

PENDAHULUAN

Pada praktikum ini, kita akan belajar bagaimana mengambil dan menganalisis data dari sistem basis data NoSQL, yaitu MongoDB dan Cassandra, menggunakan dataset *Warehouse_Retails*. Tujuan dari kegiatan ini adalah melatih kemampuan dalam menyusun query untuk mencari informasi penting dari data penjualan, seperti produk apa yang paling laku, penjualan terbanyak di bulan tertentu, hingga perbandingan antar jenis produk.

Langkah pertama yang dilakukan adalah memasukkan data ke MongoDB dalam koleksi bernama *Retails* dan ke Cassandra dalam tabel *warehouse_retails*. Setelah data berhasil dimasukkan, kita membuat 5 query pada satu tabel menggunakan fitur *Aggregation* di MongoDB. Misalnya, kita akan mengambil total penjualan per jenis produk, data produk tertentu di bulan tertentu, hingga mengurutkan penjualan tertinggi.

Dalam proses pengolahan data berbasis NoSQL, penting untuk memahami bahwa sistem seperti MongoDB dan Cassandra tidak dirancang untuk melakukan operasi join antar tabel seperti halnya sistem basis data relasional (SQL). Hal ini disebabkan oleh prinsip dasar NoSQL yang mengutamakan kecepatan dan efisiensi dalam penyimpanan serta pengambilan data dalam skala besar. Cassandra, misalnya, sepenuhnya mengandalkan desain berbasis query, sehingga tidak menyediakan kemampuan join antar tabel. Sementara MongoDB hanya menyediakan fitur terbatas seperti \$lookup, yang hanya bisa digunakan dalam konteks agregasi dan masih memiliki banyak batasan dibanding join pada SQL.

Karena itulah, ketika diperlukan analisis data dari beberapa tabel yang berbeda, proses penggabungan (join) data dilakukan di luar database, yaitu menggunakan Python dan library seperti Pandas dalam Jupyter Notebook. Pendekatan ini disebut sebagai integrasi data secara desentralisasi, karena proses penyatuan dan pemrosesan dilakukan di luar sistem database. Oleh karena itu, alasan penggunaan Python sebagai alat bantu integrasi data dalam praktikum ini adalah keputusan yang tepat dan sesuai dengan karakteristik serta keterbatasan dari sistem NoSQL yang digunakan.

Selain itu, proses *preprocessing* atau pembersihan dan penyesuaian data juga dilakukan sebelum analisis. Penjelasan lebih lengkap mengenai tahapan *preprocessing* tersebut dapat diakses melalui tautan berikut: [Penjelasan Preprocessing Data Warehouse Retails](#). Laporan ini akan berisi 10 query utama, masing-masing disertai hasil dan penjelasannya. Lima query dibuat langsung dari MongoDB, dan lima lainnya dibuat dari hasil penggabungan data di Python. Setiap query akan dianalisis dan divisualisasikan agar lebih mudah dipahami.

Download Tools:

Install Java SE Development Kit (JDK) versi 1.8.0_202

<https://www.oracle.com/id/java/technologies/javase/javase8-archive-downloads.html>

Install MongoDB (Community Edition)

dari <https://www.mongodb.com/try/download/community>

Install Apache Cassandra

dari <https://cassandra.apache.org/download/>

PRAKTIKUM MODUL

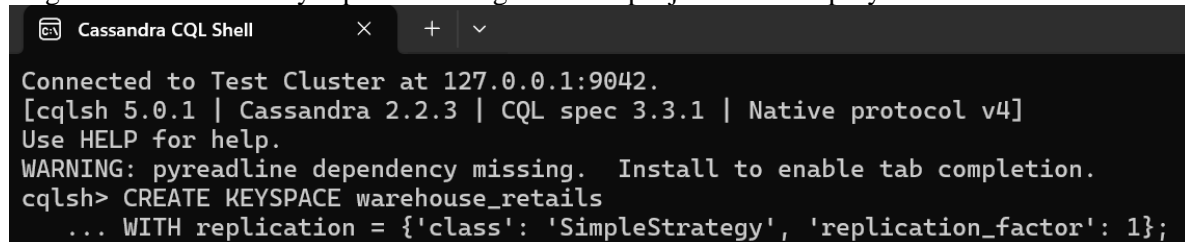
```
cqlsh> USE bigdata_ks
... ;
cqlsh:bigdata_ks> CREATE TABLE tweets (
...     id TEXT PRIMARY KEY,
...     user TEXT,
...     text TEXT,
...     created_at DATE,
...     likes INT,
...     retweets INT
Invalid syntax at line 7, char 4
^
retweets INT
cqlsh:bigdata_ks> );
SyntaxException: <ErrorMessage code=2000 [Syntax error in CQL query] message="line 1:0 no viable alternative at input ')' ([)]...)">
cqlsh:bigdata_ks> CREATE TABLE tweets (
...     id TEXT PRIMARY KEY,
...     "user" TEXT,
...     "text" TEXT,
...     created_at TIMESTAMP,
...     likes INT,
...     retweets INT
... );
cqlsh:bigdata_ks> INSERT INTO tweets (id, user, text, created_at, likes, retweets)
... VALUES ('11111', 'data_analyst', 'Cassandra scales well for Big Data!', '2023-10-04', 80, 15);
cqlsh:bigdata_ks>
cqlsh:bigdata_ks> -- Cari tweet dengan retweets > 20
cqlsh:bigdata_ks> SELECT * FROM tweets WHERE retweets > 20 ALLOW FILTERING;
Invalid syntax at line 1, char 47
^
SELECT * FROM tweets WHERE retweets > 20 ALLOW FILTERING;
cqlsh:bigdata_ks> INSERT INTO tweets (id, user, text, created_at, likes, retweets)
... VALUES ('11111', 'data_analyst', 'Cassandra scales well for Big Data!', '2023-10-04', 80, 15)
... ;
cqlsh:bigdata_ks> -- Cari tweet dengan retweets > 20
cqlsh:bigdata_ks> SELECT * FROM tweets WHERE retweets > 20 ALLOW FILTERING;
Invalid syntax at line 1, char 47
^
SELECT * FROM tweets WHERE retweets > 20 ALLOW FILTERING;
cqlsh:bigdata_ks> SELECT * FROM tweets
... ;

id      | created_at          | likes | retweets | text                                     | user
-----+-----+-----+-----+-----+-----
11111 | 2023-10-03 17:00:00+0000 | 80    | 15       | Cassandra scales well for Big Data!    | data_analyst
(1 rows)
```

DATASET DAN TOOLS

1. CASSANDRA

Gambar tersebut menunjukkan proses pembuatan keyspace warehouse_retails di Cassandra sebagai langkah awal untuk menyimpan dan mengelola data penjualan dalam proyek Warehouse Retails.



```
Cassandra CQL Shell
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.2.3 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing. Install to enable tab completion.
cqlsh> CREATE KEYSPACE warehouse_retails
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
```

Gambar tersebut menunjukkan perintah USE warehouse_retails yang digunakan untuk mengakses keyspace warehouse_retails sebagai konteks kerja utama dalam pengelolaan data pada proyek Warehouse Retails di Cassandra.



```
cqlsh> USE warehouse_retails
```

Setelah berhasil membuat keyspace warehouse_retails dan mengaktifkannya menggunakan perintah USE, langkah selanjutnya adalah mendefinisikan struktur tabel utama yang akan digunakan untuk menyimpan data warehouse dan retail. Pada tahap ini, dibuat tabel warehouse_retails dengan skema kolom yang menyesuaikan atribut dari dataset, seperti tahun, bulan, kode dan deskripsi barang, serta berbagai jenis tipe item dan data penjualan. Primary key ditetapkan dengan kombinasi (year, month) sebagai partition key dan item_code sebagai clustering key, yang memungkinkan pengelompokan data berdasarkan waktu dan identifikasi item.



```
CREATE KEYSPACE warehouse_retails WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'} AND durable_writes = true;

CREATE TABLE warehouse_retails.warehouse_retails (
  year int,
  month int,
  item_code text,
  item_description text,
  item_type_dunnage text,
  item_type_kegs text,
  item_type_liquor text,
  item_type_non_alcohol text,
  item_type_ref text,
  item_type_str_supplies text,
  item_type_wine text,
  item_type_wine2 text,
  retail_sales decimal,
  retail_transfers decimal,
  supplier_encoded text,
  warehouse_sales decimal,
  PRIMARY KEY ((year, month), item_code)
) WITH CLUSTERING ORDER BY (item_code ASC)
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}
AND compression = {'sstable_compression': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND dclocal_read_repair_chance = 0.1
```

Gambar ini menunjukkan proses impor data ke dalam tabel warehouse_retails di Cassandra menggunakan perintah COPY, di mana sebanyak 307.645 baris data dari file CSV berhasil dimuat ke dalam tabel dengan kecepatan penulisan sekitar 5526 baris per detik.

Gambar ini menunjukkan bahwa data telah berhasil diimpor ke dalam tabel `warehouse_retails`, ditunjukkan dengan tampilnya beberapa baris data yang mencerminkan isi dari file CSV yang dimuat sebelumnya.

Laporan Praktikum Big Data

2. MONGODB

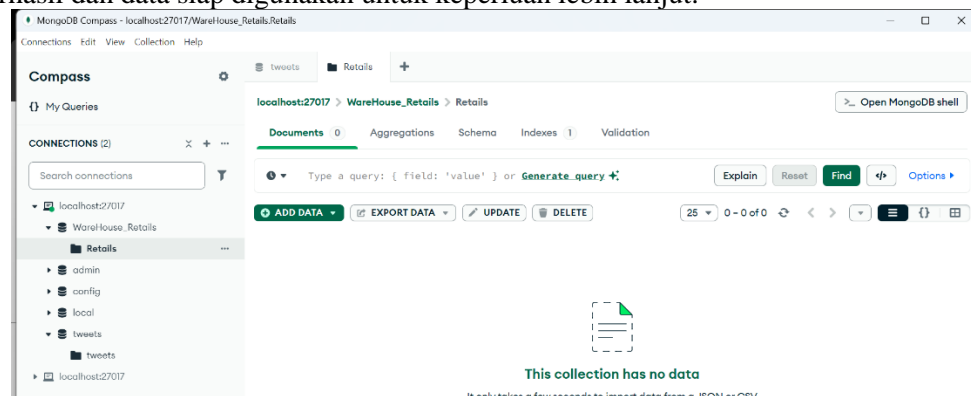
Gambar ini menunjukkan langkah awal proses impor data ke MongoDB, di mana pengguna diminta untuk memasukkan connection string untuk menghubungkan ke database, dengan nilai default `mongodb://localhost/`.

```
Please enter a MongoDB connection string (Default: mongodb://localhost/): |
```

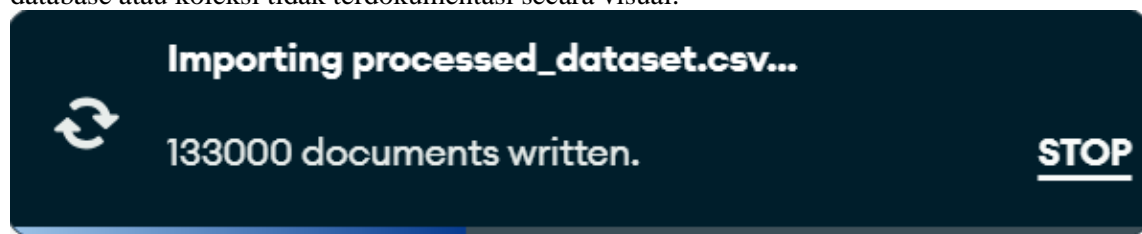
Untuk mengimpor dataset ke MongoDB, langkah pertama yang dilakukan adalah membuka MongoDB Compass dan menyambungkannya ke server lokal menggunakan connection string default ``mongodb://localhost:27017/``. Setelah berhasil terhubung, pengguna kemudian membuat database baru dengan nama ``WareHouse_Retails`` dan koleksi bernama ``Retails``.

Setelah database dan koleksi dibuat, pengguna masuk ke koleksi ``Retails``, di mana pada awalnya belum terdapat data apapun. Proses impor dilakukan dengan mengklik tombol "ADD DATA" lalu memilih opsi "Import File". Pada tahap ini, pengguna memilih file ``processed_dataset.csv`` dari direktori lokal, memastikan format file yang dipilih adalah CSV, serta mencentang opsi header agar MongoDB dapat mengenali baris pertama sebagai nama kolom. Setelah semua pengaturan sesuai, tombol "IMPORT" ditekan untuk memulai proses.

MongoDB Compass kemudian akan menampilkan progres impor dalam bentuk notifikasi yang menunjukkan jumlah dokumen yang berhasil ditulis ke dalam koleksi, seperti yang terlihat pada gambar bahwa sebanyak 133.000 dokumen telah berhasil diimpor. Setelah proses selesai, data akan langsung muncul di tab "Documents" dalam koleksi ``Retails``, yang menandakan bahwa proses impor telah berhasil dan data siap digunakan untuk keperluan lebih lanjut.

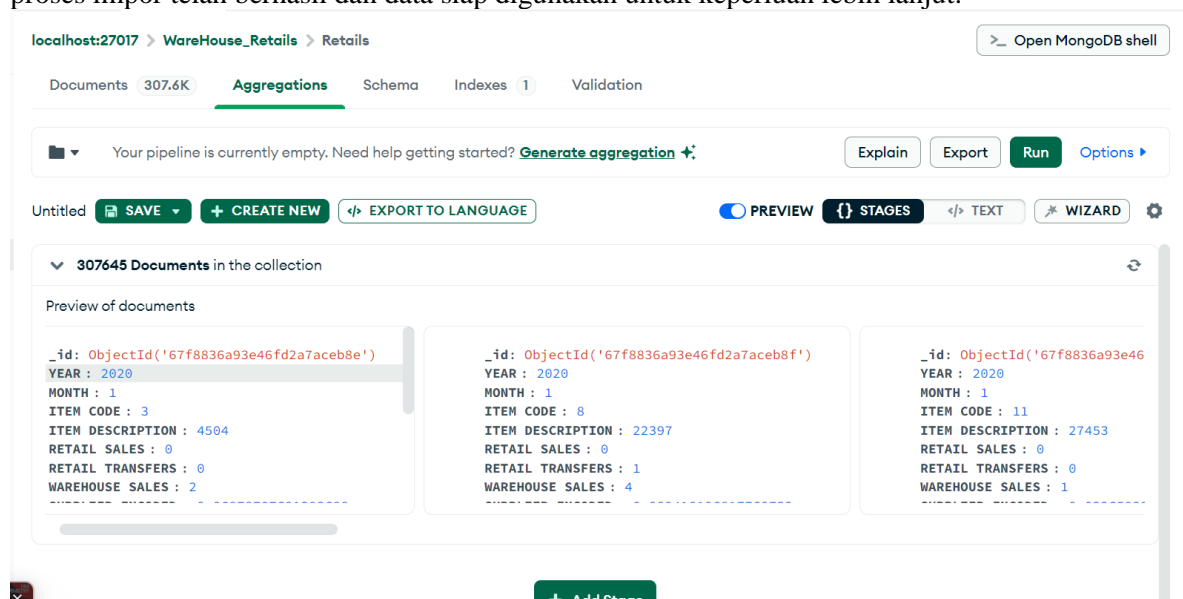


Gambar ini memperlihatkan bahwa proses impor file `processed_dataset.csv` ke MongoDB sedang berlangsung dan telah berhasil menulis 133.000 dokumen, yang menandakan bahwa data dari file CSV sedang dimasukkan ke dalam koleksi MongoDB meskipun langkah awal seperti pemilihan database atau koleksi tidak terdokumentasi secara visual.



Untuk mengimpor dataset ke MongoDB, langkah pertama yang dilakukan adalah membuka MongoDB Compass dan menyambungkannya ke server lokal menggunakan connection string default

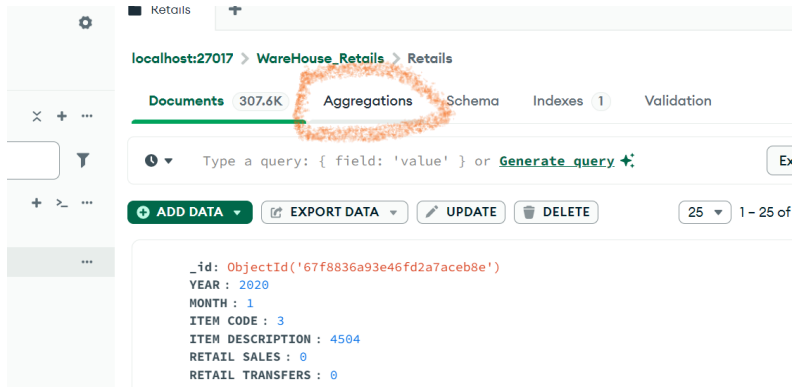
`mongodb://localhost:27017/`. Setelah berhasil terhubung, pengguna kemudian membuat database baru dengan nama `WareHouse_Retails` dan koleksi bernama `Retails`. Setelah database dan koleksi dibuat, pengguna masuk ke koleksi `Retails`, di mana pada awalnya belum terdapat data apapun. Proses impor dilakukan dengan mengklik tombol "ADD DATA" lalu memilih opsi "Import File". Pada tahap ini, pengguna memilih file `processed_dataset.csv` dari direktori lokal, memastikan format file yang dipilih adalah CSV, serta mencentang opsi header agar MongoDB dapat mengenali baris pertama sebagai nama kolom. Setelah semua pengaturan sesuai, tombol "IMPORT" ditekan untuk memulai proses. MongoDB Compass kemudian akan menampilkan progres impor dalam bentuk notifikasi yang menunjukkan jumlah dokumen yang berhasil ditulis ke dalam koleksi, seperti yang terlihat pada gambar bahwa sebanyak 133.000 dokumen telah berhasil diimpor. Setelah proses selesai, data akan langsung muncul di tab "Documents" dalam koleksi `Retails`, yang menandakan bahwa proses impor telah berhasil dan data siap digunakan untuk keperluan lebih lanjut.



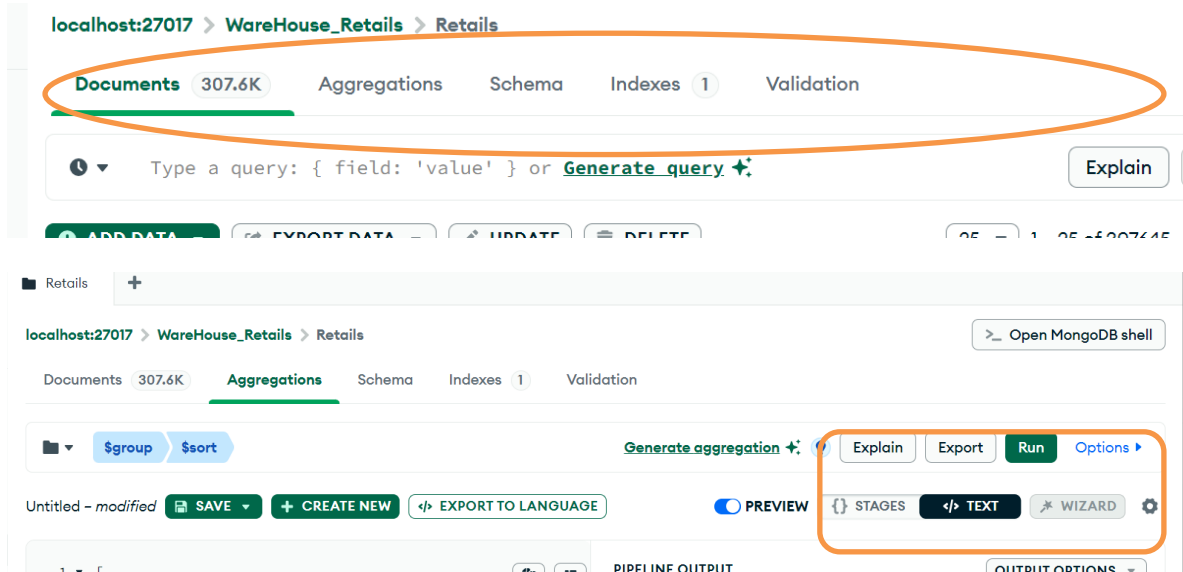
Query dan hasil

1. QUERY DALAM 1 TABEL

Setelah berhasil import dataset, kemudian pilih aggregation untuk melakukan eksekusi query dalam proses analisis data.



Terdapat dua cara yang paling umum digunakan yaitu menggunakan formula bar atau menggunakan text.



Berikut adalah 6 query dari 1 tabel.

1. Mengidentifikasi produk paling laku di retail.

```
1 [
2   { "$sort": { "RETAIL SALES": -1 } },
3   { "$project": { "ITEM DESCRIPTION": 1, "RETAIL SALES": 1 } }
4 ]
```

Sort	Sproject	Edit
------	----------	------

ALL RESULTS

_id: ObjectId('67f8836a93e46fd2a7ad1e20') ITEM DESCRIPTION : 16995 RETAIL SALES : 2739
_id: ObjectId('67f8836c93e46fd2a7ad8b2b') ITEM DESCRIPTION : 31764 RETAIL SALES : 1816.49
_id: ObjectId('67f8837993e46fd2a7b0f18e') ITEM DESCRIPTION : 31764 RETAIL SALES : 1752.45
_id: ObjectId('67f8837793e46fd2a7b09888') ITEM DESCRIPTION : 31764 RETAIL SALES : 1710.8
_id: ObjectId('67f8837193e46fd2a7af023d') ITEM DESCRIPTION : 31764 RETAIL SALES : 1616.6
_id: ObjectId('67f8837a93e46fd2a7b15d8c') ITEM DESCRIPTION : 31764 RETAIL SALES : 1615.23
_id: ObjectId('67f8837793e46fd2a7b074c9') ITEM DESCRIPTION : 9301 RETAIL SALES : 1494
_id: ObjectId('67f8836b93e46fd2a7ad38d5') ITEM DESCRIPTION : 31764 RETAIL SALES : 1405.17

2. Total penjualan berdasarkan jenis barang (all ITEM TYPE_)

```

1 [
2   {
3     "$group": {
4       "_id": null,
5       "DUNNAGE": { "$sum": "$ITEM TYPE_DUNNAGE" },
6       "KEGS": { "$sum": "$ITEM TYPE_KEGS" },
7       "LIQUOR": { "$sum": "$ITEM TYPE_LIQUOR" },
8       "NON_ALCOHOL": { "$sum": "$ITEM TYPE_NON-ALCOHOL" },
9       "REF": { "$sum": "$ITEM TYPE_REF" },
10      "SUPPLIES": { "$sum": "$ITEM TYPE_STR_SUPPLIES" },
11      "WINE": { "$sum": "$ITEM TYPE_WINE" },
12      "WINE2": { "$sum": "$ITEM TYPE_Wine" }
13    }
14  }
15 ]

```

```

_id: null
DUNNAGE : 34
KEGS : 3168
LIQUOR : 21333
NON_ALCOHOL : 630
REF : 34
SUPPLIES : 132
WINE : 60849
WINE2 : 0

```

3. Jumlah Item Tiap Bulan

```

1 [ { "$group": { "_id": "$MONTH", "TOTAL_ITEMS": { "$sum": 1 } } } ]
2

```


PIPELINE OUTPUT		OUTPUT OPTIONS ▾
Sample of 8 documents		
_id: 9 TOTAL_ITEMS : 23153		
_id: 1 TOTAL_ITEMS : 11987		
_id: 3 TOTAL_ITEMS : 11523		
_id: 8 TOTAL_ITEMS : 13590		
_id: 6 TOTAL_ITEMS : 13628		
_id: 12 TOTAL_ITEMS : 23		
_id: 7 TOTAL_ITEMS : 24212		
_id: 10 TOTAL_ITEMS : 1884		

4. Rata-rata RETAIL_SALES per Item Type

```
1 [ { "$group": { "_id": "$ITEM_TYPE_WINE", "AVG_RETAIL_SALES": { "$avg": "$RETAIL_SALES" } } } ]
2
```

PIPELINE OUTPUT		OUTPUT OPTIONS ▾
Sample of 2 documents		
_id: 0 AVG_RETAIL_SALES : 12.354211131261014		
_id: 1 AVG_RETAIL_SALES : 3.9216665844960477		

5. Menghitung rata-rata RETAIL SALES, RETAIL TRANSFERS, dan WAREHOUSE SALES per tahun

```
1 ▾ [
2 ▾ {
3 ▾   "$group": {
4     "_id": "$YEAR",
5     "AVG_RETAIL_SALES": { "$avg": "$RETAIL_SALES" },
6     "AVG_RETAIL_TRANSFERS": { "$avg": "$RETAIL_TRANSFERS" },
7     "AVG_WAREHOUSE_SALES": { "$avg": "$WAREHOUSE_SALES" }
8   }
9 },
10 ▾ {
11   "$sort": { "_id": 1 }
12 }
13 ]
```

PIPELINE OUTPUT

OUTPUT OPTIONS ▾

Sample of 2 documents

```
_id: 2017
AVG_RETAIL_SALES : 6.764837844164731
AVG_RETAIL_TRANSFERS : 6.620180253182792
AVG_WAREHOUSE_SALES : 25.478158212234167
```

```
_id: 2020
AVG_RETAIL_SALES : 7.795386916140059
AVG_RETAIL_TRANSFERS : 7.523619987856163
AVG_WAREHOUSE_SALES : 29.708720624283174
```

6. Dalam Cassandra, query berikut untuk data penjualan dari warehouse dan retail untuk periode Januari 2020.

```
cqlsh:warehouse_retails> SELECT year, retail_sales, retail_transfers, warehouse_sales
... FROM warehouse_retails
... WHERE year = 2020 AND month = 1;
```

year	retail_sales	retail_transfers	warehouse_sales
2020	5.9	11.0	2.0
2020	1.56	2.0	4.0
2020	0.0	0.0	1.0
2020	0.0	0.0	3.0
2020	0.0	0.0	1.0
2020	0.0	0.0	1.0
2020	23.4	18.0	2.0
2020	0.0	0.0	1.0
2020	0.59	0.0	0.0
2020	0.0	0.0	1.0
2020	0.0	0.0	3.0
2020	0.0	0.0	2.0
2020	0.17	2.0	20.0
2020	0.0	0.0	4.0
2020	0.0	0.0	1.0
2020	0.16	0.0	2.0
2020	0.0	0.0	4.0
2020	2.05	4.0	3.0
2020	0.0	0.0	1.0
2020	0.0	0.0	6.0
2020	0.0	0.0	1.0
2020	2.72	0.0	0.0
2020	0.64	0.0	0.0
2020	0.41	0.0	0.0
2020	0.4	0.0	0.0
2020	15.85	18.5	31.0
2020	0.0	0.0	3.0
2020	0.32	0.0	0.0
2020	0.42	1.0	4.0
2020	0.41	0.0	0.0
2020	0.0	0.0	2.0
2020	7.5	7.0	9.0
2020	0.0	0.0	2.0
2020	0.16	1.0	0.0
2020	0.08	0.0	1.0
2020	0.0	0.0	3.0
2020	0.24	0.0	1.0
2020	0.08	0.0	4.0
2020	0.0	0.0	9.0

IMPORT DATASET DALAM 3 TABEL

1. Membuatkan keyspace baru yaitu warehouse_retail_intothree

```
cqlsh> CREATE KEYSPACE warehouse_retail_intothree
... WITH replication = {
...     'class': 'SimpleStrategy',
...     'replication_factor': 1
... };
```

```
cqlsh> use warehouse_retail_intothree;
cqlsh:warehouse_retail_intothree> CREATE TABLE retails (
...     year int,
...     month int,
...     item_code int,
...     retail_sales float,
...     retail_transfers float,
...     warehouse_sales float,
...     supplier_encoded float,
...     PRIMARY KEY ((item_code), year, month)
... );
cqlsh:warehouse_retail_intothree> CREATE TABLE items (
...     item_code int PRIMARY KEY,
...     item_description text,
...     item_type_dunnage float,
...     item_type_kegs float,
...     item_type_liquor float,
...     item_type_non_alcohol float,
...     item_type_ref float,
...     item_type_str_supplies float,
...     item_type_wine float,
...     item_type_wine_dup float
... );
cqlsh:warehouse_retail_intothree> CREATE TABLE suppliers (
...     supplier_encoded float PRIMARY KEY
...     -- Tambah nama supplier kalau tersedia, misal: supplier_name text
... );
```

```
cqlsh:warehouse_retail_intothree> CREATE TABLE retails_full_joined (
...     year int,
...     month int,
...     item_code int,
...     retail_sales float,
...     retail_transfers float,
...     warehouse_sales float,
...     supplier_encoded float,
...     item_description text,
...     item_type_dunnage float,
...     item_type_kegs float,
...     item_type_liquor float,
...     item_type_non_alcohol float,
...     item_type_ref float,
...     item_type_str_supplies float,
...     item_type_wine float,
...     item_type_wine_dup float,
...     PRIMARY KEY ((item_code), year, month)
... );
```

```

cqlsh:warehouse_retail_intothree> COPY retails (
...     year,
...     month,
...     item_code,
...     retail_sales,
...     retail_transfers,
...     warehouse_sales,
...     supplier_encoded
... )
... FROM 'C:/Users/ASUS/05_BIDA/retails.csv'
... WITH HEADER = TRUE;
Processed 300000 rows; Write: 9510.81 rows/ss
307645 rows imported in 27.279 seconds.
cqlsh:warehouse_retail_intothree>
cqlsh:warehouse_retail_intothree> COPY items (
...     item_code,
...     item_description,
...     item_type_dunnage,
...     item_type_kegs,
...     item_type_liquor,
...     item_type_non_alcohol,
...     item_type_ref,
...     item_type_str_supplies,
...     item_type_wine,
...     item_type_wine_dup
... )
... FROM 'C:/Users/ASUS/05_BIDA/items.csv'
... WITH HEADER = TRUE;
Processed 30000 rows; Write: 12309.51 rows/s
35165 rows imported in 3.314 seconds.
cqlsh:warehouse_retail_intothree> COPY suppliers (
...     supplier_encoded
... )
... FROM 'C:/Users/ASUS/05_BIDA/suppliers.csv'
... WITH HEADER = TRUE;
260 rows imported in 0.713 seconds.
cqlsh:warehouse_retail_intothree> COPY retails_full_joined (
...     year,
...     month,
...     item_code,
...     retail_sales,
...     retail_transfers,
...     warehouse_sales,
...     supplier_encoded,
...     item_description,
...     item_type_dunnage,
...     item_type_kegs,
...     item_type_liquor,
...     item_type_non_alcohol,
...     item_type_ref,
...     item_type_str_supplies,
...     item_type_wine,
...     item_type_wine_dup
... )
... FROM 'C:/Users/ASUS/05_BIDA/retails_full_joined.csv'
... WITH HEADER = TRUE;
Processed 320000 rows; Write: 7078.80 rows/s
325207 rows imported in 42.734 seconds.

```

QUERY DARI 3 TABEL

1. Jumlah transaksi bulan Januari 2020

```

cqlsh:warehouse_retail_intothree> SELECT COUNT(*) FROM retails_full_joined
... WHERE year = 2020 AND month = 1 ALLOW FILTERING;

count
-----
11987

(1 rows)

```

2. Jumlah Produk dari supplier tertentu

Kolom supplier_encoded bukan bagian dari primary key, jadi untuk bisa pakai WHERE supplier_encoded pada query ini, maka perlu secondary index dulu.

```
cqlsh:warehouse_retail_intothree> CREATE INDEX ON retails_full_joined (supplier_encoded);
cqlsh:warehouse_retail_intothree> SELECT COUNT(*) FROM retails_full_joined
... WHERE supplier_encoded = 0.0082010109054267;

count
-----
2290

(1 rows)
```

3. Total Penjualan Retail per Jenis Produk Wine, untuk mengetahui berapa total penjualan (retail) untuk item yang merupakan produk wine (item_type_wine = 1.0).

```
cqlsh:warehouse_retail_intothree> CREATE INDEX IF NOT EXISTS ON retails_full_joined (item_type_wine);
cqlsh:warehouse_retail_intothree> SELECT item_code, item_description, item_type_wine, warehouse_sales
... FROM retails_full_joined
... WHERE item_type_wine = 1.0 ALLOW FILTERING;
```

item_code	item_description	item_type_wine	warehouse_sales
4317	7048	1	1
14340	10432	1	2
18417	22662	1	1
18417	22662	1	2
1584	5250	1	2
1584	5250	1	2
1584	5250	1	1
1584	5250	1	3
1584	5250	1	3
1584	5250	1	2
1584	5250	1	3
1584	5250	1	2
1584	5250	1	1
1584	5250	1	2
1584	5250	1	1
1584	5250	1	1
1584	5250	1	1
1584	5250	1	2
1584	5250	1	2
1584	5250	1	1
1584	5250	1	1
1584	5250	1	1
1584	5250	1	2

4. Tampilkan Semua Produk dari supplier_encoded tertentu

```
cqlsh:warehouse_retail_intothree> CREATE INDEX IF NOT EXISTS ON retails_full_joined (supplier_encoded);
cqlsh:warehouse_retail_intothree> SELECT item_code, item_description, supplier_encoded
... FROM retails_full_joined
... WHERE supplier_encoded = 0.0226592338572055 ALLOW FILTERING;
```

item_code	item_description	supplier_encoded
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
28572	28073	0.022659
13952	3615	0.022659
13952	3615	0.022659
13952	3615	0.022659

5. Mengetahui Produk yang Hanya Dijual di Warehouse

Beberapa produk mungkin tidak tersedia untuk penjualan retail tapi masih aktif di gudang. Ini bisa menunjukkan kebutuhan stok ulang atau produk yang masih tersimpan tapi belum dijual ke pelanggan.

```
cqlsh:warehouse_retail_intothree> SELECT item_code, item_description, retail_sales, warehouse_sales
... FROM retails_full_joined
... WHERE item_code IN (3, 8, 11, 13, 59) AND year = 2020 AND month = 1;
```

item_code	item_description	retail_sales	warehouse_sales
3	4504	0	2
8	22397	0	4
11	27659	0	1
13	28185	0	1
59	17438	0	2