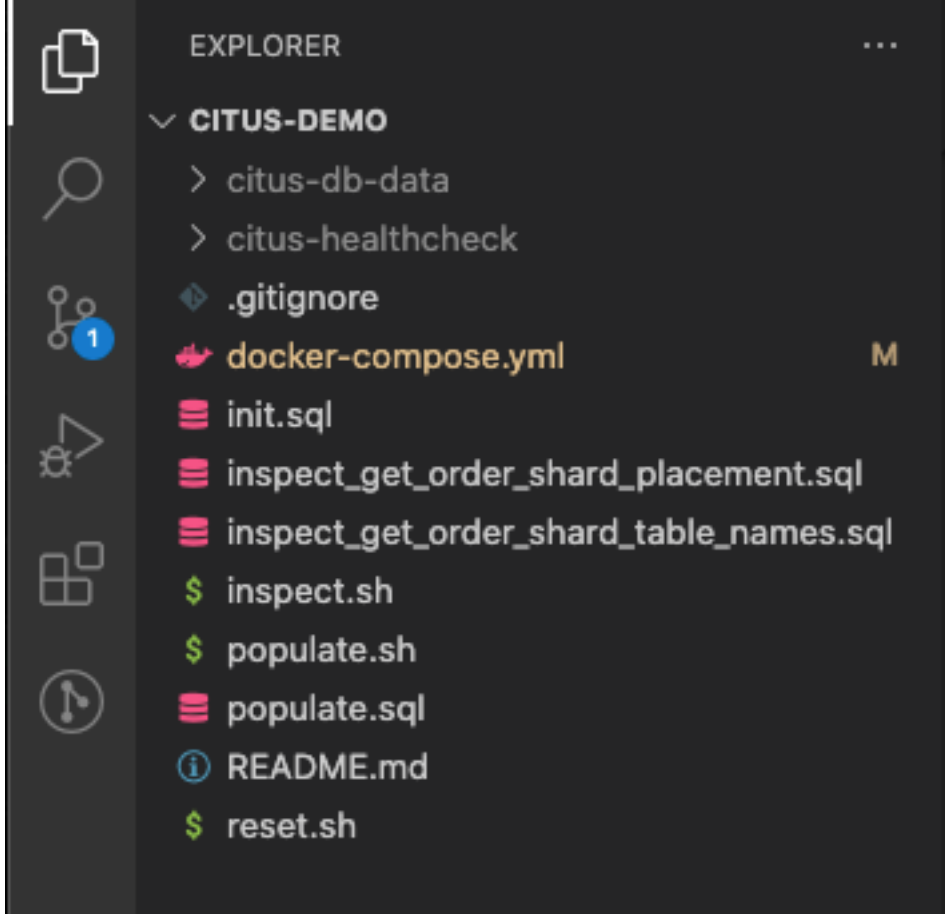




## PART 2 - DATA WAREHOUSE

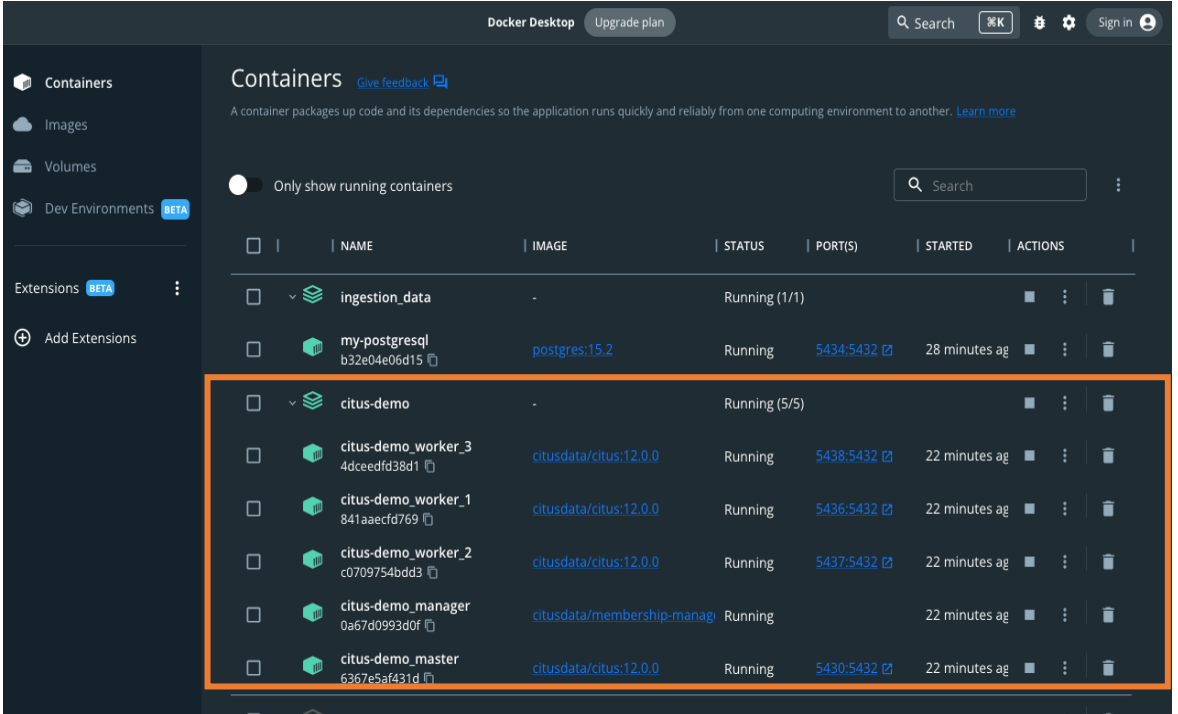
### Columnar Database

# Data Warehouse

No	Deskripsi	Airflow Connection
1	<p>Dalam program ini hal yang pertama dilakukan adalah clone git repository yang disediakan oleh Alterra ke dalam computer local kita</p> <p>Dengan perintah</p> <pre>git clone https://github.com/Immersive-DataEngineer-Resource/citus-demo</pre>	

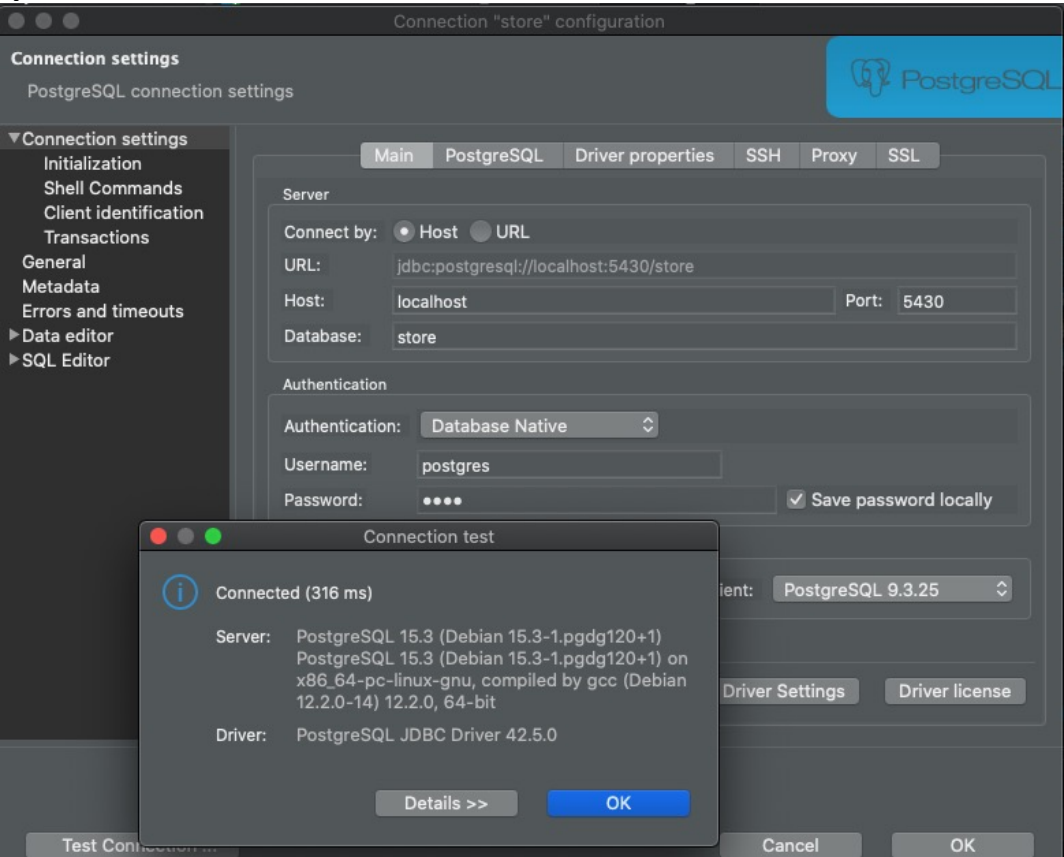
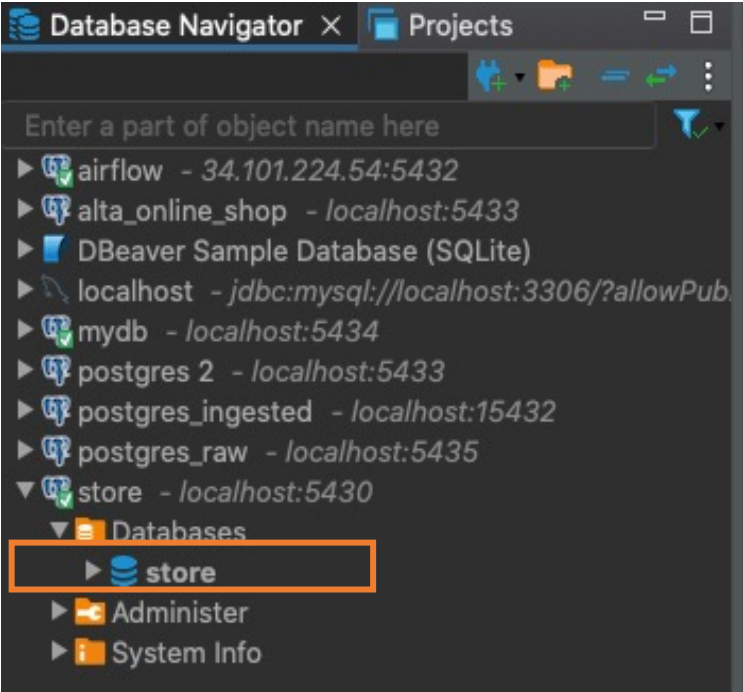
# Data Warehouse

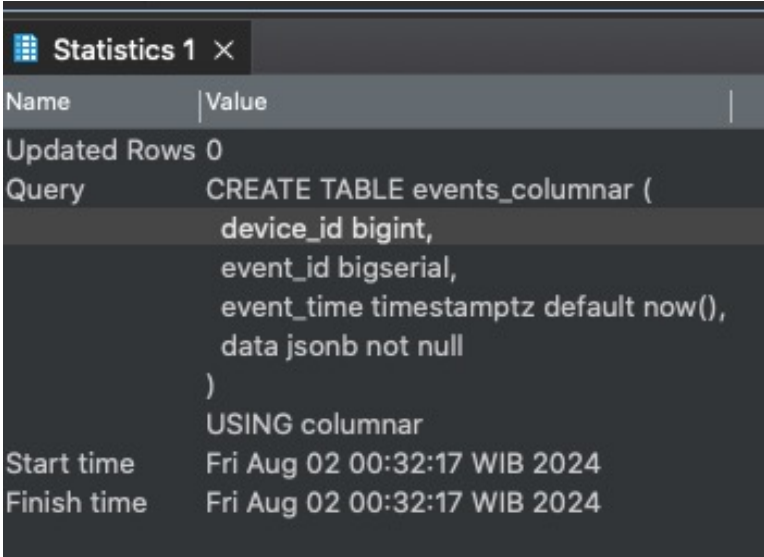


No	Deskripsi	Output
1.1	<pre>wartadi@wartadis-MacBook-Pro citus-demo % docker compose up -d [+] Running 5/5  # Container citus-demo_master   Started    1.0s  # Container citus-demo_manager Started    2.8s  # Container citus-demo_worker_1 Started    6.8s  # Container citus-demo_worker_2 Started    6.5s  # Container citus-demo_worker_3 Started    5.9s</pre> <p>Dalam tugas nomor 1 perintah untuk menjalankan Citus di computer kita dengan docker compse</p> <p>Perintah yang diberikan :</p> <p><b>docker compose up -d</b></p> <p>dalam hal ini port yang digunakan disesuaikan agar citus yang kita running tidak ada error. Port yang digunakan antara lain</p> <p>Master : 5430 Worker 1 : 5436 Worker 2 : 5437 Worker 3 : 5438</p>	 <p>Dalam citus-demo disini ada 5 container yang sedang running antara lain citus-demo-worker1, citus-demo-worker2, citus-demo-worker3, citus-demo-manager dan citus-demo-master</p>

# Data Warehouse

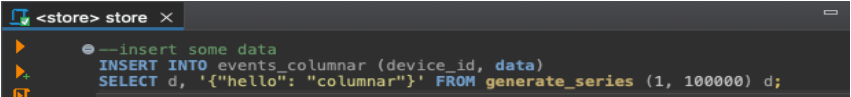
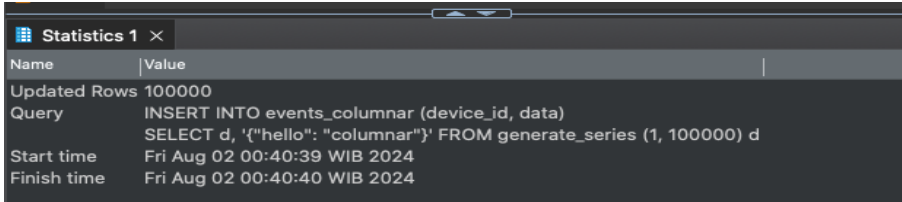
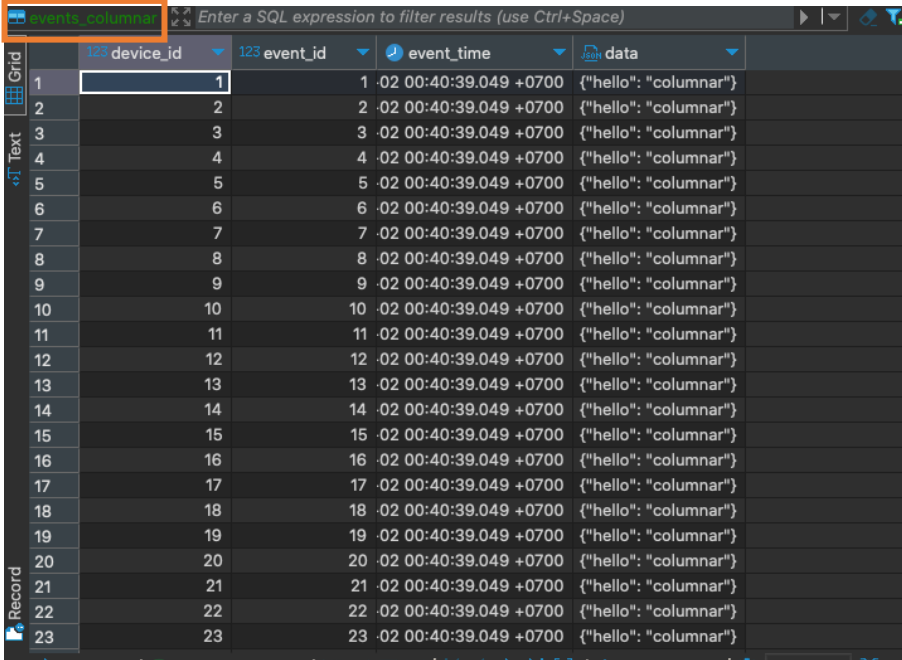


No	Perintah	Output
2	<p>Dalam membuat database perlu membuat connection database postgresQL terlebih dahulu. Input yang digunakan seperti nama <b>Database</b>, <b>Port :5430 (master)</b> , <b>Username</b> dan <b>password</b> disesuaikan dengan yang ada di <b>docker compose</b></p> 	<p>Pada Database Navigator sudah terdapat connection database dengan nama <b>store</b> dengan <b>localhost:5430</b> dan ada database dengan nama <b>store</b></p> 

No	Perintah	Output
2	<p>Tuliskan perintah untuk untuk membuat table biasa dan table columnar. Berikut untuk perintahnya:</p>  <p><b>CREATE TABLE events_columnar:</b> Membuat sebuah tabel baru bernama events_columnar.</p> <p><b>device_id bigint:</b> Kolom device_id bertipe bilangan bulat besar (bigint).</p> <p><b>event_id bigserial:</b> Kolom event_id bertipe bilangan bulat besar yang secara otomatis bertambah (serial) untuk memberikan ID unik pada setiap peristiwa.</p> <p><b>event_time timestamptz default now():</b> Kolom event_time bertipe timestamp dengan zona waktu, secara otomatis diisi dengan waktu saat data dimasukkan.</p> <p><b>data jsonb not null:</b> Kolom data bertipe JSONB untuk menyimpan data peristiwa dalam format JSON, dan data ini wajib diisi.</p> <p><b>USING columnar:</b> Instruksi khusus untuk menyimpan tabel dalam format kolumnar. Format ini mengoptimalkan penyimpanan data untuk analisis, di mana data untuk setiap kolom disimpan secara berurutan.</p>	<p>Berikut merupakan output jika perintah yang sudah dideskripsikan untuk membuat table biasa dan columnar berhasil</p> 

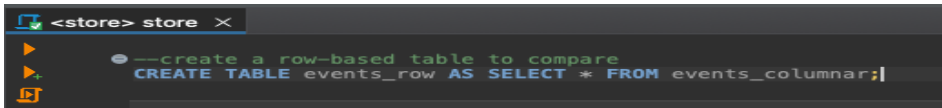
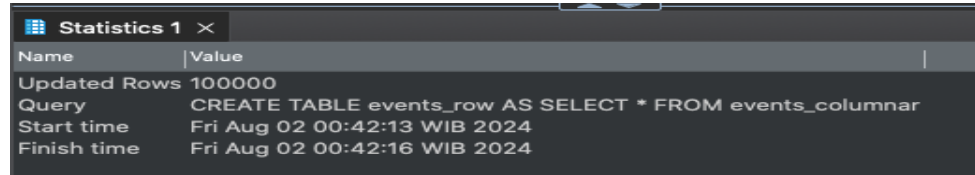
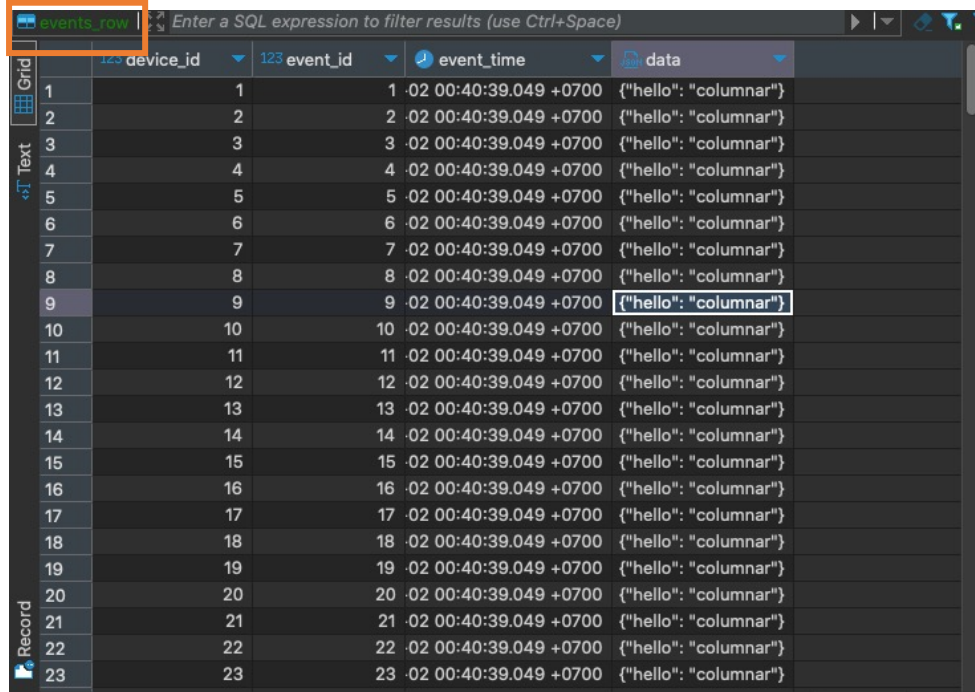
# DATA WAREHOUSE



No	Deskripsi	Output
3	<p>Memasukan 100.000 baris data kedalam table biasa dan table columnar</p> <p>Berikut untuk perintahnya :</p> <div></div> <p><b>INSERT INTO events_columnar:</b> Memasukkan data ke dalam tabel events_columnar. <b>SELECT d, '{"hello": "columnar"}' FROM generate_series (1, 100000) d:</b> generate_series (1, 100000): Menghasilkan deret angka dari 1 hingga 100.000. Setiap angka akan menjadi nilai untuk kolom device_id. <b>'{"hello": "columnar"}':</b> Data JSON statis yang akan dimasukkan ke dalam kolom data untuk setiap baris.</p>	<p>Berikut merupakan hasil dari perintah yang sudah dijalankan. Telah dilakukan proses memasukkan data ke dalam sebuah tabel database bernama events_columnar</p> <div> </div>

# Data Warehouse



No	Input & Deskripsi	Perintah/Output
3	<p>Melanjutkan perintah diatas untuk membuat table event_row dan memproses/insert data</p>  <p><b>CREATE TABLE events_row:</b> Membuat tabel baru bernama events_row. <b>AS SELECT * FROM events_columnar:</b> Mengisi tabel baru dengan semua data dari tabel events_columnar. Namun, tabel baru ini akan disimpan dalam format baris standar, bukan format kolomar.</p> <p><b>Ringkasan</b> Kode di atas melakukan hal berikut: Membuat sebuah tabel events_columnar dengan format penyimpanan kolomar untuk menyimpan data peristiwa. Memasukkan data sampel ke dalam tabel events_columnar. Membuat tabel events_row sebagai salinan dari events_columnar tetapi dengan format penyimpanan baris.</p>	<p>Sebuah tabel baru bernama events_row telah dibuat dengan cara menyalin seluruh data dari tabel events_columnar. Proses ini memakan waktu sekitar 3 detik untuk memproses 100.000 baris data.</p>  

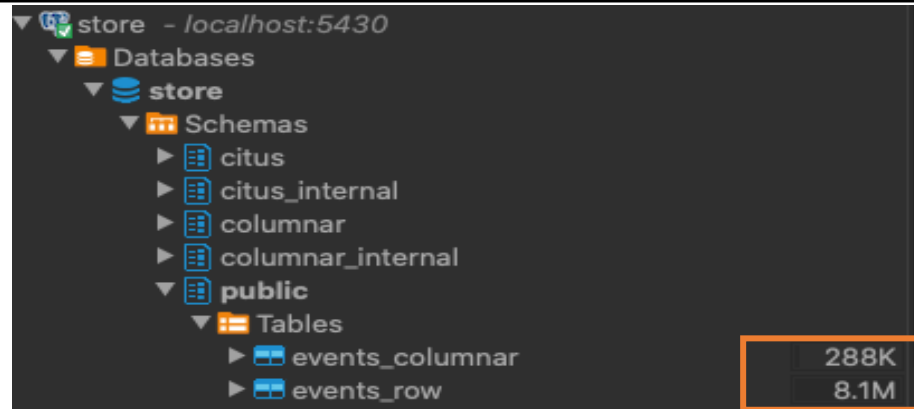


# Data Warehouse

No

Tampilkan Perbedaan Ukuran Table Columnar dan Row(Heap)

4



```
wartadi@wartadis-MacBook-Pro citus-demo % docker exec -it citus-demo_master psql -U postgres -d store
psql (15.3 (Debian 15.3-1.pgdg120+1))
Type "help" for help.
```

```
store=# d\+
invalid command \+
Try \? for help.
store=# \d+
```

List of relations							
Schema	Name	Type	Owner	Persistence	Access method	Size	Description
public	citius_schemas	view	postgres	permanent		0 bytes	
public	citius_tables	view	postgres	permanent		0 bytes	
public	events_columnar	table	postgres	permanent	columnar	288 kB	
public	events_columnar_event_id_seq	sequence	postgres	permanent		8192 bytes	
public	events_row	table	postgres	permanent	heap	8288 kB	
public	order_details	table	postgres	permanent	heap	0 bytes	
public	order_details_order_detail_id_seq	sequence	postgres	permanent		8192 bytes	
public	orders	table	postgres	permanent	heap	0 bytes	
public	orders_order_id_seq	sequence	postgres	permanent		8192 bytes	
public	products	table	postgres	permanent	heap	8192 bytes	
public	products_product_id_seq	sequence	postgres	permanent		8192 bytes	
public	users	table	postgres	permanent	heap	8192 bytes	
public	users_user_id_seq	sequence	postgres	permanent		8192 bytes	

(13 rows)

```
store=#
```



No	Kesimpulan
5	<p><b>events_columnar:</b> Tabel ini memiliki ukuran sekitar <b>288KB</b>. Nama "columnar" mengindikasikan bahwa tabel ini kemungkinan besar menggunakan format penyimpanan kolom, di mana data untuk setiap kolom disimpan secara berurutan. Format ini seringkali lebih efisien untuk query analitik. Tabel columnar terlihat ukurannya dibandingkan table events_row biasa karena kompresi kolom yang lebih efisien.</p> <p><b>events_row:</b> Tabel ini memiliki ukuran sekitar <b>8.1MB</b>. Nama "row" mengindikasikan bahwa tabel ini menggunakan format penyimpanan baris, di mana setiap baris data disimpan secara berurutan. Ini adalah format penyimpanan yang paling umum digunakan.</p> <p><b>Perbedaan Ukuran yang Signifikan:</b> Perbedaan ukuran yang sangat besar antara kedua tabel ini (hampir 30 kali lipat) mengindikasikan adanya perbedaan dalam cara data disimpan. Kemungkinan penyebab perbedaan ukuran ini adalah:</p>

THANK YOU 😊

