Nama: Qorina M. H. Mumtaza

Kelas: DE-1

1. Sebutkan perbedaan antara data warehouse dan data lake!
   Jawab:
   Data warehouse dan data lake adalah dua pendekatan yang berbeda dalam hal penyimpanan dan pengelolaan data. Penggunaannya masing-masing disesuaikan dengan jenis data dan kebutuhan analitis yang berbeda. Beberapa perbedaannya antara lain;
   - Struktur dan skema data:
     Data warehouse menyimpan data terstruktur yang sudah disusun ke dalam tabel dengan skema yang telah ditentukan sebelumnya. Data warehouse biasanya menggunakan sistem RDBMS untuk menegakkan skema dan menjaga integritas data. Sedangkan data lake menyimpan data baik yang terstruktur maupun tidak terstruktur dalam format mentah dan aslinya.
   - Penyerapan data:
     Data warehouse dirancang untuk pemrosesan batch dan data terstruktur yang di dalamnya terdapat proses di mana data dibersihkan, diubah, dan dimuat (ETL) ke dalam warehouse sebelum dapat digunakan untuk analisis. Sedangkan data lake memiliki karakteristik yang lebih fleksibel dalam hal penyerapan data. Data lake dapat menyerap data dalam bentuk mentah dan tidak memerlukan definisi skema di awal atau proses ETL. Data lake dapat menangani data streaming dan data batch secara real time sehingga cocok untuk kasus penggunaan data skala besar dan Internet of Things (IoT).
   - Skalabilitas:
     Data warehouse lebih sulit untuk diskalakan karena lebih besar kemungkinan digunakan untuk merancang data terstruktur, sehingga proses penskalaannya bisa memakan cost yang tinggi. Sedangkan data lake sangat mudah diskalakan, baik s ecara vertikal maupun horizontal. Data lake dapat berkembang untuk menampung data dalam jumlah besar tanpa membutuhkan peningkatan biaya yang signifikan.
   - Biaya/cost:
     Data warehouse cenderung memakan biaya yang lebih besar dari segi manapun. Hal ini dikarenakan data warehouse cenderung dirancang untuk pemrosesan data yang terstruktur dan terdefinisi dengan baik. Berbeda halnya dengan data lake yang lebih hemat biaya karena memanfaatkan solusi penyimpanan cloud dan teknologi open source.
   - Query dan analisis:
     Data warehouse dioptimalkan untuk query dan pelaporan berbasis SQL, menjadikannya cocok untuk intelijen bisnis tradisional dan analisis data. Data warehouse memberikan konsistensi data yang kuat dan sangat cocok untuk query ad-hoc. Sedangkan data lake lebih cocok digunakan untuk eksplorasi data dan analisis tingkat lanjut, termasuk machine learning dan deep learning.

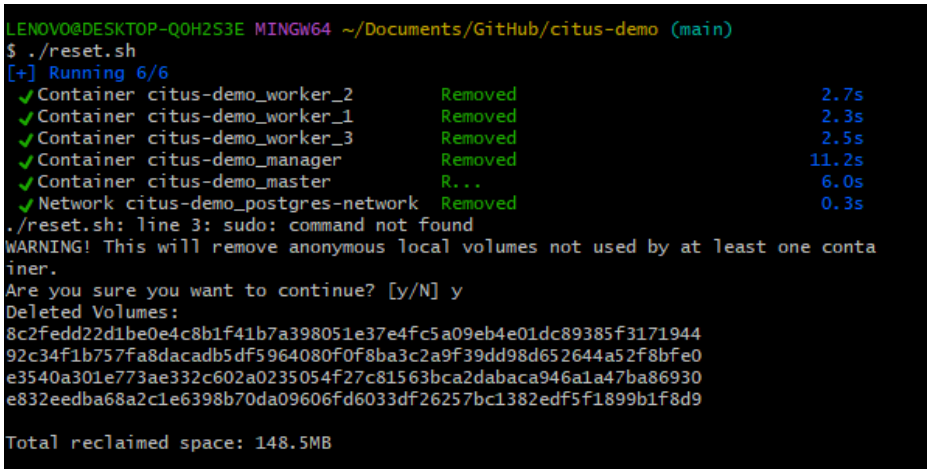
2. Apa yang membedakan teknologi database untuk data warehouse (OLAP) dari teknologi database konvensional (OLTP)?

Jawab:
Teknologi basis data untuk data warehouse (OLAP) dan teknologi basis data konvensional (OLTP) dirancang untuk tujuan berbeda dan memiliki beberapa pembeda utama berdasarkan kasus penggunaan dan persyaratannya:

- Model data: OLTP dirancang untuk menangani operasi transaksi sehari-hari, dengan data normal dan terstruktur sedangkan basis data OLAP dirancang untuk query dan pelaporan analitis yang kompleks.
- Pola baca-tulis: OLTP mendukung operasi baca dan tulis frekuensi tinggi, sedangkan OLAP fokus utamanya adalah menyediakan kinerja query yang cepat untuk query analitis yang kompleks.
- Kompleksitas query: OLTP biasanya memiliki query yang sederhana dan dirancang untuk operasi transaksional dan pengambilan data, sedangkan OLAP seringkali memiliki query yang rumit dan melibatkan agregasi, pemfilteran, dan pengelompokan kumpulan data besar.
- Volume data: OLTP mengelola volume data yang relatif kecil hingga sedang, sedangkan OLAP dioptimalkan untuk menangani data historis dalam jumlah besar.
  Singkatnya OLTP dirancang untuk pemrosesan transaksional dan pemeliharaan data operasional, sementara OLAP dirancang untuk analisis kompleks, analisis data historis, dan dukungan keputusan. Perbedaan antara keduanya disebabkan oleh kasus penggunaannya yang berbeda.

3. Teknologi apa saja yang biasanya dipakai untuk data warehouse?
   Data warehouse biasanya menggunakan kombinasi dari beberapa macam teknologi dan komponen untuk mengelola dan memproses data volume besar secara efisien untuk analisis dan keperluan pelaporan. Beberapa yang paling umum digunakan adalah RDBMS, columnar database, data integration dan ETL tools, dan masih banyak lagi tergantung dari requirements, budget, dan teknologi yang sudah ada dari suatu organisasi.

4. Tuliskan setiap perintah dari proses instalasi citus menggunakan docker compose sampai tabel terbentuk, berikan juga tangkapan layar untuk setiap langkah dan hasilnya!
   - ./reset.sh



- docker compose up -d

```
LENOVO@DESKTOP-QOH2S3E MINGW64 ~/Documents/GitHub/citus-demo (main)
$ docker compose up -d
[+] Building 0.0s (0/0)                                          docker:default
[+] Running 6/6
 ✓ Network citus-demo_postgres-network  Created                          0.2s
 ✓ Container citus-demo_master          S...                             0.5s
 ✓ Container citus-demo_manager         Started                          0.1s
 ✓ Container citus-demo_worker_2        Started                          0.2s
 ✓ Container citus-demo_worker_3        Started                          0.2s
 ✓ Container citus-demo_worker_1        Started                          0.2s
```

- ./populate.sh

```
LENOVO@DESKTOP-QOH2S3E MINGW64 ~/Documents/GitHub/citus-demo (main)
$ ./populate.sh
ERROR:  relation "users" already exists
ERROR:  This is an internal Citus function can only be used in a distributed tra
nsaction
CONTEXT:  while executing command on citus-demo_worker_3:5432
ERROR:  relation "products" already exists
ERROR:  This is an internal Citus function can only be used in a distributed tra
nsaction
 nodeid | groupid |      nodename       | nodeport | noderack | hasmetadata | isactive | noderole | nodecluster | metadatasynced | shouldhaveshards
--------+---------+---------------------+----------+----------+-------------+----------+----------+-------------+----------------+------------------
      1 |       1 | citus-demo_worker_3 |     5432 | default  | t           | t
       | primary  | default     | t              | t
      2 |       2 | citus-demo_worker_2 |     5432 | default  | t           | t
       | primary  | default     | t              | t
      3 |       3 | citus-demo_worker_1 |     5432 | default  | t           | t
       | primary  | default     | t              | t
(3 rows)
```

- ./inspect.sh

```
LENOVO@DESKTOP-QOH2S3E MINGW64 ~/Documents/GitHub/citus-demo (main)
$ ./inspect.sh
================================================================
MASTER
================================================================
Active worker count:
3

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


Order shard placement:
ERROR:  relation is not distributed

ERROR:  relation is not distributed
Order shard table names:


SELECT count(1) FROM orders
 count
-------
  2000
(1 row)
```

5. Jelaskan perbedaan antara access method heap dan columnar pada citus!
   Pemilihan metode akses ‘heap’ dan ‘columnar’ di Citus bergantung pada kasus penggunaan spesifik dan pola query yang dibutuhkan. Jika memerlukan pendukung beban kerja transaksional dan analitis, maka dapat digunakan kombinasi metode akses ini dalam lingkungan database terdistribusi. Akses heap lebih cocok untuk scenario OLTP, sedangkan columnar dioptimalkan untuk kasus penggunaan OLAP.