Nurul Taufik Kurrahman

Task Part 3 Data Warehouse
Replication & Sharding

1. Jelaskan perbedaan antara replication dan sharding!

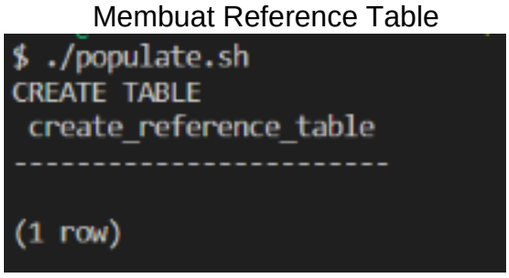
   Replication:
   - Redundansi Data: Replication melibatkan menyalin data dari satu server database ke server lain, menciptakan salinan yang identik dari data.
   - Failover dan Ketersediaan: Replication membantu meningkatkan ketersediaan dengan menyediakan salinan cadangan data jika salah satu server mengalami kegagalan. Ini mengurangi downtime.
   - Latensi Rendah: Kueri dapat diarahkan ke server lokal yang berisi salinan data, menghasilkan latensi yang rendah untuk akses baca.
   - Skalabilitas Terbatas: Replication biasanya tidak menyebabkan skalabilitas horizontal (menambahkan server secara linier) karena data ada di semua server.

   Sharding:
   - Pemecahan Data: Sharding melibatkan membagi data menjadi potongan-potongan kecil (shard) dan mendistribusikannya ke berbagai server. Setiap server hanya berisi sebagian data.
   - Skalabilitas Horizontal: Sharding meningkatkan skalabilitas horizontal dengan menambahkan server saat data tumbuh, memungkinkan peningkatan kapasitas dan kinerja.
   - Ketersediaan Tergantung Desain: Ketersediaan dalam sharding tergantung pada bagaimana sistem dirancang, dan beberapa solusi mungkin memerlukan upaya lebih lanjut untuk mencapai failover yang efektif.
   - Latensi Variabel: Kueri mungkin mengalami latensi yang bervariasi tergantung pada lokasi shard yang dipanggil.

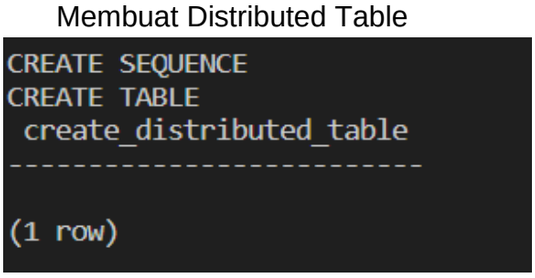
   Pilihan antara replication dan sharding tergantung pada tujuan dan kebutuhan sistem distribusi. Replication lebih cocok untuk meningkatkan ketersediaan dan memastikan kehandalan dalam kasus kegagalan server tunggal. Sharding lebih cocok untuk peningkatan kapasitas dan skalabilitas ketika data berkembang dalam skala yang sangat besar.

2. Lakukan percobaan untuk membuat reference table + distributed table seperti pada repo https://github.com/Immersive-Data Engineer-Resource/citus-demo

Membuat Reference Table



Membuat Distributed Table



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

```
WITH product_shard AS (
    SELECT
        'products_' || get_shard_id_for_distribution_column('products', product_id) AS real_table_name,
        get_shard_id_for_distribution_column('products', product_id) AS shard_id
    FROM products
    WHERE name = 'Headphones'
)

SELECT
    product_shard.real_table_name,
    product_shard.shard_id,
    placement.nodename
FROM product_shard
JOIN pg_dist_shard_placement AS placement
    ON product_shard.shard_id = placement.shardid;
```

Results 1

WITH product_shard AS (SELECT 'products_' || — Enter a SQL expression to filter results (use Ctrl+Space)

	real_table_name	shard_id	nodename
1	products_102009	102,009	citus-demo_worker_1
2	products_102009	102,009	citus-demo_worker_1
3	products_102009	102,009	citus-demo_worker_1
4	products_102009	102,009	citus-demo_worker_2
5	products_102009	102,009	citus-demo_worker_2
6	products_102009	102,009	citus-demo_worker_2
7	products_102009	102,009	citus-demo_worker_3
8	products_102009	102,009	citus-demo_worker_3
9	products_102009	102,009	citus-demo_worker_3

Value
products_102009

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

```
SELECT * FROM pg_dist_shard_placement;

SELECT * FROM pg_dist_shard;


SELECT get_shard_id_for_distribution_column('orders', 13);
```

ilts 1

CT get_shard_id_for_distribution_column(' — Enter a SQL expression to filter results (use Ctrl+Space)

get_shard_id_for_distribution_column
102,033

Value
102033

5. Kapan sebaiknya kita menggunakan replication?

Replication sebaiknya digunakan ketika memerlukan:

- Ketersediaan tinggi dan ketahanan terhadap kegagalan server.
- Salinan data yang terdistribusi secara geografis.
- Pengurangan latensi baca dan peningkatan kinerja.
- Distribusi lalu lintas baca untuk penyeimbangan beban.
- Lingkungan pengujian dan pengembangan yang serupa dengan produksi.

Ini membantu dalam situasi di mana data tidak boleh hilang dan sistem perlu beroperasi dengan minimnya gangguan.

6. Kapan sebaiknya kita menggunakan sharding?

.

Sharding sebaiknya digunakan ketika Anda menghadapi:

- Pertumbuhan data yang cepat.
- Kinerja tinggi yang diperlukan.
- Kebutuhan pengelolaan data yang terdistribusi.
- Kehadiran global yang memerlukan akses data dekat pengguna.
- Batasan kapasitas server tunggal.
- Keinginan untuk memisahkan data pelanggan atau unit bisnis.

Sharding membantu dalam meningkatkan kapasitas, kinerja, dan pengelolaan data dalam skenario seperti ini.