Part 3- Replication & Sharding

1. Jelaskan perbedaan antara replication dan sharding !
   Jawab :
   - Replication adalah proses membuat dan menjaga salinan data ke beberapa node. Data disalin ke beberapa node yang terpisah sehingga jika satu node gagal, data masih tersedia di node lainnya. Replication digunakan untuk data table master
   - Sharding adalah teknik untuk membagi dataset menjadi bagian-bagian yang disebut shard dan mendistribusikannya di antara beberapa server atau node. Sharding untuk data-data cardinalitas tinggi atau yang punya keunikan atau untuk data transaction.
2. Lakukan percobaan untuk membuat reference table dan distributed table :
   Jawab :
   Tabel reference

```
-- Create users table (Reference Table)
CREATE TABLE users (
  user_id SERIAL PRIMARY KEY,
  username TEXT NOT NULL,
  email TEXT NOT NULL UNIQUE
);
SELECT create_reference_table('users');
```

Name	Value
Updated Rows	0
Query	CREATE TABLE users (
	user_id SERIAL PRIMARY KEY,
	username TEXT NOT NULL,
	email TEXT NOT NULL UNIQUE
	);
	SELECT create_reference_table('users')
Start time	Mon Apr 08 13:29:17 WIB 2024
Finish time	Mon Apr 08 13:29:18 WIB 2024

```
-- Create products table (Reference Table)
CREATE TABLE products (
  product_id SERIAL PRIMARY KEY,
  name TEXT NOT NULL,
  price NUMERIC(10, 2) NOT NULL
);

SELECT create_reference_table('products');
```

Name	Value
Updated Rows	0

Query	CREATE TABLE products (
	  product_id SERIAL PRIMARY KEY,
	  name TEXT NOT NULL,
	  price NUMERIC(10, 2) NOT NULL
	);
	SELECT create_reference_table('products')

Start time	Mon Apr 08 13:30:54 WIB 2024
Finish time	Mon Apr 08 13:30:55 WIB 2024

Table distribute

```
-- Create sequence for orders (Distributed Table)
CREATE SEQUENCE orders_order_id_seq;

-- Create orders table
CREATE TABLE orders (
  order_id INT DEFAULT nextval('orders_order_id_seq'),
  user_id INT REFERENCES users(user_id),
  total_price NUMERIC(10, 2) NOT NULL,
  created_at TIMESTAMPTZ DEFAULT NOW()
);
SELECT create_distributed_table('orders', 'order_id');
```

Name	Value
Updated Rows	0

Query	CREATE SEQUENCE orders_order_id_seq;
	-- Create orders table
	CREATE TABLE orders (
	  order_id INT DEFAULT nextval('orders_order_id_seq'),
	  user_id INT REFERENCES users(user_id),
	  total_price NUMERIC(10, 2) NOT NULL,
	  created_at TIMESTAMPTZ DEFAULT NOW()
	);
	SELECT create_distributed_table('orders', 'order_id')

Start time	Mon Apr 08 13:33:37 WIB 2024
Finish time	Mon Apr 08 13:33:37 WIB 2024

```
-- Create sequence for order_details (Distributed Table)
CREATE SEQUENCE order_details_order_detail_id_seq;

-- Create order_details table
CREATE TABLE order_details (
  order_detail_id INT DEFAULT nextval('order_details_order_detail_id_seq'),
  order_id INT,
  product_id INT,
  quantity INT NOT NULL
);
SELECT create_distributed_table('order_details', 'order_id');
```

Name	Value
Updated Rows	0
Query	CREATE SEQUENCE order_details_order_detail_id_seq; -- Create order_details table CREATE TABLE order_details (order_detail_id INT DEFAULT nextval('order_details_order_detail_id_seq'), order_id INT, product_id INT, quantity INT NOT NULL); SELECT create_distributed_table('order_details', 'order_id')
Start time	Mon Apr 08 13:34:40 WIB 2024
Finish time	Mon Apr 08 13:34:41 WIB 2024

3. Di node/worker mana saja prodict ’Handphone’ tersimpan tunjukan shard-idnya !
   Jawab :

```
WITH placement AS (
    SELECT
        shardid as shard_id
        , nodename as node_name
    FROM pg_dist_shard_placement
)

, order_ids AS (
    SELECT product_id
    FROM products
    where name = 'Headphones'
    ORDER BY product_id

)

, order_shards AS (
    SELECT
        product_id
        , get_shard_id_for_distribution_column('orders', product_id) as shard_id
        , 'orders_' || get_shard_id_for_distribution_column('orders', product_id) as real_table_name
    FROM order_ids
)

SELECT
    order_shards.*
    , placement.node_name
FROM order_shards
INNER JOIN placement
    ON placement.shard_id = order_shards.shard_id
;
```

123 product_id	123 shard_id	ABC real_table_name	ABC node_name
3	102,193	orders_102193	citus-demo_worker_1

Keterangan : product dengan nama handphones dengan product_id 3 berada pada node worker_1 dengan shard_id 102,193. Hasil yang didapat tidak sesuai ekpetasi, seharusnya data tersimpan di tiga node atau worker 1 s/d 3 mengingat table produk merupakan table duplicate, tapi yang terjadi dilaptop saya malah seperti tersimpan secara sharding.

4. Di node/worker mana saja order dengan id 13 tersimpan ? tunjukan shard-idnya!
   Jawaban :

```
WITH placement AS (
    SELECT
        shardid as shard_id
        , nodename as node_name
    FROM pg_dist_shard_placement
)
, order_ids AS (
    SELECT order_id
    FROM orders
    where order_id = 13
    ORDER BY order_id

)
, order_shards AS (
    SELECT
        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 order_ids
)
SELECT
    order_shards.*
    , placement.node_name
FROM order_shards
INNER JOIN placement
    ON placement.shard_id = order_shards.shard_id
;
```

	123 order_id	123 shard_id	ABC real_table_name	ABC node_name
1	13	102,201	orders_102201	citus-demo_worker_3

   Keterangan : order dengan id 13 tersimpan pada node worker_3 dengan shard_id 102.201.

5. Kapan sebaiknya kita menggunakan replication ?
   Tabel replication sebaiknya digunakan untuk data master dengan volume data yang realtif kecil dan pertumbuhan data tidak signifikan.
6. Kanan sebainya kita menggunakan sharding ?
   Tabel sharding sebaiknya digunakan untuk data transaction dengan pertumbuhan data yang signifikan dan volume yang besar.