Source Code ที่เกี่ยวข้อง

โยนไฟล์จากเครื่องเราเข้า Ubuntu

- **Docker Compose**

scp -i kunakorn.pem docker-compose.yml ubuntu@ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:/home/ubuntu

- **Source 3 : Your design (relational database) (topic3)**

scp -i kunakorn.pem source3_data.py ubuntu@ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:/home/ubuntu

- **Streamlit Dashboard (จริงๆอยู่ในขั้นตอนสุดท้าย แต่ผมเอามาสรุปทีเดียวครับ)**

scp -i kunakorn.pem dashboard.py ubuntu@ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:/home/ubuntu

เข้าใช้งาน Ubuntu Server

ssh -i kunakorn.pem ubuntu@ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com

Install Docker บน Ubuntu

sudo apt-get update

sudo apt-get install -y ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \

  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io

Run Docker-Compose จากไฟล์ docker-compose.yml ที่โยนเข้าไป

sudo docker-compose up -d

install kafka บน ubuntu

Update the apt package list

sudo apt-get update

Install Java (Kafka requires Java)

sudo apt-get install openjdk-11-jdk -y

Download and install Kafka

wget https://archive.apache.org/dist/kafka/2.8.0/kafka_2.13-2.8.0.tgz

tar -xvzf kafka_2.13-2.8.0.tgz

cd kafka_2.13-2.8.0

Install Python3 and pip

sudo apt-get install python3 python3-pip -y

Kafka Python client library

pip3 install kafka-python

#Run Code source3

python3 /home/ubuntu/source3_data.py

b. Kafka system สร้าง Topic (ตาม Requirement , สำหรับ Topic9-11 ผมวิเคราะห์ข้อมูลเพิ่มเติมครับ)

sudo docker exec -it broker1 bash

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_1 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_2 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_3 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_4 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_5 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_6 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_7 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_8 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_9 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_10 --partitions 5 --replication-factor 3

kafka-topics --bootstrap-server broker1:29092 --create --topic topic_11 --partitions 5 --replication-factor 3

a. Data source 3 sources [1]

สร้าง Source1/Topic1 Datagen-pageview

```
curl -X POST -H "Content-Type: application/json" \
-d '{
  "name": "datagen-pageviews",
  "config": {
    "name": "datagen-pageviews",
    "connector.class": "io.confluent.kafka.connect.datagen.DatagenConnector",
    "key.converter": "org.apache.kafka.connect.storage.StringConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter.schemas.enable": "false",
    "kafka.topic": "topic_1",
    "max.interval": "1000",
    "quickstart": "pageviews",
    "interval.type": "random",
    "interval.range.min": "1",
    "interval.range.max": "1000"
  }
}' http://localhost:8083/connectors
```

สร้าง Source2/Topic2 Datagen-users

```
curl -X POST -H "Content-Type: application/json" \
-d '{
  "name": "datagen-users",
  "config": {
    "name": "datagen-users",
    "connector.class": "io.confluent.kafka.connect.datagen.DatagenConnector",
    "key.converter": "org.apache.kafka.connect.storage.StringConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter.schemas.enable": "false",
    "kafka.topic": "topic_2",
    "max.interval": "1000",
    "quickstart": "users",
    "interval.type": "random",
    "interval.range.min": "1",
    "interval.range.max": "1000"
  }
}' http://localhost:8083/connectors
```

สร้าง Source3/Topic3 จาก python file source3_data จากไฟล์ที่โยนเข้าไปตั้งแต่แรก

```
python3 /home/ubuntu/source3_data.py
```

c. ksqlDB operation [3]

#เข้าใช้งาน ksqldb ผ่าน ubuntu

sudo docker exec -it ksqldb-cli ksql http://ksqldb-server:8088

สร้าง Topic1 (stream)

```
CREATE STREAM topic1_stream (
    userid VARCHAR,
    pageid VARCHAR,
    viewtime BIGINT
) WITH (
    KAFKA_TOPIC='topic_1',
    VALUE_FORMAT='JSON'
);
```

#ดูข้อมูล

```
SELECT * FROM topic1_stream EMIT CHANGES;
```

```
+---------------------------------+---------------------------------+---------------------------------+
|USERID                           |PAGEID                           |VIEWTIME                         |
+---------------------------------+---------------------------------+---------------------------------+
|User_5                           |Page_25                          |90121                            |
|User_7                           |Page_91                          |90131                            |
|User_1                           |Page_99                          |90141                            |
|User_9                           |Page_32                          |90151                            |
|User_3                           |Page_27                          |90161                            |
```

สร้าง Topic 2 (stream)

```
CREATE TABLE topic2_stream (
   userid VARCHAR PRIMARY KEY,
   regionid VARCHAR,
   gender VARCHAR,
   registertime BIGINT
) WITH (
   KAFKA_TOPIC = 'topic_2',
   VALUE_FORMAT = 'JSON'
);
```

#ดูข้อมูล

```
SELECT * from topic2_stream EMIT CHANGES;
```

```
+-------------------------+-------------------------+-------------------------+-------------------------+
|USERID                   |REGIONID                 |GENDER                   |REGISTERTIME             |
+-------------------------+-------------------------+-------------------------+-------------------------+
|User_5                   |Region_1                 |FEMALE                   |1494777042498            |
|User_1                   |Region_3                 |FEMALE                   |1494573136138            |
|User_5                   |Region_8                 |OTHER                    |1514955962454            |
|User_9                   |Region_6                 |MALE                     |1489908326601            |
```

สร้าง Topic3 (Table)

```
CREATE TABLE topic3_table (
   Region_id INT PRIMARY KEY,
   Region_name VARCHAR,
   Population BIGINT,
   Area_size INT
) WITH (
   KAFKA_TOPIC='topic_3',
   VALUE_FORMAT='JSON'
);
```

#ตรวจสอบข้อมูลใน topic_3 ออกจากหน้า ksqldb ก่อน แล้วรันโค้ดด้านล่าง

```
sudo docker exec -it broker1 /bin/bash
kafka-console-consumer --bootstrap-server broker1:29092 --topic topic_3 --from-beginning
```

ถ้าข้อมูลไม่ขึ้น Run python อันนี้ใหม่ โดยออกจากหน้า ksqldb ก่อน
#สร้าง Source3/Topic3 จาก python file source3_data ที่โยนเข้าไปตั้งแต่แรก

```
python3 /home/ubuntu/source3_data.py
```

i. Clean or transform data (topic4)

สร้าง Topic4 (Stream)

```
CREATE STREAM users_formatted WITH (
    KAFKA_TOPIC='topic_4',
    VALUE_FORMAT='JSON'
) AS
SELECT
    userid AS UserId,
    regionid AS RegionId,
    gender AS Gender,
    TIMESTAMPTOSTRING(registertime, 'yyyy-MM-dd HH:mm:ss') AS RegisterTimeFormatted
FROM
    topic2_stream
EMIT CHANGES;
```

#ดูข้อมูล

```
SELECT * FROM users_formatted EMIT CHANGES;
```

```
+-------------------------+-------------------------+-------------------------+-------------------------+
|USERID                   |REGIONID                 |GENDER                   |REGISTERTIMEFORMATTED    |
+-------------------------+-------------------------+-------------------------+-------------------------+
|User_1                   |Region_3                 |FEMALE                   |2017-05-12 07:12:16      |
|User_5                   |Region_1                 |FEMALE                   |2017-05-14 15:50:42      |
|User_5                   |Region_8                 |OTHER                    |2018-01-03 05:06:02      |
|User_9                   |Region_6                 |MALE                     |2017-03-19 07:25:26      |
```

ii. Aggregation (join + group by) (topic5)

สร้าง Topic5 (Stream)

```
CREATE STREAM Consolidate_Stream WITH (
   KAFKA_TOPIC = 'topic_5',
   VALUE_FORMAT = 'JSON'
) AS
SELECT
    p.userid AS UserId,
    p.pageid AS PageId,
    u.RegionId AS RegionId,
    u.Gender AS Gender,
    r.Region_id AS RegionIdInTable,
    r.Region_name AS RegionName,
    r.Population AS Population,
    r.Area_size AS AreaSize,
    p.viewtime AS ViewTime,
    (p.viewtime * 100.0 / r.Population) AS ViewPercentage
FROM topic1_stream p
LEFT JOIN users_formatted u ON p.userid = u.UserId
LEFT JOIN topic3_table r ON CAST(SUBSTRING(u.RegionId, 8) AS INT) = r.Region_id
EMIT CHANGES;
```

#ดูข้อมูล

```
SELECT * FROM Consolidate_Stream EMIT CHANGES;
```

REGIONIDINTABLE	USERID	PAGEID	REGIONID	GENDER	REGIONNAME	POPULATION	AREASIZE	VIEWTIME	VIEWPERCENTAGE
6	User_1	Page_15	Region_6	FEMALE	Songkhla	1200000	1600	38761	null
9	User_6	Page_28	Region_9	MALE	Rayong	1000000	1400	38771	3.8771000000000000
7	User_1	Page_91	Region_7	FEMALE	Surat Thani	1800000	2000	38781	2.1545000000000000
7	User_2	Page_88	Region_7	FEMALE	Surat Thani	1800000	2000	38791	null
3	User_5	Page_69	Region_3	OTHER	Phuket	1000000	1200	38801	3.8801000000000000
4	User_6	Page_11	Region_4	MALE	Khon Kaen	1500000	1800	38811	2.5874000000000000

iii. Windows

1. Tumbling (topic6)

สร้าง Topic6 (Table)

```
CREATE TABLE PageViews_Tumbling WITH (
   KAFKA_TOPIC = 'topic_6',
   VALUE_FORMAT = 'JSON',
   PARTITIONS = 5,
   REPLICAS = 3
) AS
SELECT
   RegionId,
   LATEST_BY_OFFSET(RegionName) AS RegionName,
   COUNT(*) AS ViewCount,
   WINDOWSTART AS StartWindow,
   WINDOWEND AS EndWindow
FROM Consolidate_Stream
WINDOW TUMBLING (SIZE 1 MINUTE)
GROUP BY RegionId
EMIT CHANGES;
```

#ดูข้อมูล

```
SELECT * FROM PageViews_Tumbling EMIT CHANGES;
```

REGIONID	REGIONNAME	WINDOWSTART	WINDOWEND	VIEWCOUNT	STARTWINDOW	ENDWINDOW
Region_5	Chonburi	1731313080000	1731313140000	12	1731313080000	1731313140000
Region_5	Chonburi	1731313140000	1731313200000	16	1731313140000	1731313200000
Region_5	Chonburi	1731313200000	1731313260000	11	1731313200000	1731313260000
Region_5	Chonburi	1731313260000	1731313320000	10	1731313260000	1731313320000
Region_5	Chonburi	1731313320000	1731313380000	5	1731313320000	1731313380000
Region_5	Chonburi	1731313380000	1731313440000	13	1731313380000	1731313440000
Region_5	Chonburi	1731313440000	1731313500000	12	1731313440000	1731313500000
Region_5	Chonburi	1731313500000	1731313560000	11	1731313500000	1731313560000
Region_5	Chonburi	1731313560000	1731313620000	18	1731313560000	1731313620000

2. Hopping (topic7)

สร้าง Topic7 (Table)

```
CREATE TABLE PageViews_Hopping WITH (
   KAFKA_TOPIC = 'topic_7',
   KEY_FORMAT = 'JSON',
   VALUE_FORMAT = 'JSON'
) AS
SELECT
   RegionId,
   RegionName,
   COUNT(*) AS PageViewCount,
   WINDOWSTART AS StartPeriod,
   WINDOWEND AS EndPeriod
FROM Consolidate_Stream
WINDOW HOPPING (SIZE 5 SECONDS, ADVANCE BY 2 SECONDS)
GROUP BY RegionId, RegionName
EMIT CHANGES;
```

#ดูข้อมูล

```
SELECT * FROM PageViews_Hopping EMIT CHANGES;
```

REGIONID	REGIONNAME	WINDOWSTART	WINDOWEND	PAGEVIEWCOUNT	STARTPERIOD	ENDPERIOD
Region_7	Surat Thani	1731313256000	1731313261000	1	1731313256000	1731313261000
Region_7	Surat Thani	1731313258000	1731313263000	1	1731313258000	1731313263000
Region_7	Surat Thani	1731313260000	1731313265000	1	1731313260000	1731313265000
Region_7	Surat Thani	1731313256000	1731313261000	2	1731313256000	1731313261000
Region_7	Surat Thani	1731313258000	1731313263000	2	1731313258000	1731313263000
Region_7	Surat Thani	1731313260000	1731313265000	2	1731313260000	1731313265000
Region_7	Surat Thani	1731313258000	1731313263000	3	1731313258000	1731313263000
Region_7	Surat Thani	1731313260000	1731313265000	3	1731313260000	1731313265000
Region_7	Surat Thani	1731313258000	1731313263000	4	1731313258000	1731313263000
Region_7	Surat Thani	1731313260000	1731313265000	4	1731313260000	1731313265000
Region_4	Khon Kaen	1731313260000	1731313265000	1	1731313260000	1731313265000

3. Session (topic8)

สร้าง Topic 8 (Table)

```
CREATE TABLE Session_Window_Analysis WITH (
   KAFKA_TOPIC = 'topic_8',
   VALUE_FORMAT = 'JSON',
   PARTITIONS = 5,
   REPLICAS = 3
) AS
SELECT
   RegionId,
   LATEST_BY_OFFSET(Gender) AS Gender,
   COUNT(*) AS PageVisitCount,
   (WINDOWEND - WINDOWSTART) / 1000 AS SessionLengthSeconds,
   WINDOWSTART AS SessionStart,
   WINDOWEND AS SessionEnd
FROM Consolidate_Stream
WINDOW SESSION (5 SECONDS)
GROUP BY RegionId
EMIT CHANGES;
```

#ดูข้อมูล

```
SELECT * FROM Session_Window_Analysis EMIT CHANGES;
```

REGIONID	GENDER	WINDOWSTART	WINDOWEND	PAGEVISITCOUNT	SESSIONLENGTHSECONDS	SESSIONSTART	SESSIONEND
Region_3	OTHER	1731313341033	1731313341033	1	0	1731313341033	1731313341033
Region_9	MALE	1731313342466	1731313342466	1	0	1731313342466	1731313342466
Region_3	OTHER	1731313341033	1731313341033	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>
Region_3	OTHER	1731313341033	1731313342582	2	1	1731313341033	1731313342582
Region_9	MALE	1731313342466	1731313342466	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>
Region_9	MALE	1731313342466	1731313342632	2	0	1731313342466	1731313342632
Region_7	OTHER	1731313345253	1731313345253	1	0	1731313345253	1731313345253
Region_3	OTHER	1731313341033	1731313342582	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>
Region_3	OTHER	1731313341033	1731313347523	3	6	1731313341033	1731313347523
Region_7	OTHER	1731313345253	1731313345253	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>	<TOMBSTONE>
Region_7	OTHER	1731313345253	1731313348222	2	2	1731313345253	1731313348222

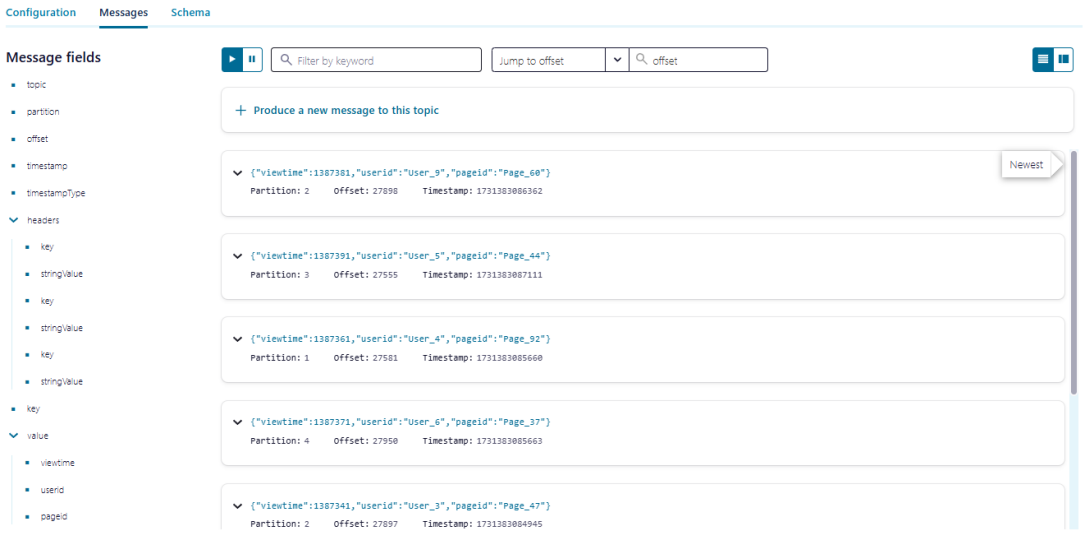
#Topic10 (เพิ่มเติม)

```
CREATE TABLE Session_Window_Analysis WITH (
   KAFKA_TOPIC = 'topic_10',
   VALUE_FORMAT = 'JSON',
   PARTITIONS = 5,
   REPLICAS = 3
) AS
SELECT
   RegionId,
   LATEST_BY_OFFSET(Gender) AS Gender,
   COUNT(*) AS PageVisitCount,
   (WINDOWEND - WINDOWSTART) / 1000 AS SessionLengthSeconds,
   WINDOWSTART AS SessionStart,
   WINDOWEND AS SessionEnd
FROM Consolidate_Stream
WINDOW SESSION (5 SECONDS)
GROUP BY RegionId
EMIT CHANGES;
```

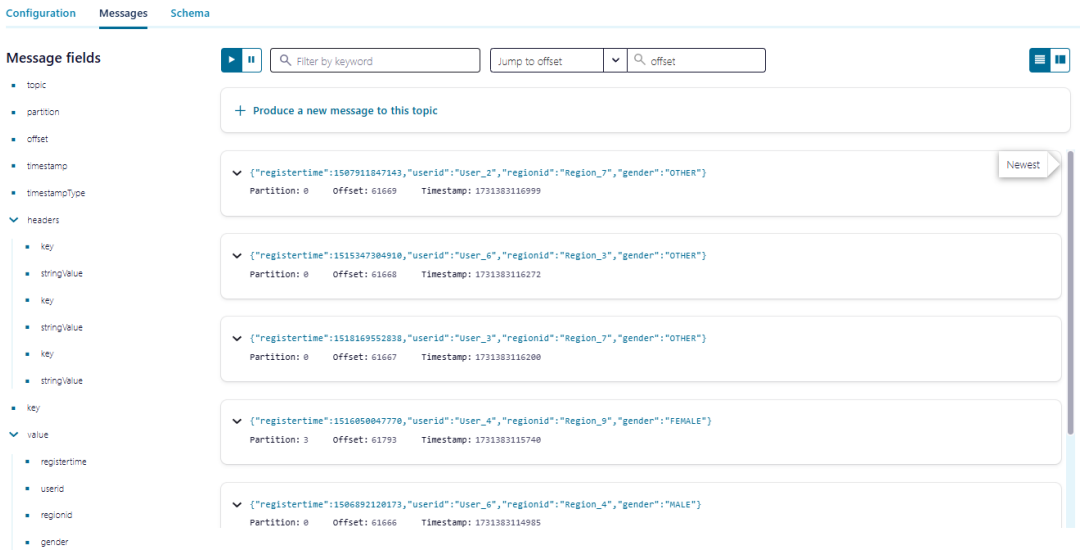
#Topic11(เพิ่มเติม)

```
CREATE TABLE Region_Page_Summary WITH (
  KAFKA_TOPIC = 'topic_11',
  VALUE_FORMAT = 'JSON',
  PARTITIONS = 5,
  REPLICAS = 3
) AS
SELECT
  RegionId,
  LATEST_BY_OFFSET(RegionName) AS RegionName,
  COUNT(*) AS TotalPageVisits,
  ROUND(AVG(ViewTime), 2) AS AverageViewTime
FROM Consolidate_Stream
GROUP BY RegionId
EMIT CHANGES;
```

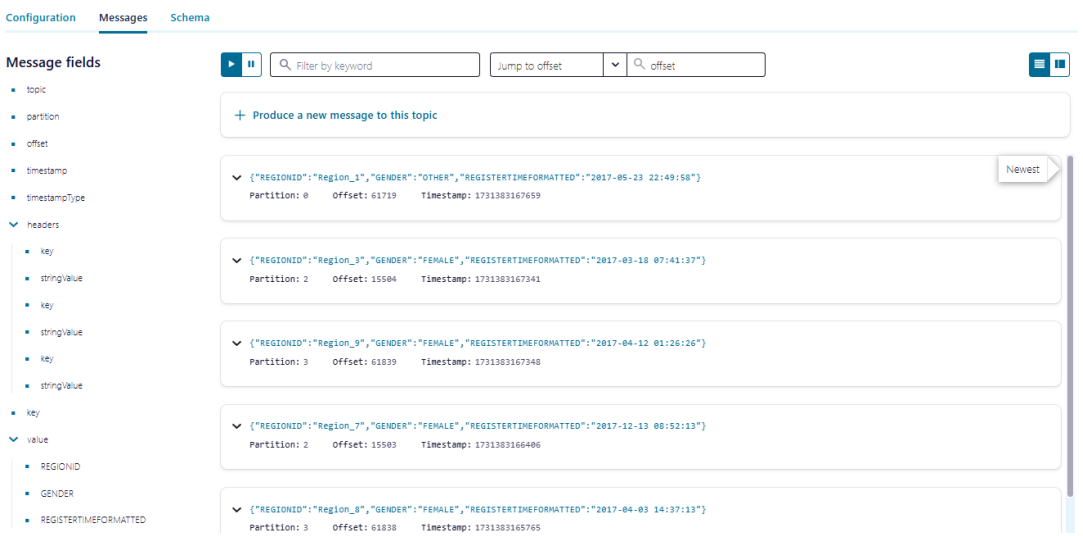
topic_1



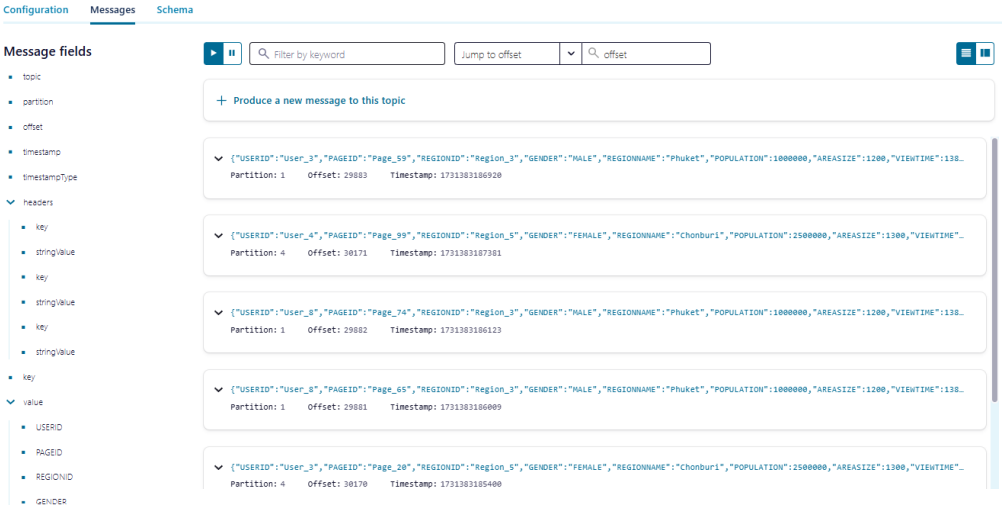
topic_2



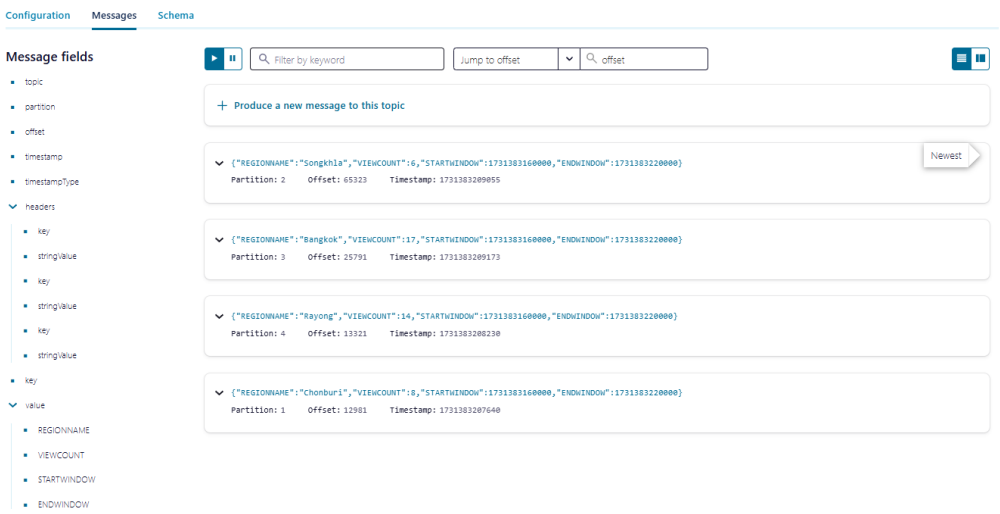
topic_4



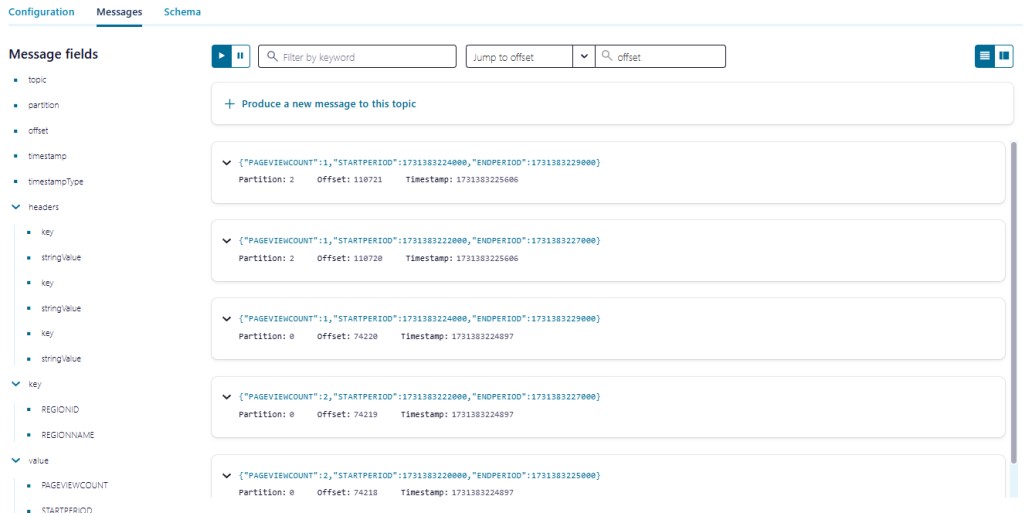
topic_5



topic_6



topic_7



d. Apache Pinot [5,6]

สร้าง Schema และ Table สำหรับ Consolidate_Stream (ข้อมูลจาก topic5)

#สร้าง Schema

```
SCHEMA_JSON='{
  "schemaName": "Consolidate",
  "enableColumnBasedNullHandling": false,
  "dimensionFieldSpecs": [
    { "name": "USERID", "dataType": "STRING", "notNull": false },
    { "name": "REGIONID", "dataType": "STRING", "notNull": false },
    { "name": "GENDER", "dataType": "STRING", "notNull": false },
    { "name": "REGIONNAME", "dataType": "STRING", "notNull": false },
    { "name": "AREASIZE", "dataType": "INT", "notNull": false }
  ],
  "metricFieldSpecs": [
    { "name": "VIEWPERCENTAGE", "dataType": "FLOAT", "notNull": false }
  ],
  "dateTimeFieldSpecs": [
    { "name": "VIEWTIME", "dataType": "LONG", "notNull": false, "format": "1:MILLISECONDS:EPOCH", "granularity":
"1:MILLISECONDS" }
  ]
}'
curl -X POST -H "Content-Type: application/json" -d "$SCHEMA_JSON" http://localhost:9000/schemas
```

#สร้าง table

```
TABLE_JSON='{
  "tableName": "Consolidate_REALTIME",
  "tableType": "REALTIME",
  "segmentsConfig": {
    "schemaName": "Consolidate",
    "replication": "1",
    "timeColumnName": "VIEWTIME",
    "replicasPerPartition": "1"
  },
  "tenants": {
    "broker": "DefaultTenant",
    "server": "DefaultTenant"
  },
  "tableIndexConfig": {
    "streamConfigs": {
      "streamType": "kafka",
      "stream.kafka.topic.name": "topic_5",
      "stream.kafka.broker.list": "broker1:29092",
      "stream.kafka.consumer.type": "lowlevel",
      "stream.kafka.consumer.prop.auto.offset.reset": "smallest",
      "stream.kafka.consumer.factory.class.name":
"org.apache.pinot.plugin.stream.kafka20.KafkaConsumerFactory",
      "stream.kafka.decoder.class.name": "org.apache.pinot.plugin.stream.kafka.KafkaJSONMessageDecoder"
    }
  },
```

```
  "metadata": {}
}'
curl -X POST -H "Content-Type: application/json" -d "$TABLE_JSON" http://localhost:9000/tables
```

สร้าง Schema และ Table สำหรับ PageViews_Tumbling (ข้อมูลจาก topic_6)

 # สร้าง Schema

```
SCHEMA_JSON='{
  "schemaName": "PageViews_Tumbling",
  "enableColumnBasedNullHandling": false,
  "dimensionFieldSpecs": [
    { "name": "REGIONNAME", "dataType": "STRING", "notNull": false }
  ],
  "metricFieldSpecs": [
    { "name": "VIEWCOUNT", "dataType": "INT", "notNull": false }
  ],
  "dateTimeFieldSpecs": [
    { "name": "STARTWINDOW", "dataType": "LONG", "notNull": false, "format": "1:MILLISECONDS:EPOCH", "granularity": "1:MILLISECONDS" },
    { "name": "ENDWINDOW", "dataType": "LONG", "notNull": false, "format": "1:MILLISECONDS:EPOCH", "granularity": "1:MILLISECONDS" }
  ]
}'
curl -X POST -H "Content-Type: application/json" -d "$SCHEMA_JSON" http://localhost:9000/schemas
```

 # สร้าง Table

```
TABLE_JSON='{
  "tableName": "PageViews_Tumbling_REALTIME",
  "tableType": "REALTIME",
  "segmentsConfig": {
    "schemaName": "PageViews_Tumbling",
    "replication": "1",
    "timeColumnName": "STARTWINDOW",
    "replicasPerPartition": "1"
  },
  "tenants": {
    "broker": "DefaultTenant",
    "server": "DefaultTenant"
  },
  "tableIndexConfig": {
    "streamConfigs": {
      "streamType": "kafka",
      "stream.kafka.topic.name": "topic_6",
      "stream.kafka.broker.list": "broker1:29092",
      "stream.kafka.consumer.type": "lowlevel",
      "stream.kafka.consumer.prop.auto.offset.reset": "smallest",
      "stream.kafka.consumer.factory.class.name":
"org.apache.pinot.plugin.stream.kafka20.KafkaConsumerFactory",
      "stream.kafka.decoder.class.name": "org.apache.pinot.plugin.stream.kafka.KafkaJSONMessageDecoder"
    }
```

```
  },
  "metadata": {}
}'
curl -X POST -H "Content-Type: application/json" -d "$TABLE_JSON" http://localhost:9000/tables
```

สร้าง Schema และ Table สำหรับ Session_Window_Analysis (ข้อมูลจาก topic8)

#สร้าง Schema

```
SCHEMA_JSON='{
  "schemaName": "Session_Window_Analysis",
  "enableColumnBasedNullHandling": false,
  "dimensionFieldSpecs": [
    { "name": "GENDER", "dataType": "STRING", "notNull": false },
    { "name": "PAGEVISITCOUNT", "dataType": "INT", "notNull": false }
  ],
  "metricFieldSpecs": [
    { "name": "SESSIONLENGTHSECONDS", "dataType": "LONG", "notNull": false }
  ],
  "dateTimeFieldSpecs": [
    { "name": "SESSIONSTART", "dataType": "LONG", "notNull": false, "format": "1:MILLISECONDS:EPOCH",
"granularity": "1:MILLISECONDS" }
  ]
}'
curl -X POST -H "Content-Type: application/json" -d "$SCHEMA_JSON" http://localhost:9000/schemas
```

#สร้าง Table

```
TABLE_JSON='{
  "tableName": "Session_Window_Analysis_REALTIME",
  "tableType": "REALTIME",
  "segmentsConfig": {
    "schemaName": "Session_Window_Analysis",
    "replication": "1",
    "timeColumnName": "SESSIONSTART",
    "replicasPerPartition": "1"
  },
  "tenants": {
    "broker": "DefaultTenant",
    "server": "DefaultTenant"
  },
  "tableIndexConfig": {
    "streamConfigs": {
      "streamType": "kafka",
      "stream.kafka.topic.name": "topic_8",
      "stream.kafka.broker.list": "broker1:29092",
      "stream.kafka.consumer.type": "lowlevel",
      "stream.kafka.consumer.prop.auto.offset.reset": "smallest",
      "stream.kafka.consumer.factory.class.name":
"org.apache.pinot.plugin.stream.kafka20.KafkaConsumerFactory",
      "stream.kafka.decoder.class.name": "org.apache.pinot.plugin.stream.kafka.KafkaJSONMessageDecoder"
    }
  },
```

```
  "metadata": {}
}'
curl -X POST -H "Content-Type: application/json" -d "$TABLE_JSON" http://localhost:9000/tables
```

สร้าง Schema และ Table สำหรับ Region_Page_Summary (ข้อมูลจาก topic11)

#สร้าง Schema

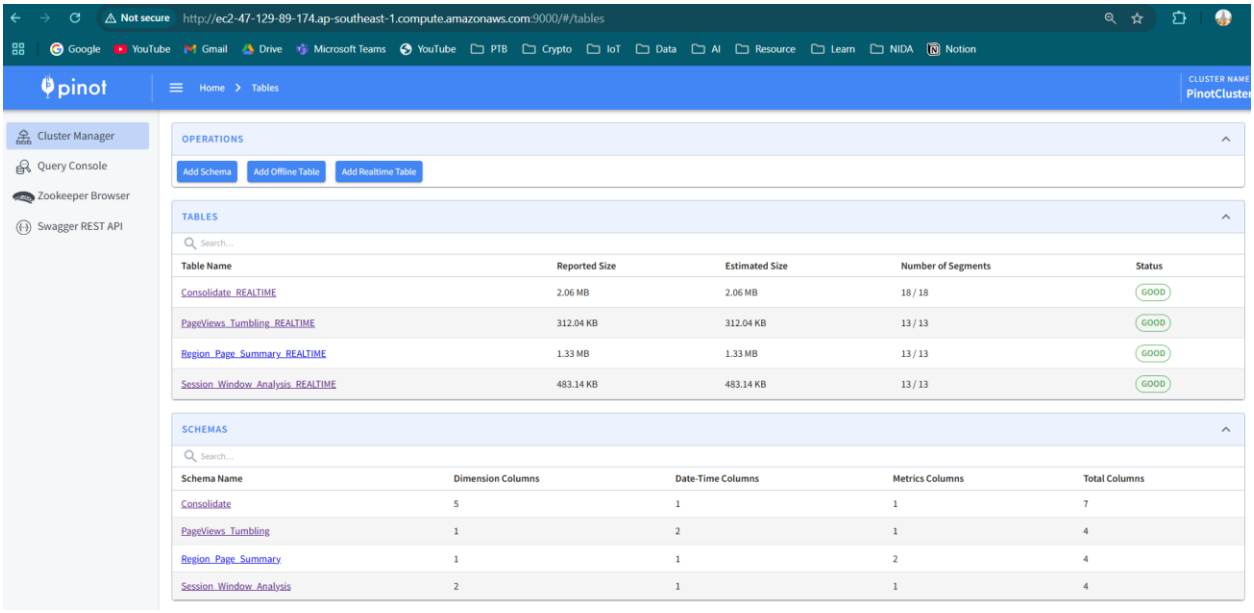
```
SCHEMA_JSON='{
  "schemaName": "Region_Page_Summary",
  "enableColumnBasedNullHandling": false,
  "dimensionFieldSpecs": [
    { "name": "REGIONNAME", "dataType": "STRING", "notNull": false }
  ],
  "metricFieldSpecs": [
    { "name": "TOTALPAGEVISITS", "dataType": "INT", "notNull": false },
    { "name": "AVERAGEVIEWTIME", "dataType": "DOUBLE", "notNull": false }
  ],
  "dateTimeFieldSpecs": [
    { "name": "TIMESTAMP", "dataType": "LONG", "notNull": false, "format": "1:MILLISECONDS:EPOCH", "granularity": "1:MILLISECONDS" }
  ]
}'
curl -X POST -H "Content-Type: application/json" -d "$SCHEMA_JSON" http://localhost:9000/schemas
```

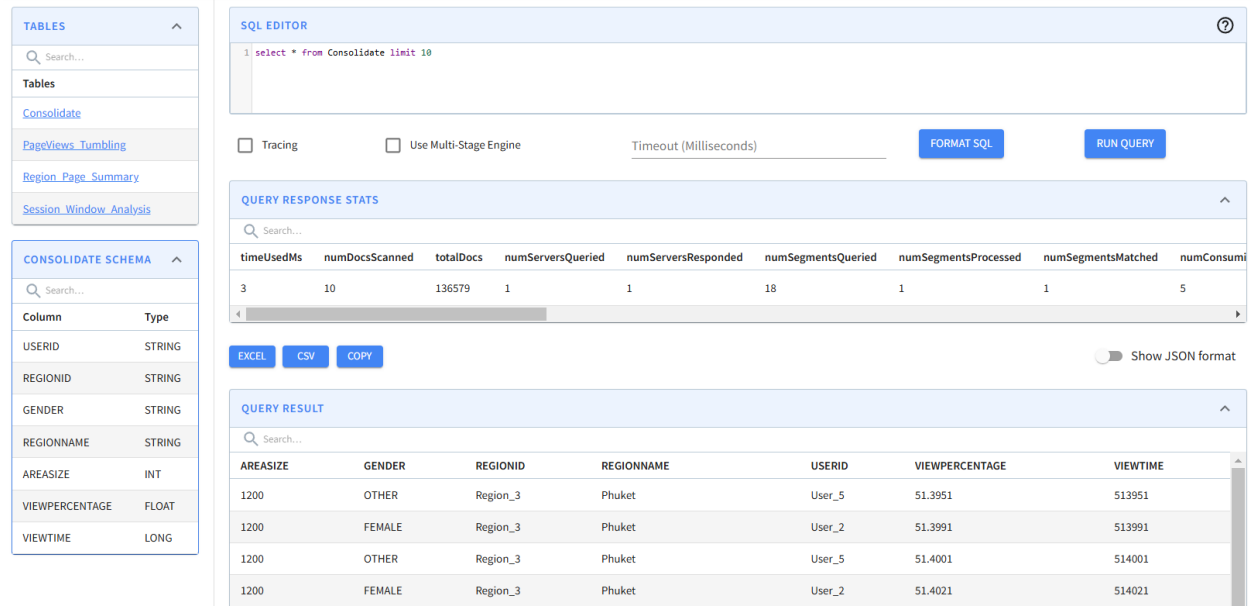
#สร้าง Table

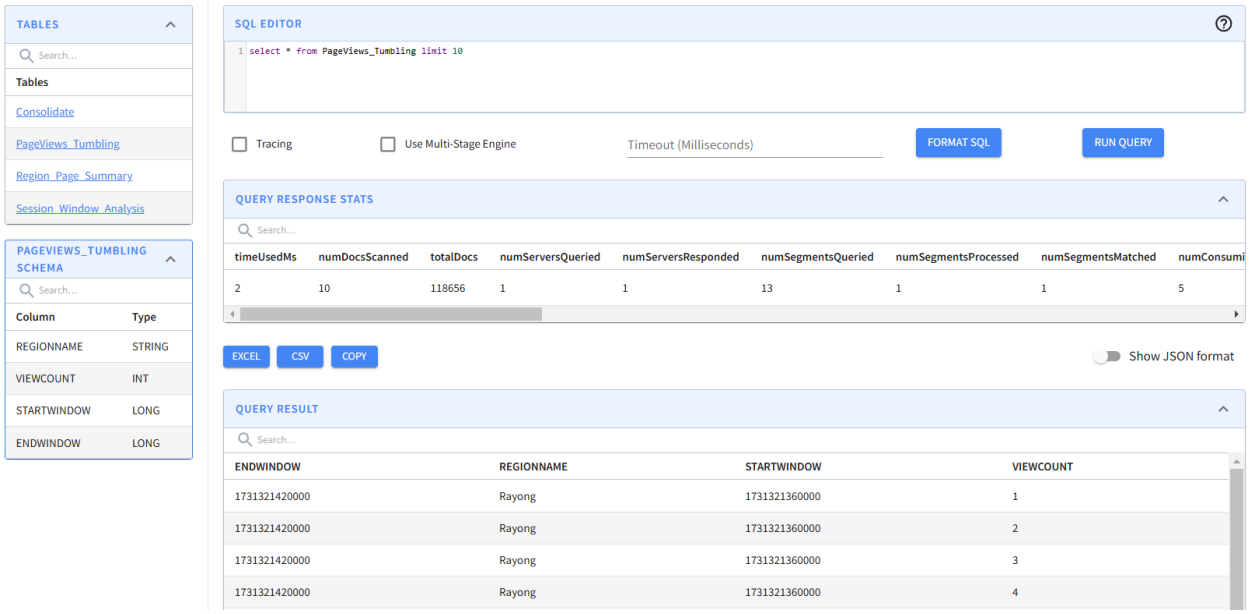
```
TABLE_JSON='{
  "tableName": "Region_Page_Summary_REALTIME",
  "tableType": "REALTIME",
  "segmentsConfig": {
    "schemaName": "Region_Page_Summary",
    "replication": "1",
    "timeColumnName": "TIMESTAMP",
    "replicasPerPartition": "1"
  },
  "tenants": {
    "broker": "DefaultTenant",
    "server": "DefaultTenant"
  },
  "tableIndexConfig": {
    "streamConfigs": {
      "streamType": "kafka",
      "stream.kafka.topic.name": "topic_11",
      "stream.kafka.broker.list": "broker1:29092",
      "stream.kafka.consumer.type": "lowlevel",
      "stream.kafka.consumer.prop.auto.offset.reset": "smallest",
      "stream.kafka.consumer.factory.class.name":
"org.apache.pinot.plugin.stream.kafka20.KafkaConsumerFactory",
      "stream.kafka.decoder.class.name": "org.apache.pinot.plugin.stream.kafka.KafkaJSONMessageDecoder"
    }
  },
```

```
  "metadata": {}
}'
```

curl -X POST -H "Content-Type: application/json" -d "$TABLE_JSON" http://localhost:9000/tables







TABLES

Search...

Tables

Consolidate

PageViews_Tumbling

Region_Page_Summary

Session_Window_Analysis

PAGEVIEWS_TUMBLING SCHEMA

Search...

Column	Type
REGIONNAME	STRING
VIEWCOUNT	INT
STARTWINDOW	LONG
ENDWINDOW	LONG

SQL EDITOR

```
1 select * from PageViews_Tumbling limit 10
```

Tracing Use Multi-Stage Engine Timeout (Milliseconds) FORMAT SQL RUN QUERY

QUERY RESPONSE STATS

Search...

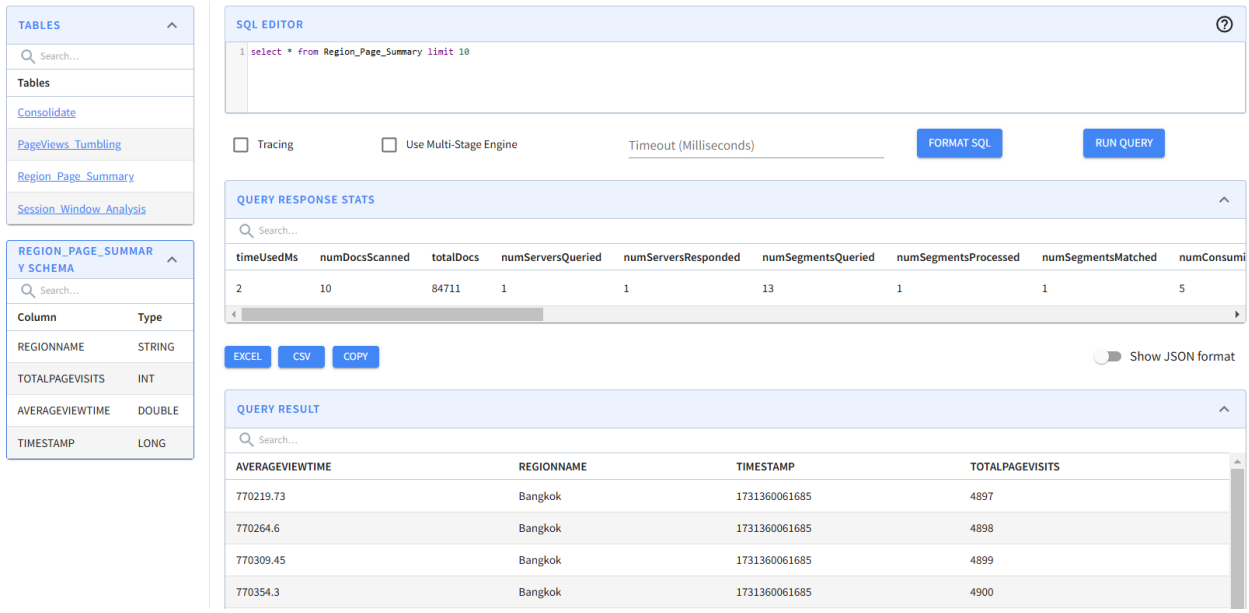
timeUsedMs	numDocsScanned	totalDocs	numServersQueried	numServersResponded	numSegmentsQueried	numSegmentsProcessed	numSegmentsMatched	numConsumi
2	10	118656	1	1	13	1	1	5

EXCEL CSV COPY Show JSON format

QUERY RESULT

Search...

ENDWINDOW	REGIONNAME	STARTWINDOW	VIEWCOUNT
1731321420000	Rayong	1731321360000	1
1731321420000	Rayong	1731321360000	2
1731321420000	Rayong	1731321360000	3
1731321420000	Rayong	1731321360000	4



TABLES

Search...

Tables

Consolidate

PageViews_Tumbling

Region_Page_Summary

Session_Window_Analysis

REGION_PAGE_SUMMARY SCHEMA

Search...

Column	Type
REGIONNAME	STRING
TOTALPAGEVISITS	INT
AVERAGEVIEWTIME	DOUBLE
TIMESTAMP	LONG

SQL EDITOR

```
1 select * from Region_Page_Summary limit 10
```

Tracing Use Multi-Stage Engine Timeout (Milliseconds) FORMAT SQL RUN QUERY

QUERY RESPONSE STATS

Search...

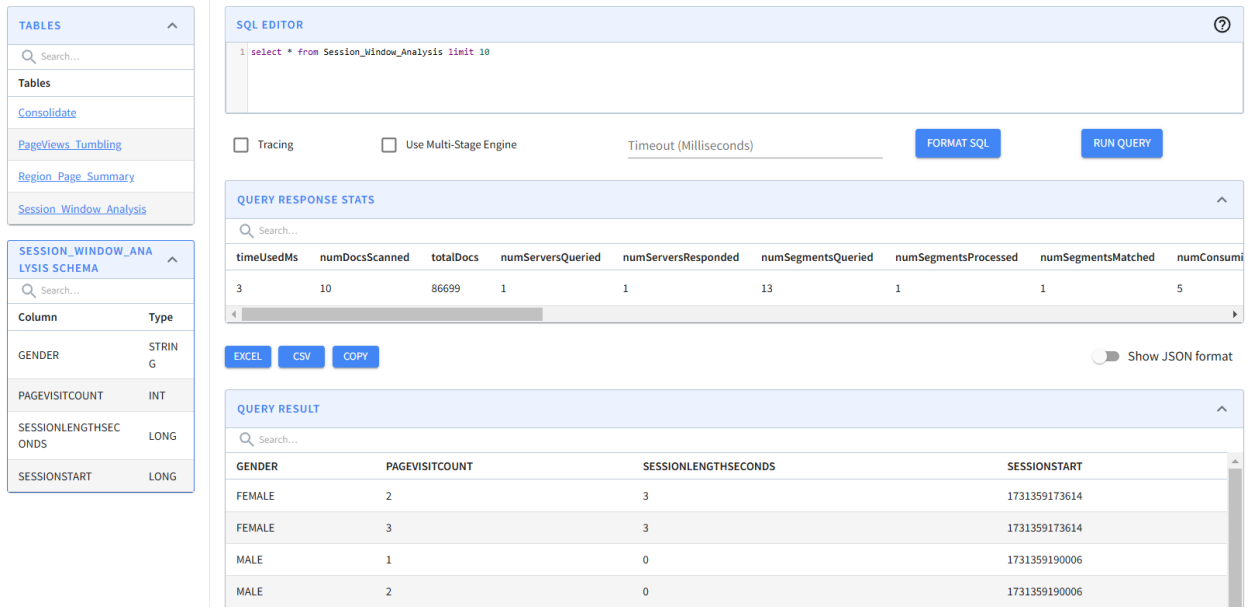
timeUsedMs	numDocsScanned	totalDocs	numServersQueried	numServersResponded	numSegmentsQueried	numSegmentsProcessed	numSegmentsMatched	numConsumi
2	10	84711	1	1	13	1	1	5

EXCEL CSV COPY Show JSON format

QUERY RESULT

Search...

AVERAGEVIEWTIME	REGIONNAME	TIMESTAMP	TOTALPAGEVISITS
770219.73	Bangkok	1731360061685	4897
770264.6	Bangkok	1731360061685	4898
770309.45	Bangkok	1731360061685	4899
770354.3	Bangkok	1731360061685	4900



TABLES

Search...

Tables

Consolidate

PageViews_Tumbling

Region_Page_Summary

Session_Window_Analysis

SESSION_WINDOW_ANALYSIS SCHEMA

Search...

Column	Type
GENDER	STRING
PAGEVISITCOUNT	INT
SESSIONLENGTHSECONDS	LONG
SESSIONSTART	LONG

SQL EDITOR

```
1 select * from Session_Window_Analysis limit 10
```

Tracing Use Multi-Stage Engine Timeout (Milliseconds) FORMAT SQL RUN QUERY

QUERY RESPONSE STATS

Search...

timeUsedMs	numDocsScanned	totalDocs	numServersQueried	numServersResponded	numSegmentsQueried	numSegmentsProcessed	numSegmentsMatched	numConsumi
3	10	86699	1	1	13	1	1	5

EXCEL CSV COPY Show JSON format

QUERY RESULT

Search...

GENDER	PAGEVISITCOUNT	SESSIONLENGTHSECONDS	SESSIONSTART
FEMALE	2	3	1731359173614
FEMALE	3	3	1731359173614
MALE	1	0	1731359190006
MALE	2	0	1731359190006

e. Dashboard [7]

Dashboard ใช้ streamlit

pip install streamlit pandas pinotdb plotly

~/.local/bin/streamlit run dashboard.py --server.port 8501 --server.enableCORS false

export PATH=$PATH:~/.local/bin

streamlit --version

nohup streamlit run dashboard.py --server.port 8501 --server.enableCORS false > streamlit.log 2>&1 &

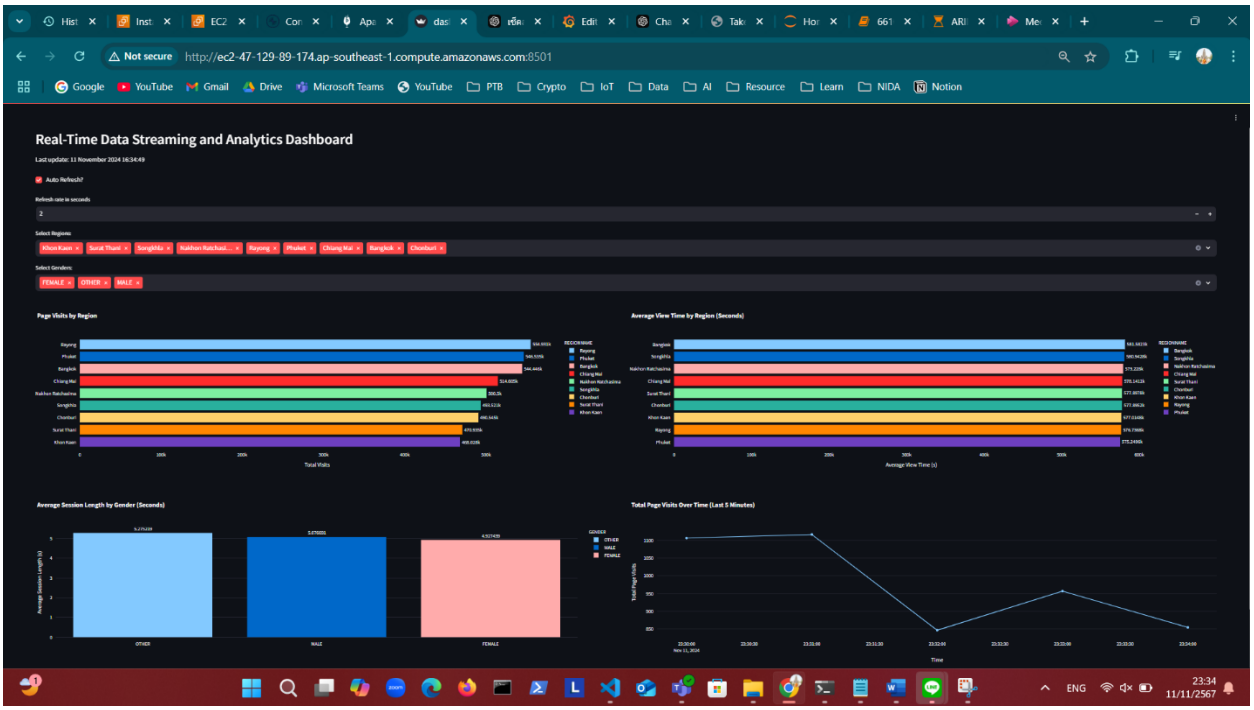
Run Streamlit ที่เขียนใน python ไฟล์ (คำสั่งโยนไฟล์อยู่ด้านบนรวมกับโยนไฟล์อื่นๆครับ File : dashboard.py)

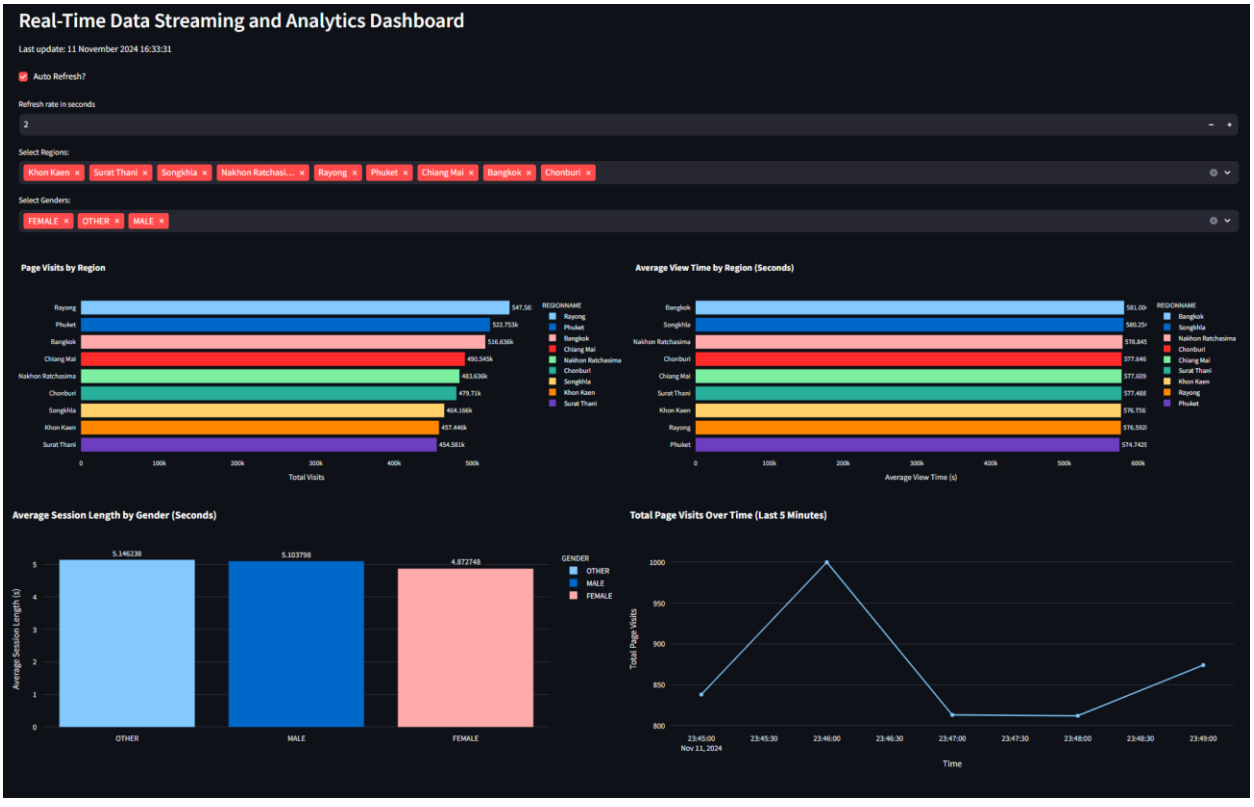
python3 -m streamlit run dashboard.py --server.port 8501 --server.enableCORS false

[ดู Dashboard ผ่าน URL นี้](http://ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:8501)

http://ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:8501

ตัวอย่างหน้าตา Dashboard





Port ทั้งหมด

- Confluent

http://ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:9021

- Apache Pinot

http://ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:9000

- Streamlit real-time dashboard

http://ec2-47-129-89-174.ap-southeast-1.compute.amazonaws.com:8501

#ลบ Datagen (Topic1,2)

curl -X DELETE http://localhost:8083/connectors/datagen-pageviews

curl -X DELETE http://localhost:8083/connectors/datagen-users