L1 E3 - Columnar Vs Row Storage - Solution

March 10, 2022

1 Exercise 03 - Columnar Vs Row Storage - Solution

- The columnar storage extension used here:
    - cstore_fdw by citus_data https://github.com/citusdata/cstore_fdw
- The data tables are the ones used by citus_data to show the storage extension

```
In [1]: %load_ext sql
```

1.1 STEP 0 : Connect to the local database where Pagila is loaded

1.1.1 Create the database

```
In [2]: !sudo -u postgres psql -c 'CREATE DATABASE reviews;'

        !wget http://examples.citusdata.com/customer_reviews_1998.csv.gz
        !wget http://examples.citusdata.com/customer_reviews_1999.csv.gz

        !gzip -d customer_reviews_1998.csv.gz
        !gzip -d customer_reviews_1999.csv.gz

        !mv customer_reviews_1998.csv /tmp/customer_reviews_1998.csv
        !mv customer_reviews_1999.csv /tmp/customer_reviews_1999.csv
```

```
CREATE DATABASE
--2022-03-10 06:24:12--  http://examples.citusdata.com/customer_reviews_1998.csv.gz
Resolving examples.citusdata.com (examples.citusdata.com)... 104.26.15.56, 104.26.14.56, 172.67.
Connecting to examples.citusdata.com (examples.citusdata.com)|104.26.15.56|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://examples.citusdata.com/customer_reviews_1998.csv.gz [following]
--2022-03-10 06:24:12--  https://examples.citusdata.com/customer_reviews_1998.csv.gz
Connecting to examples.citusdata.com (examples.citusdata.com)|104.26.15.56|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24774482 (24M) [application/x-gzip]
Saving to: customer_reviews_1998.csv.gz

customer_reviews_19 100%[===================>]  23.63M  22.1MB/s    in 1.1s
```

```
2022-03-10 06:24:14 (22.1 MB/s) - customer_reviews_1998.csv.gz saved [24774482/24774482]

URL transformed to HTTPS due to an HSTS policy
--2022-03-10 06:24:14--  https://examples.citusdata.com/customer_reviews_1999.csv.gz
Resolving examples.citusdata.com (examples.citusdata.com)... 104.26.15.56, 104.26.14.56, 172.67.
Connecting to examples.citusdata.com (examples.citusdata.com)|104.26.15.56|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48996256 (47M) [application/x-gzip]
Saving to: customer_reviews_1999.csv.gz

customer_reviews_19 100%[===================>]  46.73M  95.2MB/s    in 0.5s

2022-03-10 06:24:15 (95.2 MB/s) - customer_reviews_1999.csv.gz saved [48996256/48996256]
```

1.1.2 Connect to the database

```
In [3]: DB_ENDPOINT = "127.0.0.1"
        DB = 'reviews'
        DB_USER = 'student'
        DB_PASSWORD = 'student'
        DB_PORT = '5432'

        # postgresql://username:password@host:port/database
        conn_string = "postgresql://{}:{}@{}:{}/{}" \
                                .format(DB_USER, DB_PASSWORD, DB_ENDPOINT, DB_PORT, DB)

        print(conn_string)
```

```
postgresql://student:student@127.0.0.1:5432/reviews
```

```
In [4]: %sql $conn_string
```

```
Out[4]: 'Connected: student@reviews'
```

1.2 STEP 1: Create a table with a normal (Row) storage & load data

```
In [5]: %%sql
        DROP TABLE IF EXISTS customer_reviews_row;
        CREATE TABLE customer_reviews_row
        (
            customer_id TEXT,
            review_date DATE,
            review_rating INTEGER,
            review_votes INTEGER,
            review_helpful_votes INTEGER,
            product_id CHAR(10),
            product_title TEXT,
            product_sales_rank BIGINT,
            product_group TEXT,
            product_category TEXT,
            product_subcategory TEXT,
            similar_product_ids CHAR(10)[]
        )

 * postgresql://student:***@127.0.0.1:5432/reviews
Done.
Done.


Out[5]: []
```

```
In [6]: %%sql
        COPY customer_reviews_row FROM '/tmp/customer_reviews_1998.csv' WITH CSV;
        COPY customer_reviews_row FROM '/tmp/customer_reviews_1999.csv' WITH CSV;

 * postgresql://student:***@127.0.0.1:5432/reviews
589859 rows affected.
1172645 rows affected.


Out[6]: []
```

1.3 STEP 2: Create a table with columnar storage & load data

```
In [7]: %%sql

        -- load extension first time after install
        CREATE EXTENSION cstore_fdw;

        -- create server object
        CREATE SERVER cstore_server FOREIGN DATA WRAPPER cstore_fdw;

 * postgresql://student:***@127.0.0.1:5432/reviews
Done.
Done.


Out[7]: []
```

```
In [8]: %%sql
        -- create foreign table
        DROP FOREIGN TABLE IF EXISTS customer_reviews_col;

        CREATE FOREIGN TABLE customer_reviews_col
        (
            customer_id TEXT,
            review_date DATE,
            review_rating INTEGER,
            review_votes INTEGER,
            review_helpful_votes INTEGER,
            product_id CHAR(10),
            product_title TEXT,
            product_sales_rank BIGINT,
            product_group TEXT,
            product_category TEXT,
            product_subcategory TEXT,
            similar_product_ids CHAR(10)[]
        )
        SERVER cstore_server
        OPTIONS(compression 'pglz');
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
Done.
Done.
```

```
Out[8]: []
```

```
In [9]: %%sql
        COPY customer_reviews_col FROM '/tmp/customer_reviews_1998.csv' WITH CSV;
        COPY customer_reviews_col FROM '/tmp/customer_reviews_1999.csv' WITH CSV;
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
589859 rows affected.
1172645 rows affected.
```

```
Out[9]: []
```

1.4 Step 3: Compare performance

```
In [10]: %%time
         %%sql
         SELECT
             customer_id, review_date, review_rating, product_id, product_title
         FROM
             customer_reviews_row
         WHERE
             customer_id ='A27T7HVDXA3K2A' AND
             product_title LIKE '%Dune%' AND
             review_date >= '1998-01-01' AND
             review_date <= '1998-12-31';
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
5 rows affected.
```

```
CPU times: user 2.43 ms, sys: 2.77 ms, total: 5.2 ms
Wall time: 4.86 s
```

```
Out[10]: [('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0399128964', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '044100590X', 'Dune'),
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0441172717', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0881036366', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '1559949570', 'Dune Audio Collection
```

```
In [11]: %sql select * from customer_reviews_row limit 10
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
10 rows affected.
```

```
Out[11]: [('AE22YDHSBFYIP', datetime.date(1970, 12, 30), 5, 10, 0, '1551803542', 'Start and Run
          ('AE22YDHSBFYIP', datetime.date(1970, 12, 30), 5, 9, 0, '1551802538', 'Start and Run a
          ('ATVPDKIKX0DER', datetime.date(1995, 6, 19), 4, 19, 18, '0898624932', 'The Power of M
          ('AH7OKBE1Z35YA', datetime.date(1995, 6, 23), 5, 4, 4, '0521469112', 'Invention and Ev
          ('ATVPDKIKX0DER', datetime.date(1995, 7, 14), 5, 0, 0, '0679722955', 'The Names (Vinta
          ('A102UKC71I5DU8', datetime.date(1995, 7, 18), 4, 2, 2, '0471114251', 'Bitter Winds ',
          ('A1HPIDTM9SRBLP', datetime.date(1995, 7, 18), 5, 0, 0, '0517887290', 'Fingerprints of
          ('A1HPIDTM9SRBLP', datetime.date(1995, 7, 18), 5, 0, 0, '1574531093', 'Fingerprints of
          ('ATVPDKIKX0DER', datetime.date(1995, 7, 18), 5, 1, 0, '0962344788', 'Heavy Light', 66
          ('ATVPDKIKX0DER', datetime.date(1995, 7, 18), 5, 1, 1, '0195069056', "Albion's Seed",
```

```
In [12]: %%time
         %%sql
         SELECT
             customer_id, review_date, review_rating, product_id, product_title
         FROM
             customer_reviews_col
         WHERE
             customer_id ='A27T7HVDXA3K2A' AND
             product_title LIKE '%Dune%' AND
             review_date >= '1998-01-01' AND
             review_date <= '1998-12-31';
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
5 rows affected.
CPU times: user 4.7 ms, sys: 0 ns, total: 4.7 ms
Wall time: 246 ms
```

```
Out[12]: [('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0399128964', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '044100590X', 'Dune'),
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0441172717', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '0881036366', 'Dune (Dune Chronicles
          ('A27T7HVDXA3K2A', datetime.date(1998, 4, 10), 5, '1559949570', 'Dune Audio Collection
```

1.5 Conclusion: We can see that the columnar storage is faster !

```
In [13]: %%time
         %%sql
         SELECT product_title, avg(review_rating)
         FROM customer_reviews_col
         WHERE review_date >= '1995-01-01'
             AND review_date <= '1998-12-31'
         GROUP BY product_title
         ORDER by product_title
         LIMIT 20;
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
20 rows affected.
CPU times: user 4.93 ms, sys: 123 ţs, total: 5.05 ms
Wall time: 548 ms
```

```
Out[13]: [('!Yo!', Decimal('4.7500000000000000')),
          ("# 1's", Decimal('4.2682926829268293')),
          ('#1 Record/Radio City', Decimal('5.0000000000000000')),
          ("#1 Soul Hits Of The 60's, Vol. 3", Decimal('5.0000000000000000')),
          ("#1's", Decimal('4.2409638554216867')),
          ("'58 Miles Featuring Stella by Starlight", Decimal('5.0000000000000000')),
          ("'Bout It", Decimal('3.0000000000000000')),
          ("'Round Midnight", Decimal('5.0000000000000000')),
          ("'Salem's Lot", Decimal('4.6333333333333333')),
          ("'The Moon by Whale Light", Decimal('4.2500000000000000')),
          ("'The Radical Reformation (3rd ed)", Decimal('5.0000000000000000')),
          ("'The Verilog Hardware Description Language (with CD-Rom)", Decimal('3.66666666666666
          ("'Til It Kills", Decimal('5.0000000000000000')),
          ("'Til Shiloh", Decimal('5.0000000000000000')),
          ("'Til Their Eyes Shine (The Lullaby Album)", Decimal('5.0000000000000000')),
          ("'night, Mother ", Decimal('5.0000000000000000')),
          ("(I'm) Stranded", Decimal('5.0000000000000000')),
          ('(Sick) ', Decimal('4.0000000000000000')),
          ("(What's The Story) Morning Glory?", Decimal('4.1538461538461538')),
          ("(Who's Afraid Of?) The Art of Noise!", Decimal('3.3333333333333333'))]
```

```
In [14]: %%time
         %%sql
         SELECT product_title, avg(review_rating)
         FROM customer_reviews_row
         WHERE review_date >= '1995-01-01'
             AND review_date <= '1998-12-31'
         GROUP BY product_title
         ORDER by product_title
         LIMIT 20;
```

```
 * postgresql://student:***@127.0.0.1:5432/reviews
20 rows affected.
CPU times: user 3.99 ms, sys: 522 ţs, total: 4.52 ms
Wall time: 1.21 s
```

```
Out[14]: [('!Yo!', Decimal('4.7500000000000000')),
          ("# 1's", Decimal('4.2682926829268293')),
          ('#1 Record/Radio City', Decimal('5.0000000000000000')),
          ("#1 Soul Hits Of The 60's, Vol. 3", Decimal('5.0000000000000000')),
          ("#1's", Decimal('4.2409638554216867')),
          ("'58 Miles Featuring Stella by Starlight", Decimal('5.0000000000000000')),
          ("'Bout It", Decimal('3.0000000000000000')),
          ("'Round Midnight", Decimal('5.0000000000000000')),
          ("'Salem's Lot", Decimal('4.6333333333333333')),
          ("'The Moon by Whale Light", Decimal('4.2500000000000000')),
          ("'The Radical Reformation (3rd ed)", Decimal('5.0000000000000000')),
          ("'The Verilog Hardware Description Language (with CD-Rom)", Decimal('3.66666666666666
          ("'Til It Kills", Decimal('5.0000000000000000')),
          ("'Til Shiloh", Decimal('5.0000000000000000')),
          ("'Til Their Eyes Shine (The Lullaby Album)", Decimal('5.0000000000000000')),
          ("'night, Mother ", Decimal('5.0000000000000000')),
          ("(I'm) Stranded", Decimal('5.0000000000000000')),
          ('(Sick) ', Decimal('4.0000000000000000')),
          ("(What's The Story) Morning Glory?", Decimal('4.1538461538461538')),
          ("(Who's Afraid Of?) The Art of Noise!", Decimal('3.3333333333333333'))]
```