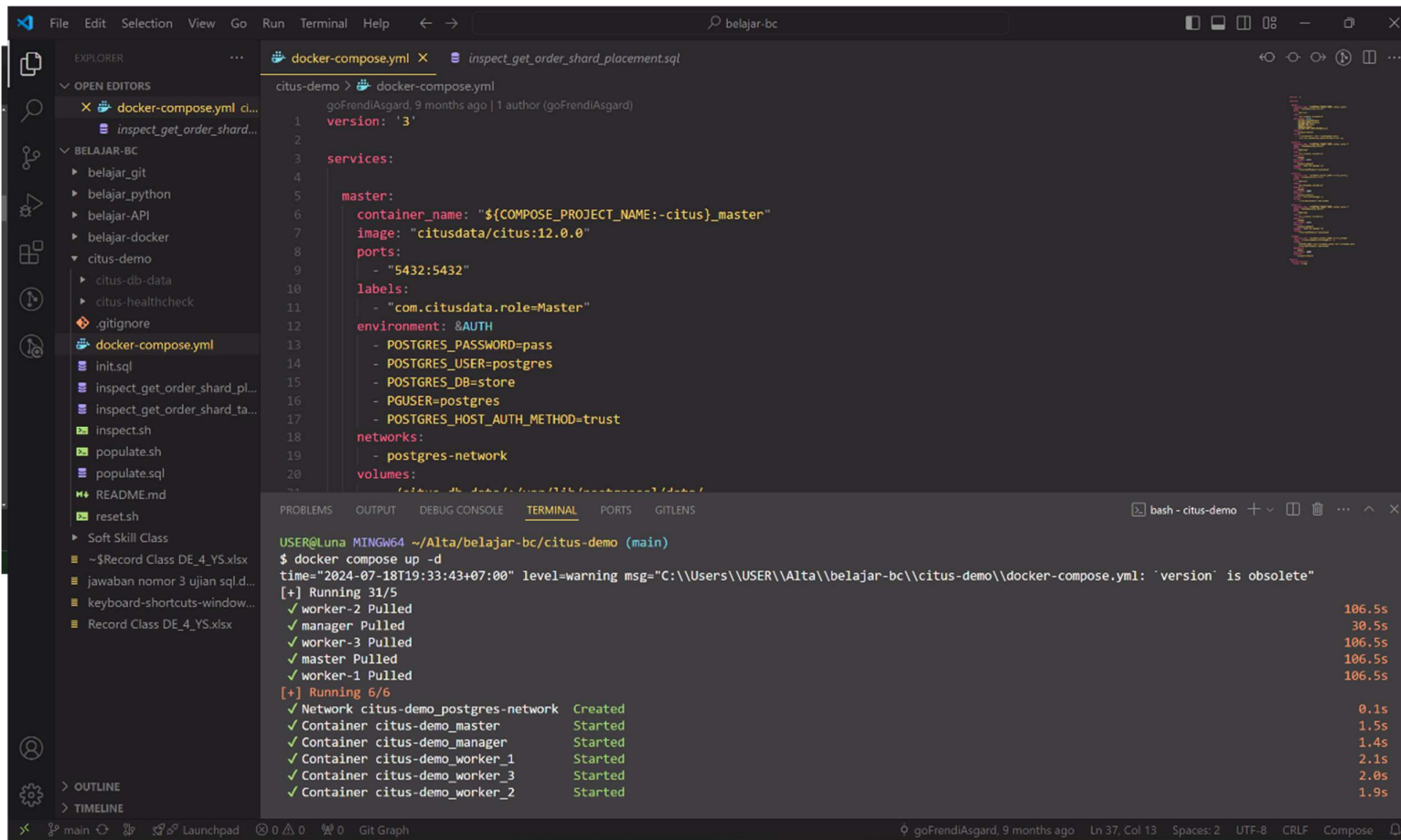


TASK 2 - Columnar Database

1. Jalankan Citus di komputer lokal dengan menggunakan docker compose

- Jalankan docker compose untuk menjalankan container-container citus-demo yang ada di docker-compose.yml



```
File Edit Selection View Go Run Terminal Help
citus-demo > docker-compose.yml
goFrendiAsgard, 9 months ago | 1 author (goFrendiAsgard)
version: '3'

services:
  master:
    container_name: "${COMPOSE_PROJECT_NAME:-citus}_master"
    image: "citusdata/citus:12.0.0"
    ports:
      - "5432:5432"
    labels:
      - "com.citusdata.role=Master"
    environment: &AUTH
      - POSTGRES_PASSWORD=pass
      - POSTGRES_USER=postgres
      - POSTGRES_DB=store
      - PGUSER=postgres
      - POSTGRES_HOST_AUTH_METHOD=trust
    networks:
      - postgres-network
    volumes:
      - citus_data:/var/lib/postgresql/data

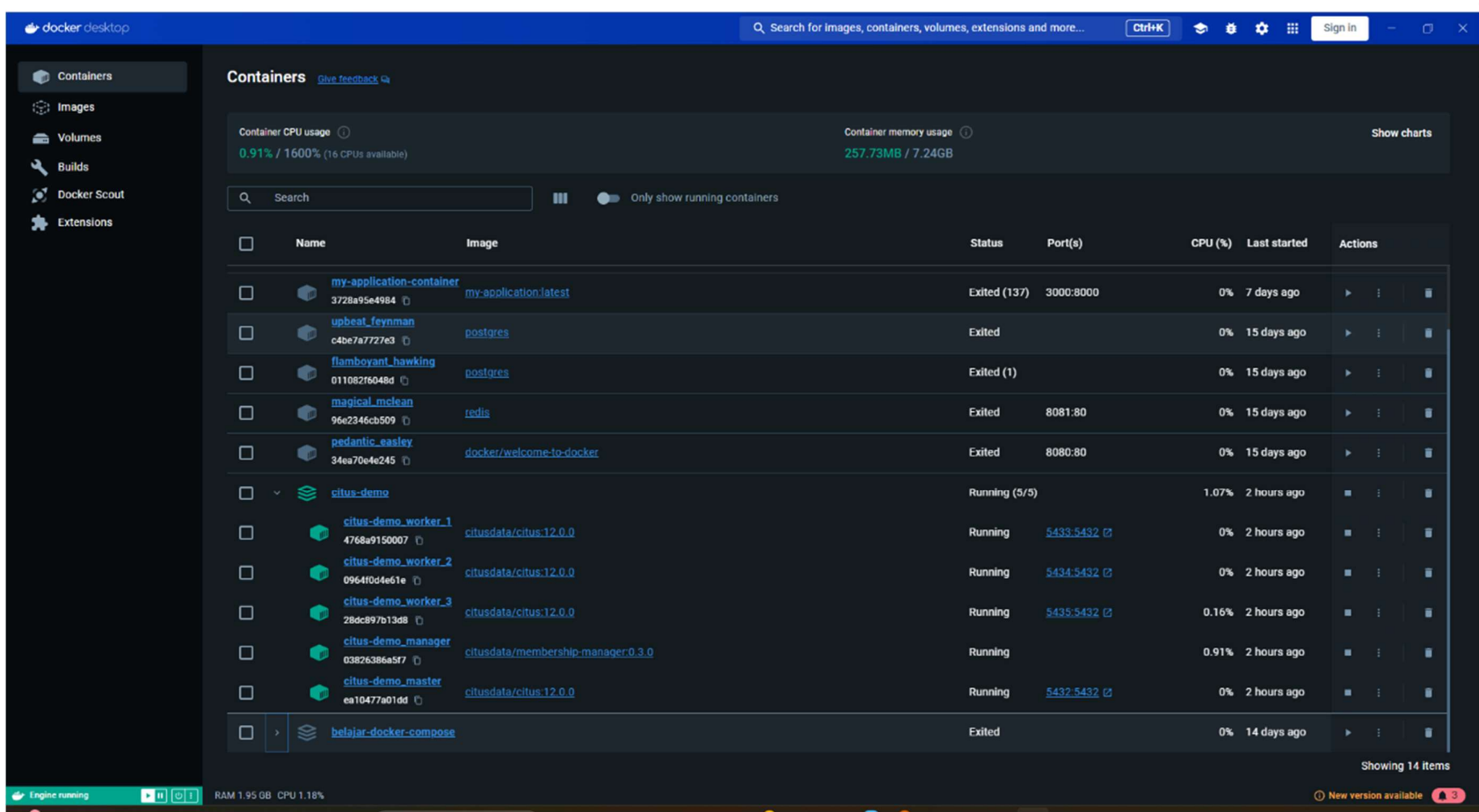
citus-demo > docker-compose up -d
time="2024-07-18T19:33:43+07:00" level=warning msg="C:\Users\USER\Alta\belajar-bc\citus-demo\docker-compose.yml: 'version' is obsolete"
[+] Running 31/5
  ✓ worker-2 Pulled                                106.5s
  ✓ manager Pulled                                30.5s
  ✓ worker-3 Pulled                                106.5s
  ✓ master Pulled                                  106.5s
  ✓ worker-1 Pulled                                106.5s
[+] Running 6/6
  ✓ Network citus-demo_postgres-network Created      0.1s
  ✓ Container citus-demo_master Started              1.5s
  ✓ Container citus-demo_manager Started             1.4s
  ✓ Container citus-demo_worker_1 Started            2.1s
  ✓ Container citus-demo_worker_3 Started            2.0s
  ✓ Container citus-demo_worker_2 Started            1.9s
```

DATA ENGINEER BATCH 4

Mentee: Yovina Silvia

Mentor: Bilal Benefit

b. Lalu cek di docker desktop untuk memastikan container citus-demo benar sudah bejalan



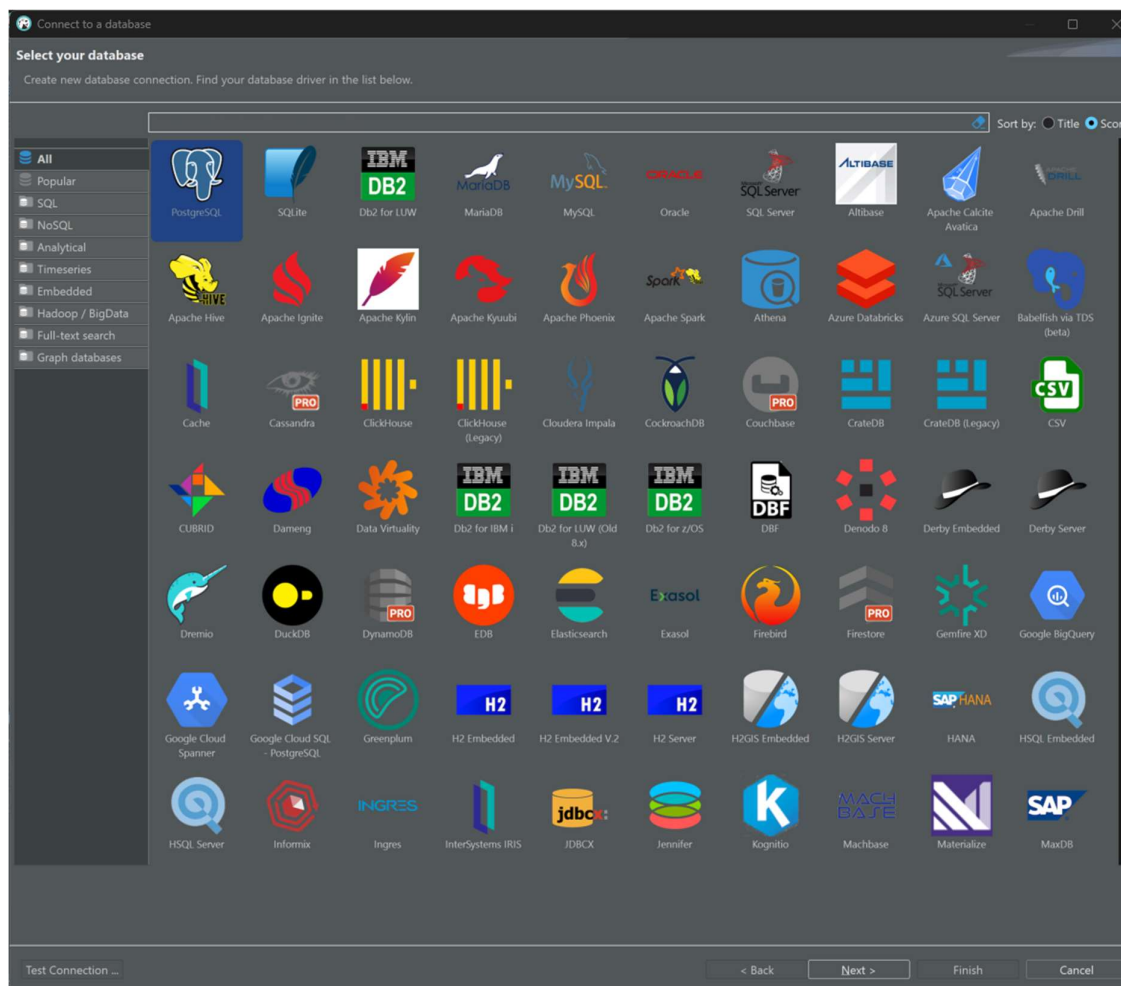
The screenshot shows the Docker Desktop interface. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Docker Scout, and Extensions. The main area is titled 'Containers' and displays a summary of container usage: 'Container CPU usage 0.91% / 1600% (16 CPUs available)' and 'Container memory usage 257.73MB / 7.24GB'. Below this is a search bar and a toggle for 'Only show running containers'. A table lists the containers, including their names, images, status, ports, CPU usage, and last started time. The 'citus-demo' group is expanded, showing five running containers: 'citus-demo_worker_1', 'citus-demo_worker_2', 'citus-demo_worker_3', 'citus-demo_manager', and 'citus-demo_master'. The 'belajar-docker-compose' container is also listed as 'Exited'.

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
my-application-container	my-application:latest	Exited (137)	3000:8000	0%	7 days ago	
upbeat_feynman	postgres	Exited		0%	15 days ago	
flamboyant_hawking	postgres	Exited (1)		0%	15 days ago	
magical_mclean	redis	Exited	8081:80	0%	15 days ago	
pedantic_easley	docker/welcome-to-docker	Exited	8080:80	0%	15 days ago	
citus-demo		Running (5/5)		1.07%	2 hours ago	
citus-demo_worker_1	citusdata/citus:12.0.0	Running	5433:5432	0%	2 hours ago	
citus-demo_worker_2	citusdata/citus:12.0.0	Running	5434:5432	0%	2 hours ago	
citus-demo_worker_3	citusdata/citus:12.0.0	Running	5435:5432	0.16%	2 hours ago	
citus-demo_manager	citusdata/membership-manager:0.3.0	Running		0.91%	2 hours ago	
citus-demo_master	citusdata/citus:12.0.0	Running	5432:5432	0%	2 hours ago	
belajar-docker-compose		Exited		0%	14 days ago	

2. Tuliskan perintah untuk membuat

- Tabel biasa
- Columnar tabel

➤ Buka postgresql pada dbeaver, lalu buat connection baru ke database postgresql dengan port master yang sudah kita jalankan di docker-compose.yml



DATA ENGINEER BATCH 4

Mentee: Yovina Silva

Mentor: Bilal Benefit



Connect to a database

Connection Settings
PostgreSQL connection settings

Main PostgreSQL Driver properties SSH SSL + Network configurations...

Server

Connect by: ☒ Host ☐ URL

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Port: 5432

Database: postgres ☐ Show all databases

Authentication

Authentication: Database Native

Username: store

Password: ☒ Save password

Advanced

Session role: Local Client: PostgreSQL 16

① Connection variables information Database documentation Connection details (name, type, ...)

Driver name: PostgreSQL Driver Settings Driver license

Test Connection ... < Back Next > Finish Cancel

- Create table `events_columnar` dengan using `columnar` dan buat table `events_row` sebagai table biasa

```
CREATE TABLE events_columnar (
  device_id bigint,
  event_id bigserial,
  event_time timestamptz default now(),
  data jsonb not null
)
USING columnar;
--insert some data
INSERT INTO events_columnar (device_id, data)
SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d;

--create a row-based table to compare
CREATE TABLE events_row AS SELECT * FROM events_columnar;

--select * from events_columnar
--select * from events_row
```

- **events_columnar:** Tabel ini dirancang untuk menyimpan data peristiwa menggunakan format penyimpanan kolom.
- **Kolom-kolom:**
 - `device_id` `bigint`: Pengidentifikasi unik untuk perangkat.
 - `event_id` `bigserial`: Pengidentifikasi unik untuk peristiwa, diurutkan secara otomatis.
 - `event_time` `timestamptz default now()`: Waktu terjadinya peristiwa, dengan nilai default waktu saat ini.
 - `data` `jsonb not null`: Kolom JSONB (binary JSON) untuk menyimpan data peristiwa dalam format JSON.
- **USING columnar:** Menentukan bahwa tabel ini harus menggunakan format penyimpanan kolom, yang bisa lebih efisien untuk jenis kueri tertentu.
- **INSERT INTO events_columnar (device_id, data):** Menyisipkan data ke dalam tabel `events_columnar`.
- **SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d:** Menghasilkan dan menyisipkan 100.000 baris data dengan `device_id` dari 1 hingga 100.000, dan data JSON `{"hello": "columnar"}`.
- **CREATE TABLE events_row AS SELECT * FROM events_columnar:** Membuat tabel baru `events_row` yang merupakan salinan dari tabel `events_columnar`, tetapi menggunakan format penyimpanan baris tradisional.

DATA ENGINEER BATCH 4

Mentee: Yovina Silvia

Mentor: Bilal Benefit

```
CREATE TABLE events_columnar (  
  device_id bigint,  
  event_id bigserial,  
  event_time timestamptz default now(),  
  data jsonb not null  
)  
USING columnar;  
--insert some data  
INSERT INTO events_columnar (device_id, data)  
SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d;  
--create a row-based table to compare  
CREATE TABLE events_row AS SELECT * FROM events_columnar;
```

events_columnar 1 x

Enter a SQL expression to filter results (use Ctrl+Space)

Grid	device_id	event_id	event_time	data
1	1	1	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
2	2	2	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
3	3	3	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
4	4	4	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
5	5	5	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
6	6	6	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
7	7	7	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
8	8	8	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
9	9	9	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
10	10	10	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
11	11	11	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
12	12	12	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
13	13	13	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
14	14	14	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
15	15	15	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
16	16	16	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
17	17	17	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
18	18	18	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
19	19	19	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
20	20	20	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
21	21	21	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
22	22	22	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
23	23	23	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
24	24	24	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
25	25	25	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
26	26	26	2024-07-19 16:38:48.863 +0700	("hello": "columnar")
27	27	27	2024-07-19 16:38:48.863 +0700	("hello": "columnar")

Refresh Save Cancel Export data 200 200+ 200 row(s) fetched - 0.064s (t)

WIB en Writable Smart Insert 13 : 58 : 392 Sel: 0 | 0

DATA ENGINEER BATCH 4

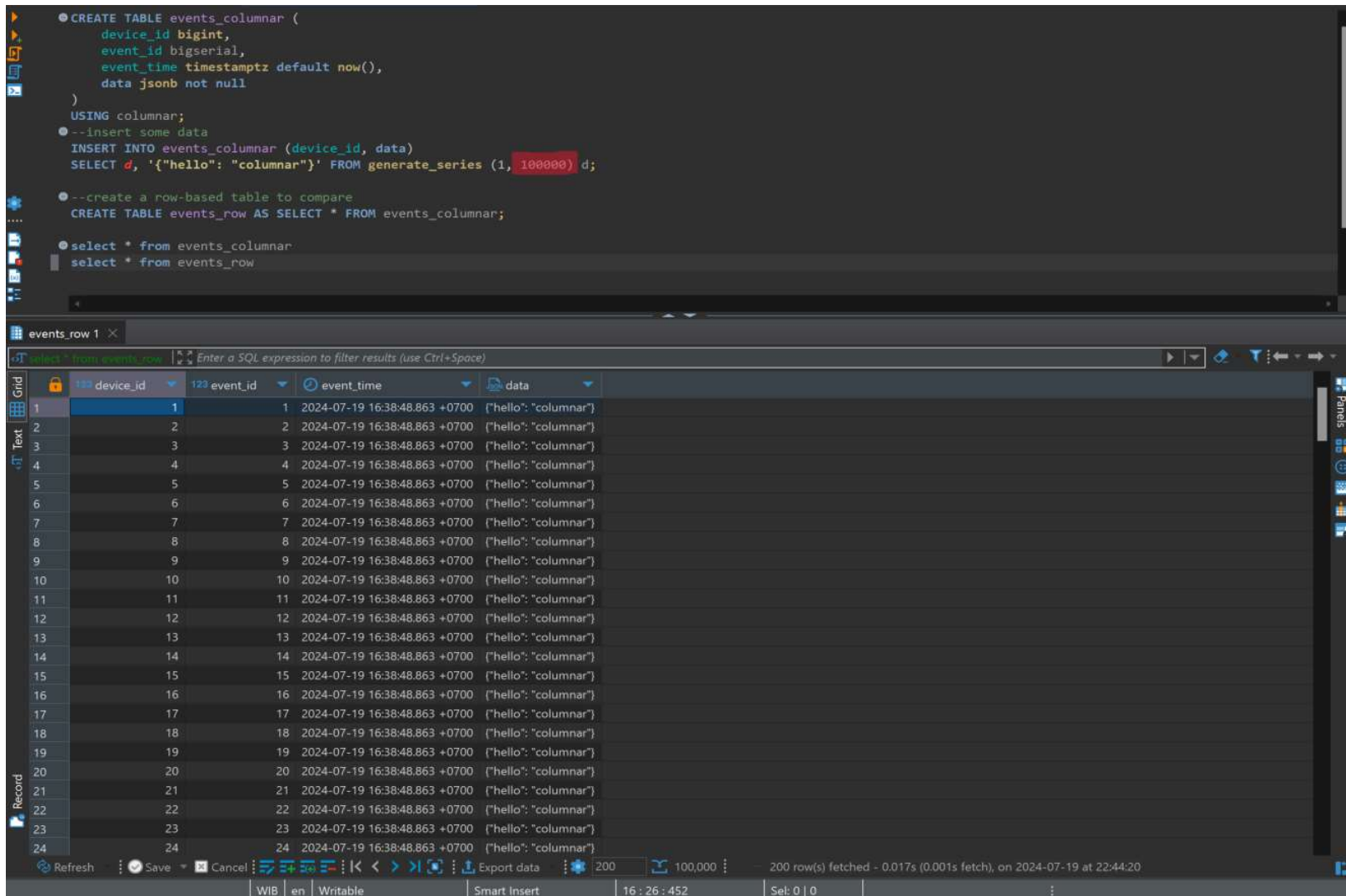
Mentee: Yovina Silvia

Mentor: Bilal Benefit

3. Masukkan 100 baris data ke dalam tabel biasa dan tabel columnar

Baris yang dimasukkan 100.000 agar terlihat perbedaan ukuran storage antara table columnar dengan table row biasa

```
INSERT INTO events_columnar (device_id, data)
SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d;
```



The screenshot shows a database interface with a SQL editor and a table view. The SQL editor contains the following code:

```
CREATE TABLE events_columnar (
  device_id bigint,
  event_id bigserial,
  event_time timestampz default now(),
  data jsonb not null
)
USING columnar;
--insert some data
INSERT INTO events_columnar (device_id, data)
SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d;
--create a row-based table to compare
CREATE TABLE events_row AS SELECT * FROM events_columnar;
select * from events_columnar;
select * from events_row;
```

The table view shows the first 24 rows of the `events_row` table. The columns are `device_id`, `event_id`, `event_time`, and `data`. The data is as follows:

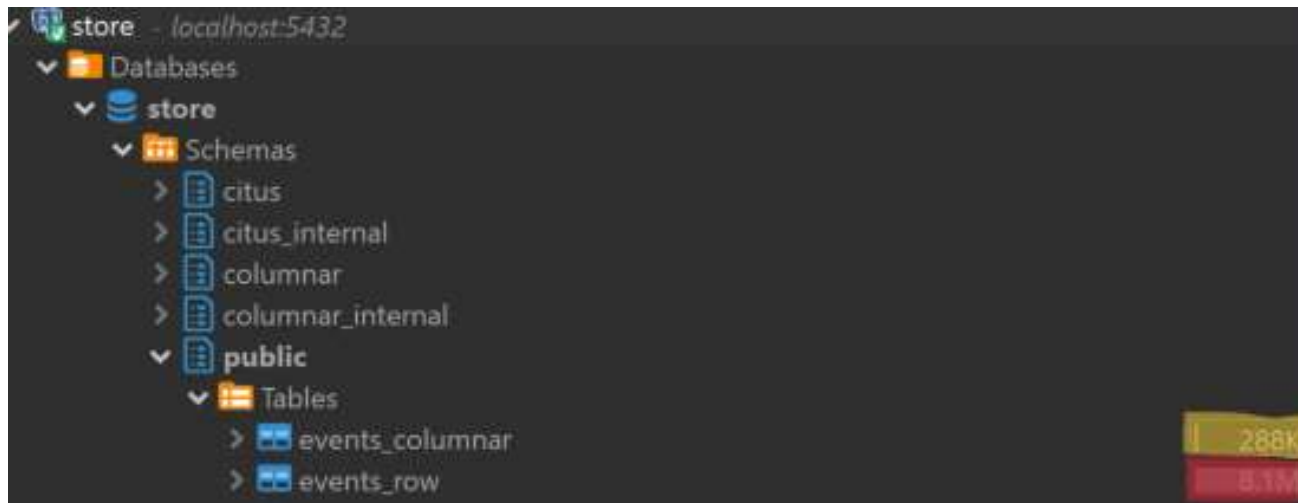
device_id	event_id	event_time	data
1	1	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
2	2	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
3	3	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
4	4	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
5	5	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
6	6	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
7	7	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
8	8	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
9	9	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
10	10	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
11	11	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
12	12	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
13	13	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
14	14	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
15	15	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
16	16	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
17	17	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
18	18	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
19	19	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
20	20	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
21	21	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
22	22	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
23	23	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}
24	24	2024-07-19 16:38:48.863 +0700	{"hello": "columnar"}

The interface also shows a status bar at the bottom indicating that 200 row(s) were fetched, with a fetch rate of 0.017s (0.001s fetch), on 2024-07-19 at 22:44:20.

4. Tampilkan perbedaan ukuran antara tabel biasa dan tabel columnar

Table events_columnar = table columnar

Table_row = table biasa



5. Tuliskan kesimpulannya

Berdasarkan hasil dari perintah yang dijalankan, kita akan melihat perbedaan ukuran antara tabel dengan penyimpanan kolom dan tabel biasa.

Kesimpulan umum yang bisa diambil:

- Tabel dengan penyimpanan kolom (columnar) biasanya lebih efisien dalam penggunaan ruang untuk data yang besar dan terkompresi dengan baik, terutama jika data tersebut sering diakses dalam bentuk kolom (misalnya, analisis data).
- Tabel biasa (row-based) mungkin lebih efisien untuk transaksi yang sering melakukan operasi penulisan dan pembacaan pada satu atau beberapa baris secara sekaligus, karena desainnya yang mengoptimalkan akses baris demi baris.

Dapat dilihat bahwa perbedaan ukuran antara table biasa dan table columnar. Table columnar akan lebih efisien dalam hal penggunaan ruang penyimpanan karena format penyimpanan kolom mengurangi duplikasi data dan meningkatkan kompresi.