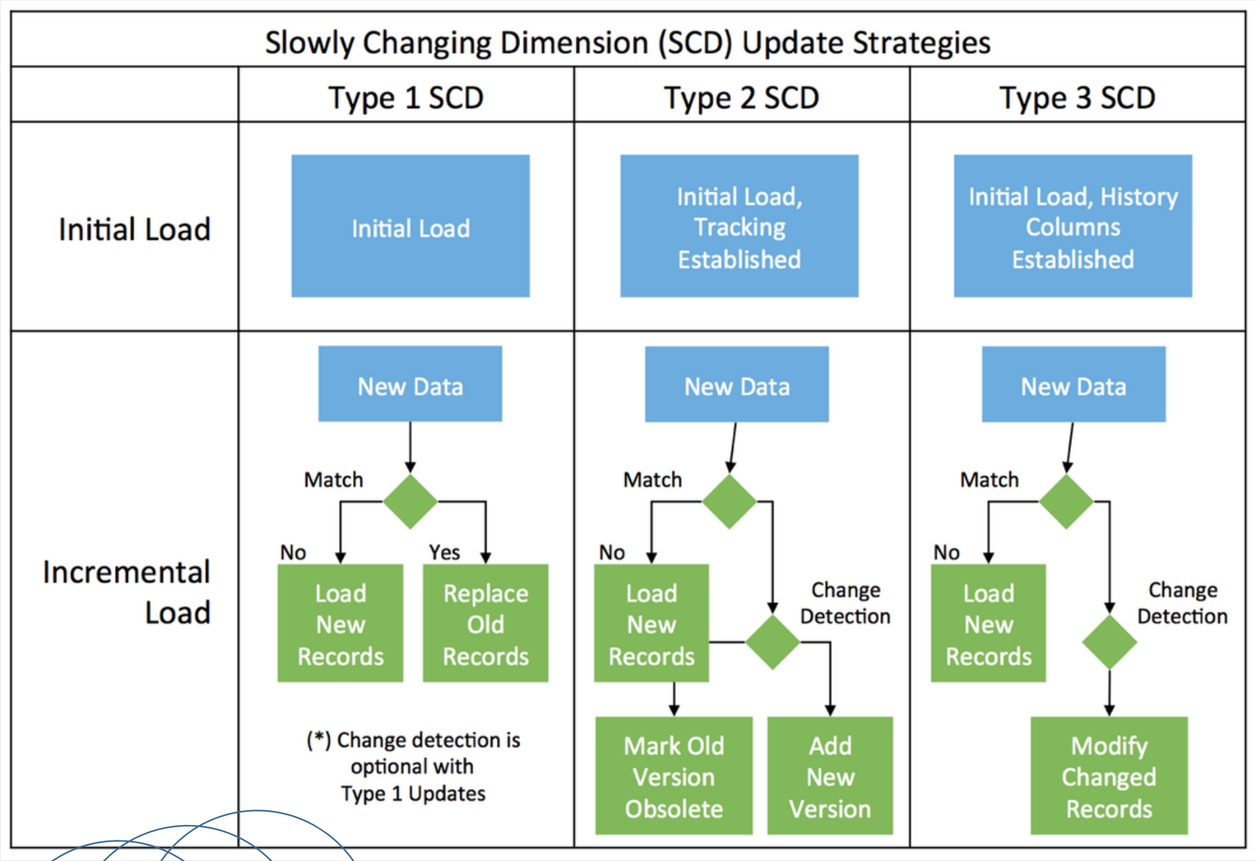
Example of Snowflake Schema

Slowly Changing Dimension (SCD)



Slowly Changing Dimension (SCD) adalah teknik untuk menangani perubahan data dalam tabel dimensi data warehouse seiring waktu. Dimensi data warehouse umumnya statis, seperti kategori produk, wilayah geografis, dan profil pelanggan. Namun, nilai dalam dimensi ini dapat berubah, seperti perubahan nama pelanggan, alamat, atau status pernikahan. SCD membantu mengelola perubahan ini dan menjaga integritas data dalam data warehouse.

3 Tipe Utama SCD



SCD Tipe 1: Overwrite

- Nilai lama ditimpa dengan nilai baru.
- Implementasi sederhana, tetapi kehilangan data historis.

SCD Tipe 2: Historis

- Baris baru ditambahkan dengan nilai baru, dan nilai lama disimpan.
- Melacak perubahan data historis, tetapi membutuhkan ruang penyimpanan lebih.

SCD Tipe 3: Atribut Baru

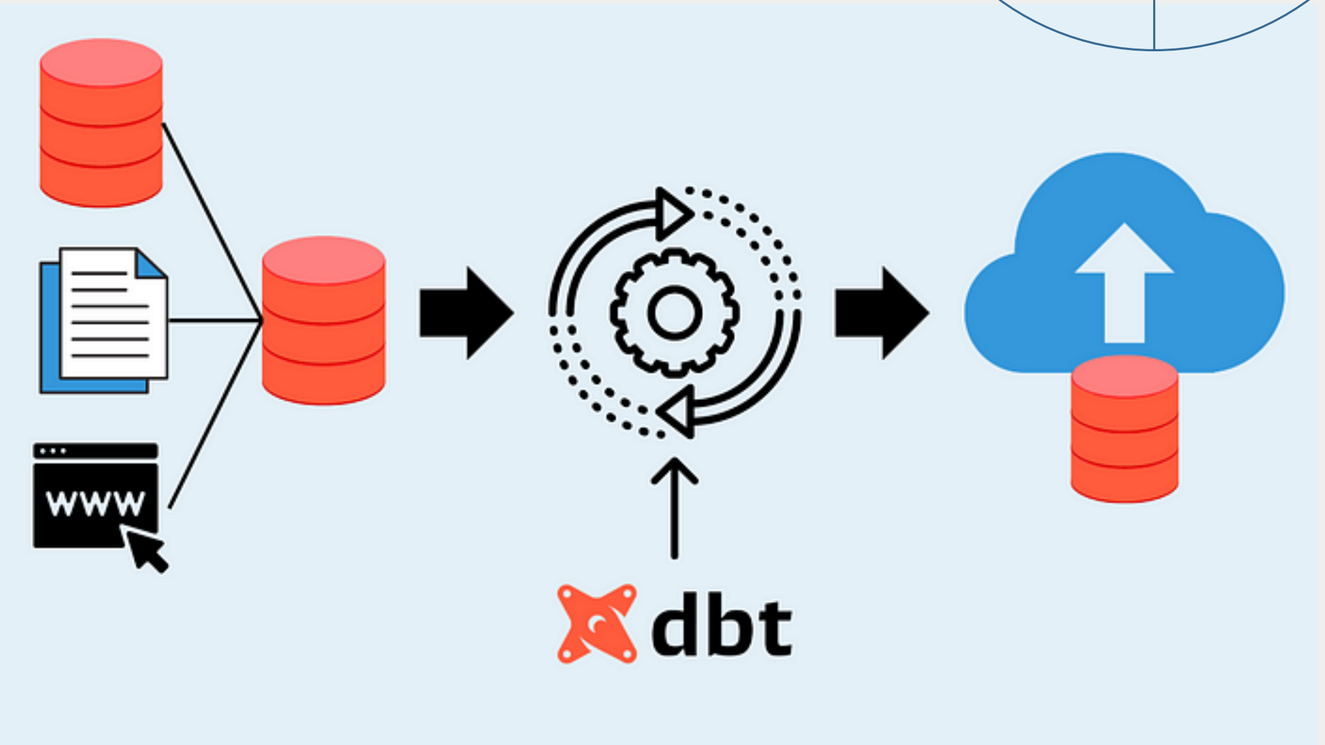
- Menyimpan beberapa versi data dengan kolom tambahan untuk melacak perubahan terakhir.
- Menyeimbangkan kinerja dan pelacakan historis.

DigitalSkola

Data Built Tool (DBT)

What is DBT?

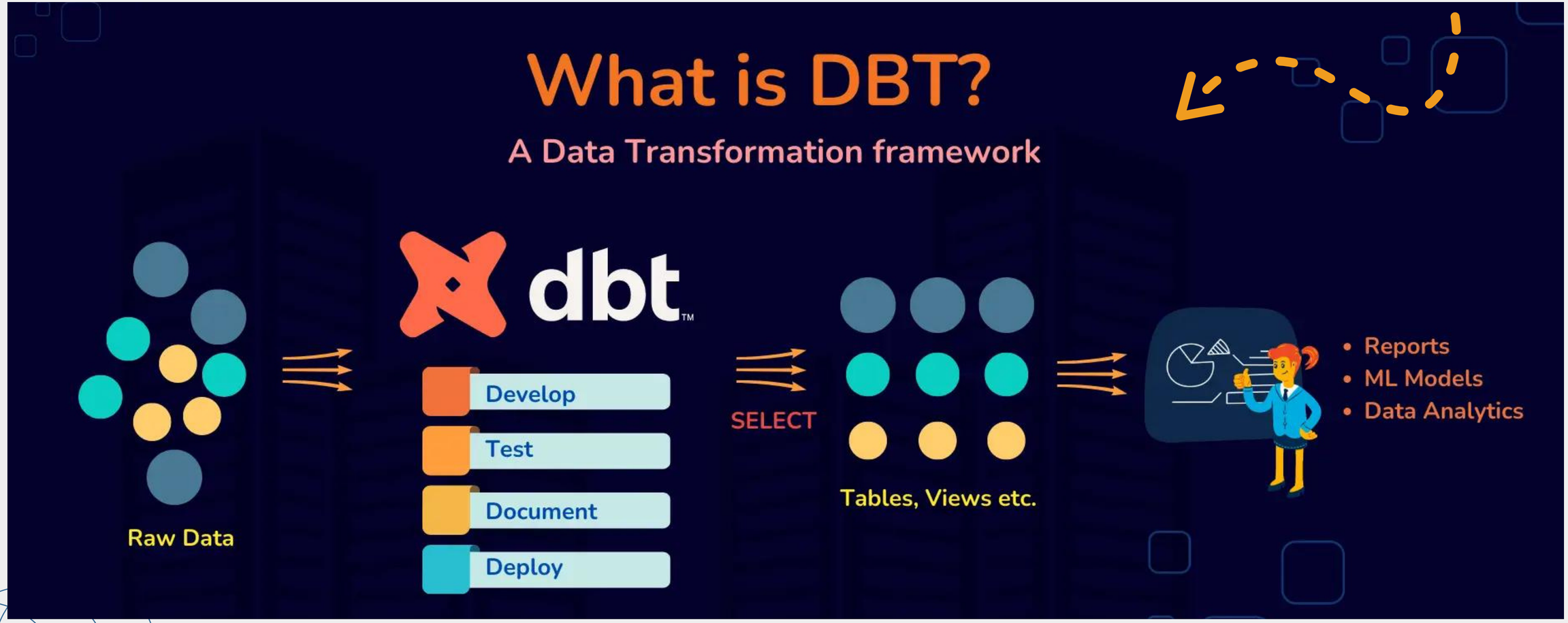
dbt adalah open-source tool yang membantu analis data dan developer membangun, mengelola, dan menjalankan transformasi data dalam data warehouse. DBT bertujuan untuk meningkatkan efisiensi dan keandalan proses ETL (Extract, Transform, Load) dengan menyediakan lingkungan yang terstruktur dan kolaboratif.



Manfaat Menggunakan DBT

- **Produktivitas yang Lebih Tinggi:** DBT memungkinkan penulisan kode SQL modular dan reusable, yang menghemat waktu dan mengurangi kesalahan.
- **Kolaborasi yang Lebih Baik:** DBT memfasilitasi kolaborasi antara analis data dan developer dengan menyediakan dokumentasi otomatis dan kontrol versi.
- **Keamanan dan Tata Kelola Data yang Ditingkatkan:** DBT mendukung praktik terbaik untuk keamanan data dan tata kelola dengan fitur seperti kontrol akses granular dan pelacakan lineage data.
- **Skalabilitas dan Fleksibilitas:** DBT dapat menangani data warehouse besar dan mendukung berbagai sumber data dan cloud platform.

DBT Workflow



Bagaimana DPT Bekerja?

Pengembangan model

Analis data dan developer menulis model SQL untuk mendefinisikan transformasi data.

Kompilasi dan Dokumentasi

DBT mengkompilasi model menjadi kode SQL standar dan menghasilkan dokumentasi otomatis.

Pengujian

Tes dijalankan untuk memvalidasi output model dan memastikan transformasi data berjalan dengan benar.

Eksekusi

DBT mengeksekusi model secara berurutan, sesuai dengan dependensi mereka, untuk melakukan transformasi data ke dalam data warehouse.

Project 2

DigitalSkola
Uncover The World of Digital Skills with Us

DBT Installation Part for windows

- Buat virtual environment untuk project ini. Note: di sini kami menamainya dengan ‘venvironment’
- Jika sudah aktifkan virtual environment yang sudah dibuat tadi

```
C:\Users\LENOVO\Stupen\postgresql-ddl-example>.\venvironment\Scripts\activate
```

- Kemudian install DBT pip install dbt-core==1.6.0 dbt-postgres==1.6.0

```
(venvironment) C:\Users\LENOVO\Stupen\postgresql-ddl-example>pip install dbt-core==1.6.0 dbt-postgres==1.6.0
Requirement already satisfied: dbt-core==1.6.0 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironment\l
Requirement already satisfied: dbt-postgres==1.6.0 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironme
Requirement already satisfied: agate~=1.7.0 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironment\lib\
=1.6.0) (1.7.1)
Requirement already satisfied: Jinja2~=3.1.2 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironment\lib
==1.6.0) (3.1.3)
Requirement already satisfied: mashumaro~=3.8.1 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironment\
maro[msgpack]~=3.8.1->dbt-core==1.6.0) (3.8.1)
Requirement already satisfied: logbook<1.6,>=1.5 in c:\users\lenovo\stupen\postgresql-ddl-example\venvironment
```

**ss yang digunakan menunjukkan bahwa dbt sudah terinstall sebelumnya*

DBT SETUP ‘init’

```
(env) C:\Users\afroh\postgresql-ddl-example>C:\Users\afroh\postgresql-ddl-example\env\Scripts\dbt.exe init
08:15:47  Running with dbt=1.6.0
08:15:47  [ConfigFolderDirectory]: Unable to parse dict {'dir': WindowsPath('C:/Users/afroh/.dbt')}
08:15:47  Creating dbt configuration folder at
Enter a name for your project (letters, digits, underscore): first_dw
08:16:14
Your new dbt project "first_dw" was created!

For more information on how to configure the profiles.yml file,
please consult the dbt documentation here:

  https://docs.getdbt.com/docs/configure-your-profile

One more thing:

Need help? Don't hesitate to reach out to us via GitHub issues or on Slack:

  https://community.getdbt.com/

Happy modeling!

08:16:14  Setting up your profile.
Which database would you like to use?
[1] postgres

(Don't see the one you want? https://docs.getdbt.com/docs/available-adapters)

Enter a number: 1
08:17:03  Profile first_dw written to C:\Users\afroh\.dbt\profiles.yml using target's sample configuration. Once up
dated, you'll be able to start developing with dbt.
```

nama project

ketik 1

Dengan mengubah direktori terminal ke direktori proyek, selanjutnya dilakukan inisialisasi proyek DBT dengan menjalankan dbt (copy path) init

DBT SETUP ‘debug’

```
09:55:21    database: postgres
09:55:21    schema: data_warehouse
09:55:21    connect_timeout: 10
09:55:21    role: None
09:55:21    search_path: None
09:55:21    keepalives_idle: 0
09:55:21    sslmode: None
09:55:21    sslcert: None
09:55:21    sslkey: None
09:55:21    sslrootcert: None
09:55:21    application_name: dbt
09:55:21    retries: 1
09:55:21  Registered adapter: postgres=1.6.0
09:55:21    Connection test: [OK connection ok]

09:55:21  All checks passed!
```

memeriksa konfigurasi dan koneksi dengan database

```
5   name: 'first_dw'
6   version: '1.0.0'
7   config-version: 2
8
9   # This setting configures which "profile" dbt uses for this project.
10  profile: 'postgres_dw'
```

PROBLEMS OUTPUT TERMINAL PORTS GITLENS DEBUG CONSOLE cmd

```
(env) C:\Users\afroh\postgresql-ddl-example>cd first_dw

(env) C:\Users\afroh\postgresql-ddl-example\first_dw>C:\Users\afroh\postgresql-ddl-example\env\Scripts\dbt.exe debug
07:20:53  Running with dbt=1.6.0
07:20:53  dbt version: 1.6.0
07:20:53  python version: 3.11.7
07:20:53  python path: C:\Users\afroh\postgresql-ddl-example\env\Scripts\python.exe
07:20:53  os info: Windows-10-10.0.19045-SP0
07:20:54  Using profiles dir at C:\Users\afroh\postgresql-ddl-example\first_dw
07:20:54  Using profiles.yml file at C:\Users\afroh\postgresql-ddl-example\first_dw\profiles.yml
07:20:54  Using dbt_project.yml file at C:\Users\afroh\postgresql-ddl-example\first_dw\dbt_project.yml
07:20:54  adapter type: postgres
07:20:54  adapter version: 1.6.0
07:20:54  Configuration:
07:20:54    profiles.yml file [OK found and valid]
07:20:54    dbt_project.yml file [OK found and valid]
07:20:54  Required dependencies:
07:20:54   - git [OK found]

07:20:54  Connection:
07:20:54    host: localhost
07:20:54    port: 5432
07:20:54    user: postgres
```

Untuk memvalidasi koneksi, ubah direktori ke root project (first_dw) dan run dengan copy path dbt debug

DBT SETUP
‘run & docs generate’

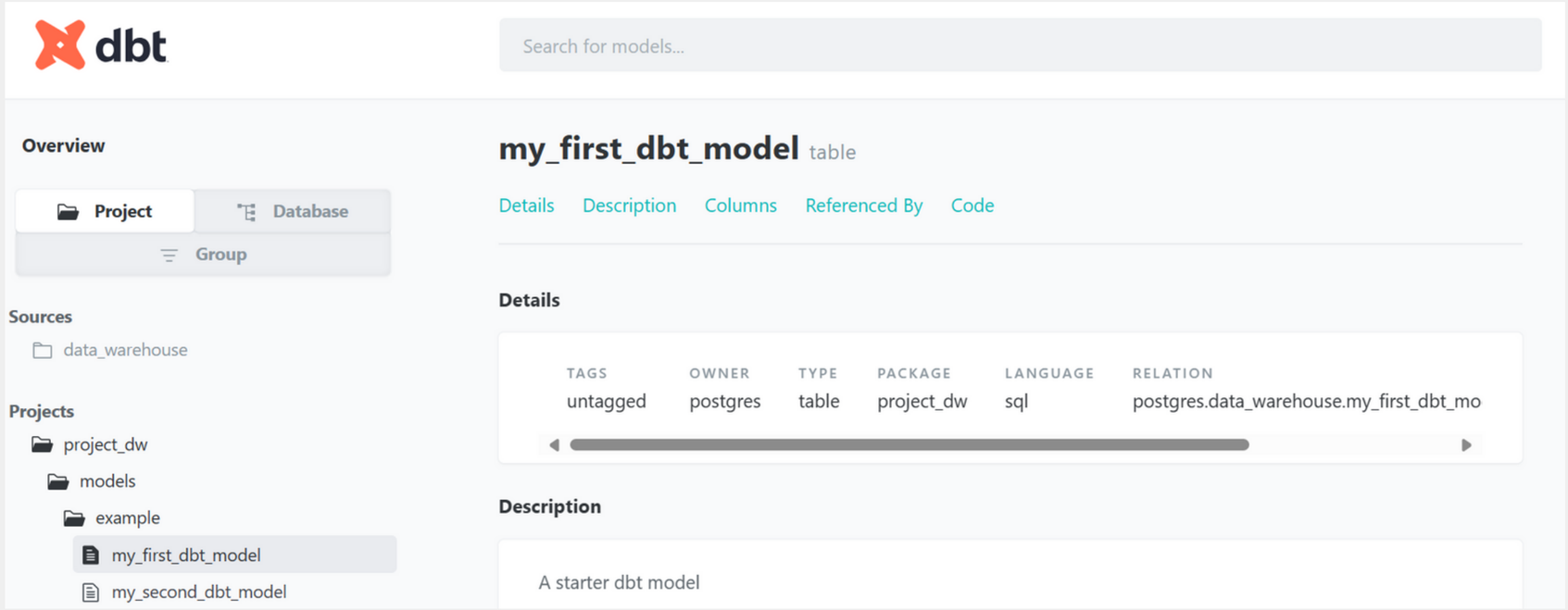
```
(venvironment) C:\Users\LENOVO\Stupen\postgresql-ddl-example\project_dw>dbt run -s my_first_dbt_model
09:56:06  Running with dbt=1.6.0
09:56:06  Registered adapter: postgres=1.6.0
09:56:06  Unable to do partial parsing because saved manifest not found. Starting full parse.
09:56:08  Found 2 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 349 macros, 0 groups, 0 semantic models
09:56:08
09:56:08  Concurrency: 1 threads (target='dev')
09:56:08
09:56:08  1 of 1 START sql table model data_warehouse.my_first_dbt_model ................. [RUN]
09:56:08  1 of 1 OK created sql table model data_warehouse.my_first_dbt_model ............ [SELECT 2 in 0.15s]
09:56:08
09:56:08  Finished running 1 table model in 0 hours 0 minutes and 0.37 seconds (0.37s).
09:56:08
09:56:08  Completed successfully
09:56:08
09:56:08  Done. PASS=1 WARN=0 ERROR=0 SKIP=0 TOTAL=1
```

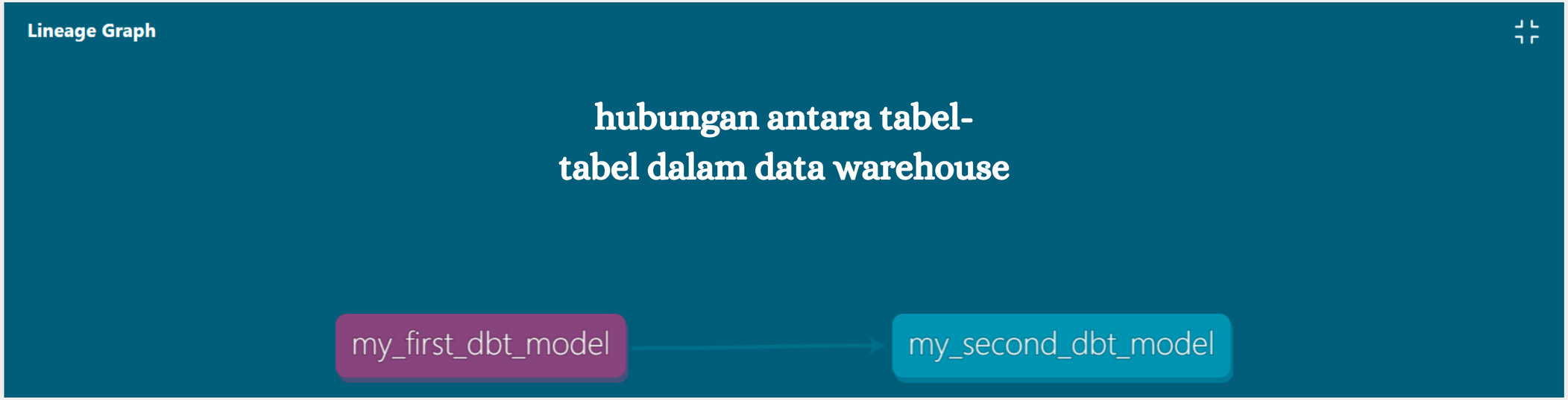
Jalankan dengan dbt run -s my_first_dbt_model

Jalankan first model yang ada pada folder example

```
(venvironment) C:\Users\LENOVO\Stupen\postgresql-ddl-example\project_dw>dbt docs generate
10:02:17  Running with dbt=1.6.0
10:02:17  Registered adapter: postgres=1.6.0
10:02:17  Found 2 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 349 macros, 0 groups, 0 semantic models
10:02:17
10:02:18  Concurrency: 1 threads (target='dev')
10:02:18
10:02:18  Building catalog
10:02:18  Catalog written to C:\Users\LENOVO\Stupen\postgresql-ddl-example\project_dw\target\catalog.json

(venvironment) C:\Users\LENOVO\Stupen\postgresql-ddl-example\project_dw>dbt docs serve
```

GUI DBT


dbt
Search for models...
Overview
Project
Database
Group
Sources
data_warehouse
Projects
project_dw
models
example
my_first_dbt_model
my_second_dbt_model
my_first_dbt_model table
Details
Description
Columns
Referenced By
Code
Details
TAGS
OWNER
TYPE
PACKAGE
LANGUAGE
RELATION
untagged
postgres
table
project_dw
sql
postgres.data_warehouse.my_first_dbt_mo
Description
A starter dbt model


Lineage Graph
hubungan antara tabel-
tabel dalam data warehouse
my_first_dbt_model
my_second_dbt_model

BUILD DATA WAREHOUSE 'DBT PROJECT'

DIM_CUSTOMER.SQL

dim_customer.sql U

project_dw > models > intermediate > dim_customer.sql

```sql
{{config(schema='dbt', materialized = 'table')}}

select
    customer_id,
    customer_name,
    email,
    phone_number
from {{source('data_warehouse', 'customer_dimension')}}
```

Tabel dimensi customer berisi kolom customer_id, customer_name, email, dan phone_number

Tabel dimensi time berisi kolom date_id, day_of_week, month, quarter, dan year.

DIM_TIME.SQL

dim_time.sql U

project_dw > models > intermediate > dim_time.sql

```sql
{{config(schema='dbt', materialized = 'table')}}

select
    date_id,
    day_of_week,
    month,
    quarter,
    year
from{{source('data_warehouse', 'time_dimension')}}
```

BUILD DATA WAREHOUSE 'DBT PROJECT'

DIM_PRODUCT.SQL

dim_product.sql U

project_dw > models > intermediate > dim_product.sql

```sql
{{config(schema='dbt', materialized = 'table')}}

select
    product_id,
    product_name,
    category,
    price
from {{source('data_warehouse', 'product_dimension')}}
```

Tabel dimensi product berisi kolom product_id, product_name, category, dan price

FACT_SALES.SQL

fact_sales.sql U

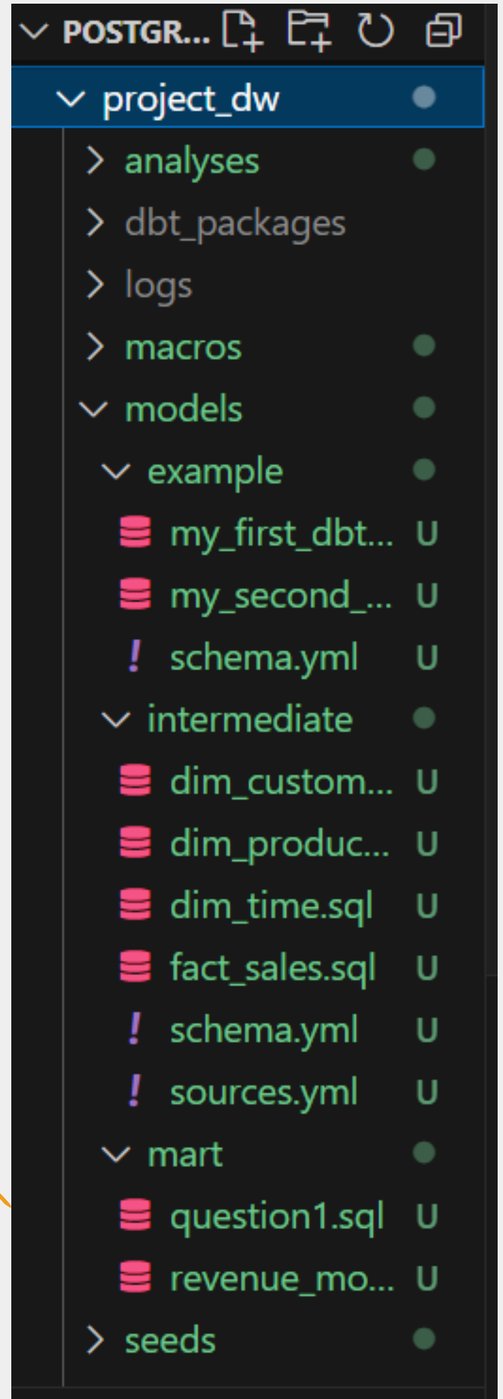
project_dw > models > intermediate > fact_sales.sql

```sql
{{config(schema='dbt', materialized = 'table')}}

select
    sale_id,
    customer_id,
    product_id,
    date_id,
    quantity,
    revenue
from {{source('data_warehouse', 'sales_fact')}}
```

Tabel fact sales berisi data transaksi penjualan dan memuat kolom sale_id, customer_id, product_id, date_id, quantity, serta revenue

MART DIRECTORY

TAMPILAN VSC



full file

Di dalam direktori model dengan mart_revenue_monthly.sql

revenue_monthly.sql U

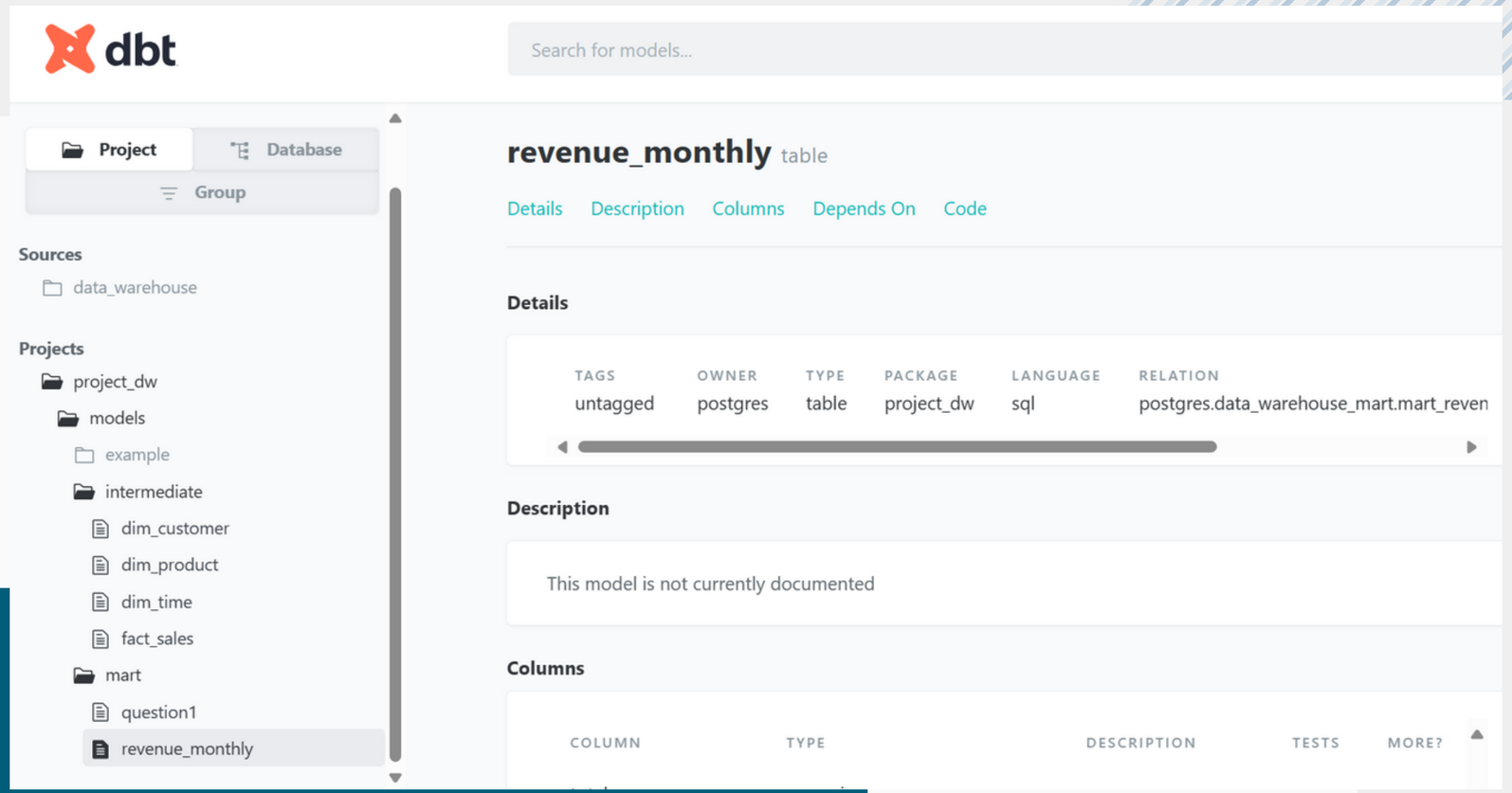
project_dw > models > mart > revenue_monthly.sql

```sql
{{config(schema='mart', materialized = 'table', alias='mart_revenue_monthly')}}


-- BUSINESS QUESTION
{{ config(schema='mart', materialized='table', alias='mart_revenue_monthly') }}

select
    SUM(quantity * revenue) as total_revenue
    , dt.month
    , dt.year
    , dp.product_name
from {{ ref('fact_sales') }} as fs
left join {{ ref('dim_time') }} as dt on fs.date_id = dt.date_id
left join {{ ref('dim_product') }} as dp on dp.product_id = fs.product_id
group by dt.month, dt.year, dp.product_name
```

DigitalSkola

DBT revenue monthly

Dihasilkan dari data yang dijoin dari tabel dimension product, dimension time, dan tabel fact sales.



TERIMA KASIH

Table of Content

What will We Learn Today?

1. Apa itu Relational Database SQL
2. Relational Database Design
3. Primary Key
4. Foreign Key
5. Data Types Dalam Relational Database SQL
6. SQL Command
7. CRUD

Session 14
(04/03/2024)

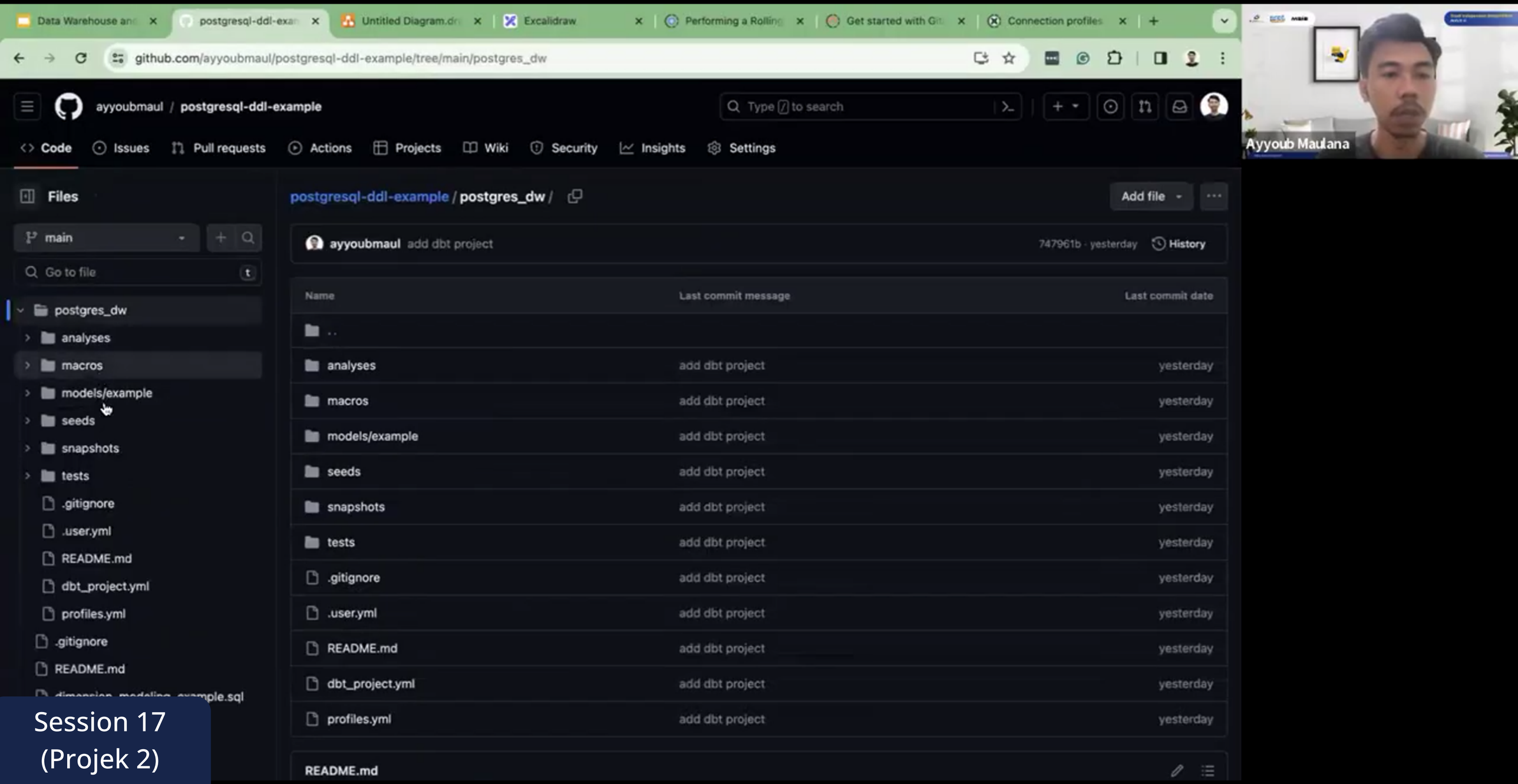
Talking Points

- Introduction to Data Warehouse
- Building Data Warehouse
- Data Warehouse Tools
- Data Modeling
- ETL/ELT Process

Session 15
(05/03/2024)


Data Warehouse and Data Modeling II
Session 16
(06/03/2024)


Ayyoub Maulana

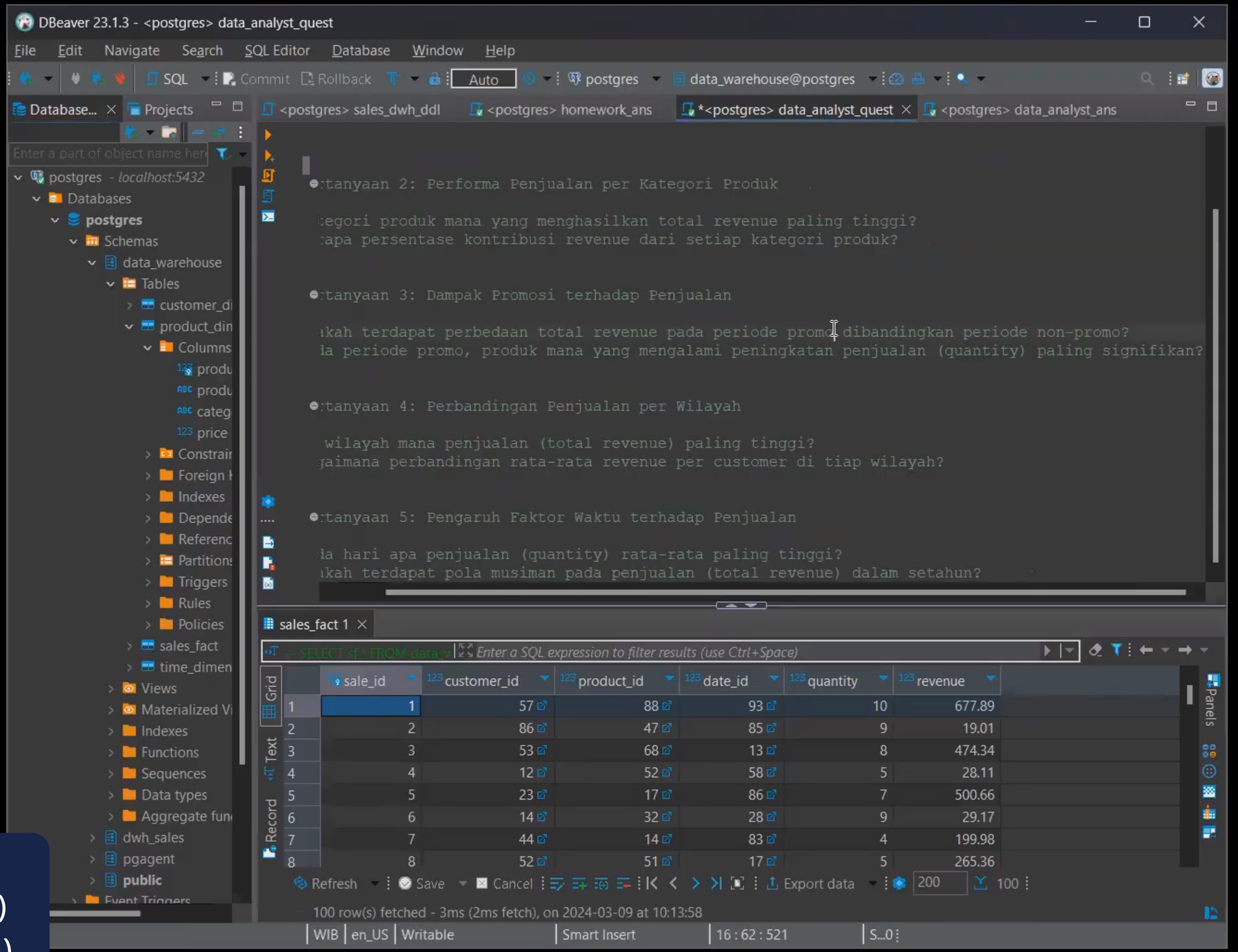


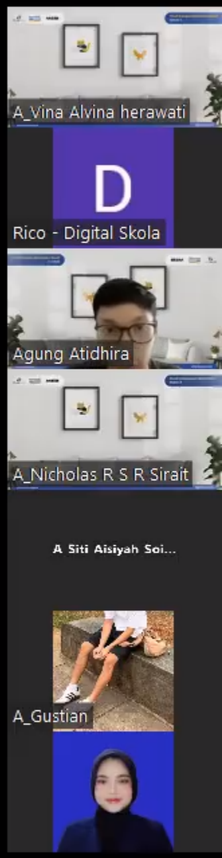
Session 17
(Projek 2)
(07/03/2024)

What is DWH?

- Apa saja *key components* nya?
- Perbedaan dengan traditional database (OLTP vs OLAP)
- Apa gunanya?
- Apa saja tipe-tipe DWH (EDW, ODS, Data Mart)

Session 18
(mentoring)
(08/03/2024)





Session 19
(mentoring)
(09/03/2024)