# L1 E3 - Columnar Vs Row Storage - Solution

May 23, 2021

## 1 Exercise 03 - Columnar Vs Row Storage - Solution

In [ ]:

- 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 [ ]: %load_ext sql

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

#### 1.1.1 Create the database

In [ ]: !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
```

#### 1.1.2 Connect to the database

In [ ]:
```
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)
```

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

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

In [ ]:
```
%%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)[]
)
```

In [ ]:
```
%%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;
```

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

In [ ]:
```
%%sql

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

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

In [ ]:
```
%%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');
```

In [ ]:
```
%%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;
```

### 1.4 Step 3: Compare perfromamce

In [ ]:
```
%%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';
```

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

In [ ]:
```
%%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';
```

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

In [ ]:
```
%%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;
```

In [ ]:
```
%%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;
```