

DATA WAREHOUSE

PART 3 – Replication + Sharding

TASK

1. Jelaskan perbedaan antara replication dan sharding!

Jawab:

Replication: Proses menduplikasi data ke beberapa server atau node. Setiap node memiliki salinan lengkap dari data. Tujuannya untuk meningkatkan ketersediaan data (high availability) dan redundansi, serta mempercepat waktu baca karena data bisa dibaca dari beberapa lokasi.

Sharding: Proses membagi data ke dalam beberapa bagian yang lebih kecil (shard) dan menyimpannya di node yang berbeda. Setiap node hanya menyimpan bagian tertentu dari keseluruhan data. Tujuannya untuk meningkatkan kinerja dengan membagi beban kerja ke beberapa server dan memungkinkan skalabilitas horizontal.

2. Lakukan percobaan untuk membuat reference table + distributed table seperti pada repo <https://github.com/Immersive-DataEngineer-Resource/citus-demo>!

Jawab:

- a. Menjalankan container

```
irul_jj@Pringo:~/IrulJJ2/Altera2/citus-demo$ docker compose up -d
WARN[0001] /home/irul_jj/IrulJJ2/Altera2/citus-demo/docker-compose.yml: `version` is obsolete
[+] Running 5/5
 ✓ Container citus-demo_master      Running      0.0s
 ✓ Container citus-demo_manager     Running      0.0s
 ✓ Container citus-demo_worker_3    Running      0.0s
 ✓ Container citus-demo_worker_1    Running      0.0s
 ✓ Container citus-demo_worker_2    Running      0.0s
```

- b. Membuat reference table dan distributed table (part1)

```
irul_jj@Pringo:~/IrulJJ2/Altera2/citus-demo$ ./populate.sh
CREATE TABLE
  create_reference_table
-----
(1 row)

CREATE TABLE
  create_reference_table
-----
(1 row)

CREATE SEQUENCE
CREATE TABLE
  create_distributed_table
-----
(1 row)

CREATE SEQUENCE
CREATE TABLE
  create_distributed_table
-----
(1 row)

INSERT 0 2
INSERT 0 4
DO
```

c. Membuat reference table dan distributed table (part2)

```

irul_jj@Pringgo:~/Iru1J32/Altera2/citus-demo$ ./inspect.sh
=====
MASTER
=====
Active worker count:
3

Get nodes
nodeid | groupid | node name | nodeport | noderack | hasmetadata | isactive | noderole | nodecluster | metadatasynced | shouldhaveshards
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----
1 | 1 | citus-demo_worker_1 | 5432 | default | t | t | primary | default | t | t
2 | 2 | citus-demo_worker_3 | 5432 | default | t | t | primary | default | t | t
3 | 3 | citus-demo_worker_2 | 5432 | default | t | t | primary | default | t | t
(3 rows)

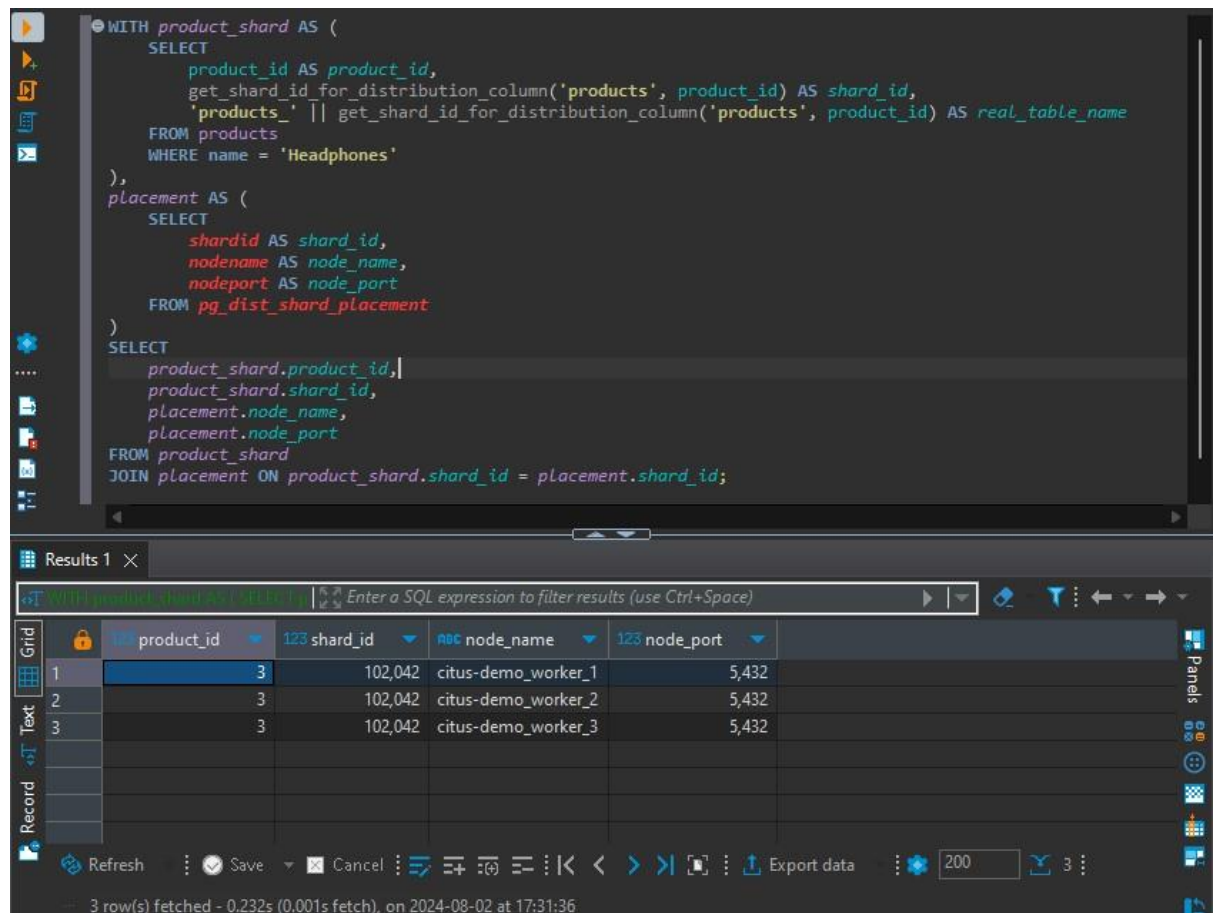
Order shard placement:
order_id | shard_id | real_table_name | node_name
-----|-----|-----|-----
1 | 102044 | orders_102044 | citus-demo_worker_2
5 | 102049 | orders_102049 | citus-demo_worker_1
4 | 102051 | orders_102051 | citus-demo_worker_3
3 | 102058 | orders_102058 | citus-demo_worker_1
2 | 102067 | orders_102067 | citus-demo_worker_1
(5 rows)

Order shard table names:
orders_102044
orders_102067
orders_102058
orders_102051
orders_102049

```

3. Di node/worker mana saja product "Headphone" tersimpan? Tunjukkan shard id nya!

Jawab:



```

WITH product_shard AS (
    SELECT
        product_id AS product_id,
        get_shard_id_for_distribution_column('products', product_id) AS shard_id,
        'products_' || get_shard_id_for_distribution_column('products', product_id) AS real_table_name
    FROM products
    WHERE name = 'Headphones'
),
placement AS (
    SELECT
        shardid AS shard_id,
        node name AS node_name,
        nodeport AS node_port
    FROM pg_dist_shard_placement
)
SELECT
    product_shard.product_id,
    product_shard.shard_id,
    placement.node_name,
    placement.node_port
FROM product_shard
JOIN placement ON product_shard.shard_id = placement.shard_id;

```

Results 1

	product_id	shard_id	node_name	node_port
1	3	102,042	citus-demo_worker_1	5,432
2	3	102,042	citus-demo_worker_2	5,432
3	3	102,042	citus-demo_worker_3	5,432

3 row(s) fetched - 0.232s (0.001s fetch), on 2024-08-02 at 17:31:36

4. Di node/worker mana saja order dengan id 13 tersimpan? Tunjukkan shard id nya!

Jawab:

```
WITH order_shard AS (  
  SELECT  
    order_id AS order_id,  
    get_shard_id_for_distribution_column('orders',order_id) AS shard_id,  
    'orders_' || get_shard_id_for_distribution_column('orders', order_id) AS real_table_name  
  FROM orders  
  WHERE order_id = 13  
,  
  placement AS (  
    SELECT  
      shardid AS shard_id,  
      node_name AS node_name,  
      nodeport AS node_port  
    FROM pg_dist_shard_placement  
  )  
SELECT  
  order_shard.order_id,  
  order_shard.shard_id,  
  placement.node_name,  
  placement.node_port  
FROM order_shard  
JOIN placement ON order_shard.shard_id = placement.shard_id;
```

Results 1

	order_id	shard_id	node_name	node_port
1	13	102,066	citus-demo_worker_3	5,432

1 row(s) fetched - 2s, on 2024-08-02 at 17:38:31

5. Kapan sebaiknya kita menggunakan replication?

Jawab:

- Ketika data harus selalu tersedia meskipun terjadi kegagalan server.
- Ketika prioritas utama adalah kecepatan membaca data.
- Untuk redundansi data guna mencegah kehilangan data.

6. Kapan sebaiknya kita menggunakan sharding?

Jawab:

- Ketika data yang dikelola sangat besar sehingga tidak dapat disimpan di satu server saja.
- Ketika perlu meningkatkan kinerja dengan mendistribusikan beban kerja ke beberapa server.
- Ketika aplikasi memerlukan skalabilitas horizontal untuk menangani pertumbuhan data dan beban pengguna.