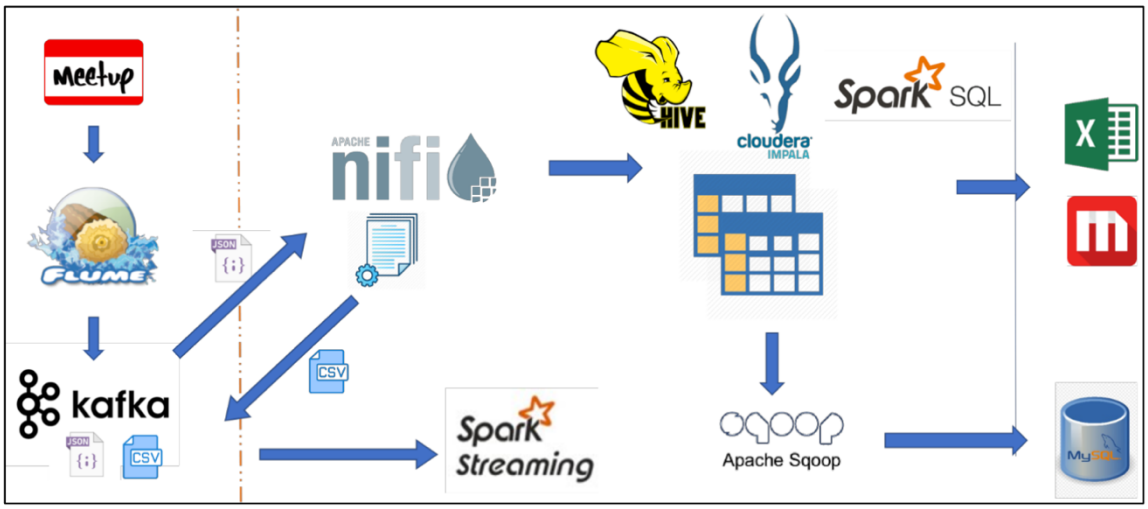
# Assignment 3

## Meetup stream data ingestion and analysis case study




December 7, 2021

**Team #5**
Marie-Noël Lepage et al.

In our team, all of us were working and participating. We all worked on code on our own and shared our progress with each other to solve issues and obtain a final working code. In parallel we produced and review the documentation to supplement the exercise.

# PLATFORM PLANNING

**To determine the memory size of the NameNode server, we need to add the memory needed by NameNode to manage the HDFS cluster metadata (in memory) and the memory needed for the Operating System.**
**The IT department defined the hardware configuration for each node in the cluster:**

- **1CPU8VCores,16GBmemory,20TB Hard Disk Drivespace. (to be shared between the OS and applications)**
- **The datanode storage capacity was calculated as 13TB**

- **<u>Assuming a replication factor of 3 how many nodes in the cluster we need to store all the data?</u>**

*We used the formula from the following article to plan the HDFS cluster size:*

*https://www.edureka.co/blog/hadoop-cluster-capacity-planning/#factor*

**Hadoop Storage (HS) = CRS / (1-i)**

Where

- **C**= Compression Ratio
- **R**= Replication Factor
- **S**= Size of the data to be moved into Hadoop
- **i**= Intermediate Factor

1. **Determine the size of the data from project requirements.**

   *Big Company Group collect data in real time from social networks platforms. For a particular experience, there are 50 Flume agents (one per machine) in a data center that collect data in real time. There are multiple sources each with their own data elements, but they follow a common data format.*

   - *The data is in JSON format*
   - *Each Flume agent transmits on average 50 attribute records every second.*
   - *Each record's data attribute row is on average 1600 bytes wide.*
   - *Corporate data standards require all input data to be persisted for 12 months.*

- *All QA test results data (estimated to 10% of the ingested data per year) to be stored for a period of 5 years.*

Based on this information, the amount of data we need to store for one year is equal to:

50 Flume agents * 50 attribute records/sec * 1600 bytes/attribute * 60 sec * 60 min * 24h * 365d = 126 144 000 000 000 bytes = **114.727 TB**

In addition, we need to store 10% of this data from the past 5 years. So the total size of the data stored is :

114.727*1 TB * (1 + 5*0.1) = **172.1 TB**

2. **Determine the number of nodes required to store all the data.**

Assuming a replication factor of 3, intermediate factor of 25% and compression ratio of 1 (no compression) we need a storage capacity of:

HS = CRS/(1-i) = 1*3*172.1 TB / (1-0.25) = **688.4 TB**

Based on the 13TB storage capacity of the nodes, we will require:

688.4 TB / 13 TB/node = **53 nodes**

- **Assuming a HDFS bloc size is 128 MB and each block need 680 Bytes for its metadata. What would be the recommended NameNode memory size if the memory allocated to the OS is 8 GB? (should be round up multiple of 2)**

1. **Determine the number of blocks required for storing 172.1 TB (before replication factor):**

172.1 TB = 180 459 929.6 MB

180 459 929.6 MB / 128 MB/block = 1 409 843.2 blocks

2. Determine metadata memory size based on the number of blocks:

1 409 843.2 blocks * 680 bytes/block = 958 693 376 bytes = 0.89 GB

3. Determine NameNode memory size required including OS:

8 GB + 0.89 GB = 8.89Gb ≈ **10 GB**

# PLATFORM PREPARATION

## TASK 01
## Prepare Kafka topics

**The topics characteristics should meet your hardware limitation**

a. **Create a Kafka topic meetup-data that will be used to store events collected by the Flume agent. (Sink 1)**
b. **Create a Kafka topic meetup-agg that will be used to store events collected by the Flume agent. (Sink 2)**

Open a Terminal and navigate to the kafka directory:
$ cd /usr/lib/kafka

Run the kafka server using the following command:
$ sudo bin/kafka-server-start.sh config/server.properties

Open a new terminal window and navigate to the kafka directory
$ cd /usr/lib/kafka

Create a new topic called meetup-data, having a single partition and a replication factor set to of 1
> bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic meetup-data

Create a new topic called meetup-agg, having a single partition and a replication factor set to of 1
> bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic meetup-agg

# MEETUP REAL-TIME DATA INGESTION

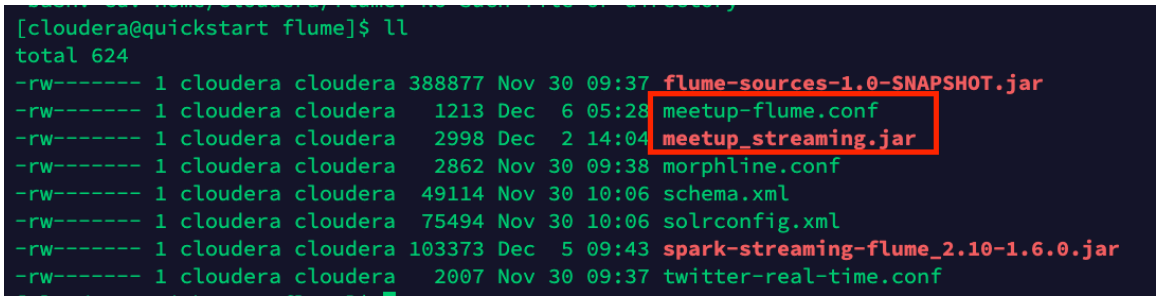
## TASK 02
## Create Flume agent

**Part A:**
Create a flume agent configuration file.

```
# Naming the components on the current agent.
MeetupAgent.sources = kafka-source
MeetupAgent.channels = memory-channel
MeetupAgent.sinks = kafka-sink

# Describing/Configuring the source
MeetupAgent.sources.kafka-source.type = StreamingAPISource
MeetupAgent.sources.kafka-source.zookeeperConnect = localhost:2181
MeetupAgent.sources.kafka-source.url = https://stream.meetup.com/2/rsvps
MeetupAgent.sources.kafka-source.batch.size = 5
MeetupAgent.sources.kafka-source.channels = memory-channel
MeetupAgent.sources.kafka-source.interceptors = ts
MeetupAgent.sources.kafka-source.interceptors.ts.type = timestamp

# Describing/Configuring the channel
MeetupAgent.channels.memory-channel.type = memory
MeetupAgent.channels.memory-channel.capacity = 1000000
MeetupAgent.channels.memory-channel.transactionCapacity = 500000

# Describing/Configuring kafka sink
MeetupAgent.sinks.kafka-sink.type = org.apache.flume.sink.kafka.KafkaSink
MeetupAgent.sinks.kafka-sink.kafka.bootstrap.servers = localhost:9092
MeetupAgent.sinks.kafka-sink.kafka.topic= meetup-data

# Binding the sources And Sink to the channel
MeetupAgent.sources.kafka-source.channels= memory-channel
MeetupAgent.sinks.kafka-sink.channel= memory-channel
```

**Part B:**
Write the command line to run the Flume agent.

We put the Flume agent meetup-flume.conf and the meetup_streaming.jar in the directory /home/cloudera/flume.

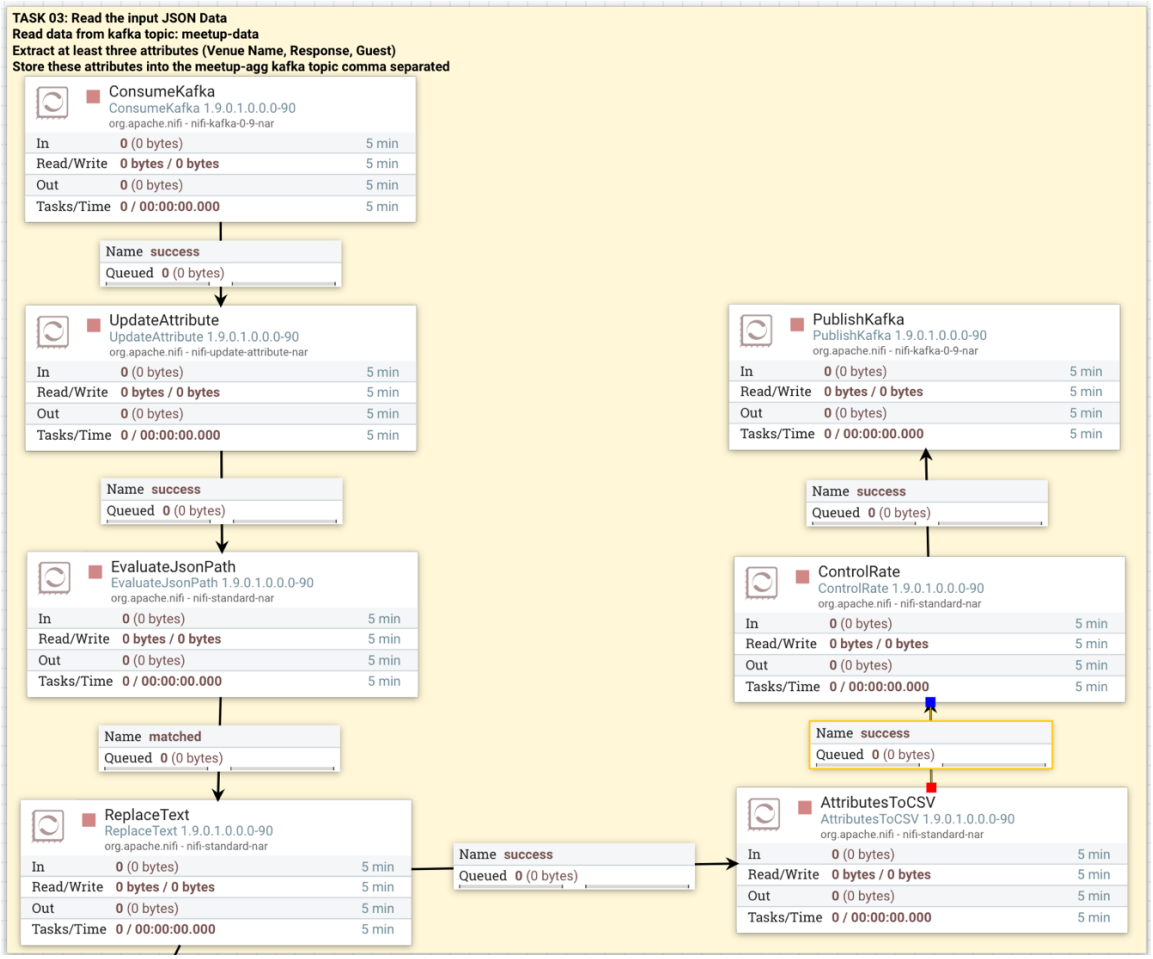



We run after the flume Agent in the flume directory:

flume-ng agent –n MeetupAgent –c conf –f meetup-flume.conf –C meetup_streaming.jar

# PREAPARING MEETUP DATA

## Task 03
## Read the input JSON Data (Nifi)




- **Read Data from Kafka topic: meetup-data**

| Processor | Properties | |
|---|---|---|
| **ConsumeKafka** | Kafka Brokers | localhost:9092 |
| | Security Protocol | PLAINTEXT |
| | Topic Name(s) | Meetup-data |
| | Group ID | flume |
| | Offset Reset | latest |
| | Key Attribute Encoding | UTF-8 Encoded |

Link on success to:

| Processor | Properties | |
|---|---|---|
| **UpdateAttribute** | Store State | Do not store state |
| | Cache Value Lookup Cache Size | 100 |
| | Add these custom attributes | |
| | filename | ${UUID()} |
| | mime.type | application/json |
| | schema.name | meetup |

- **Extract at least three attributes (Venue Name, Response, Guest)**

Link on success to:

| Processor | Properties | |
|---|---|---|
| **EvaluateJsonPath**<br><br>Relationships:<br>failure<br>unmatched | Destination | flowfile-attribute |
| | Return Type | auto-detect |
| | Path Not Found Behavior | ignore |
| | Null Value Representation | Empty string |
| | Add these custom attributes | |
| | venue_name | $.venue.venue_name |
| | guest | $.guests |
| | response | $.response |

Link on matched to:

| Processor | Properties | |
|---|---|---|
| **ReplaceText**<br><br>Relationships:<br>failure | Search Value | , →(, ) |
| | Replacement Value | ; |
| | Character Set | UTF-8 |
| | Maximum Buffer Size | 1 MB |
| | Replacement Strategy | Literal Replace |
| | Evaluation Mode | Entire text |

We use the processor Replace text for delete the comma (, ) in the attribute venue_name. The rule is not perfect and with more time will be ameliorated. The comma generated in the text could be give some problems when we create after table with delimited ‘,’.

- **Store these attributes into the meetup-agg Kafka topic comma separated**

Link on success to:

| Processor | Properties | |
|---|---|---|
| **AttributesToCSV**<br><br>Relationships:<br>failure | Attribute List | venue_name,guest,response |
| | Destination | flowfile-content |
| | Include Core Attributes | false |
| | Null Value | true |
| | Include Schema | false |

Link on success to:

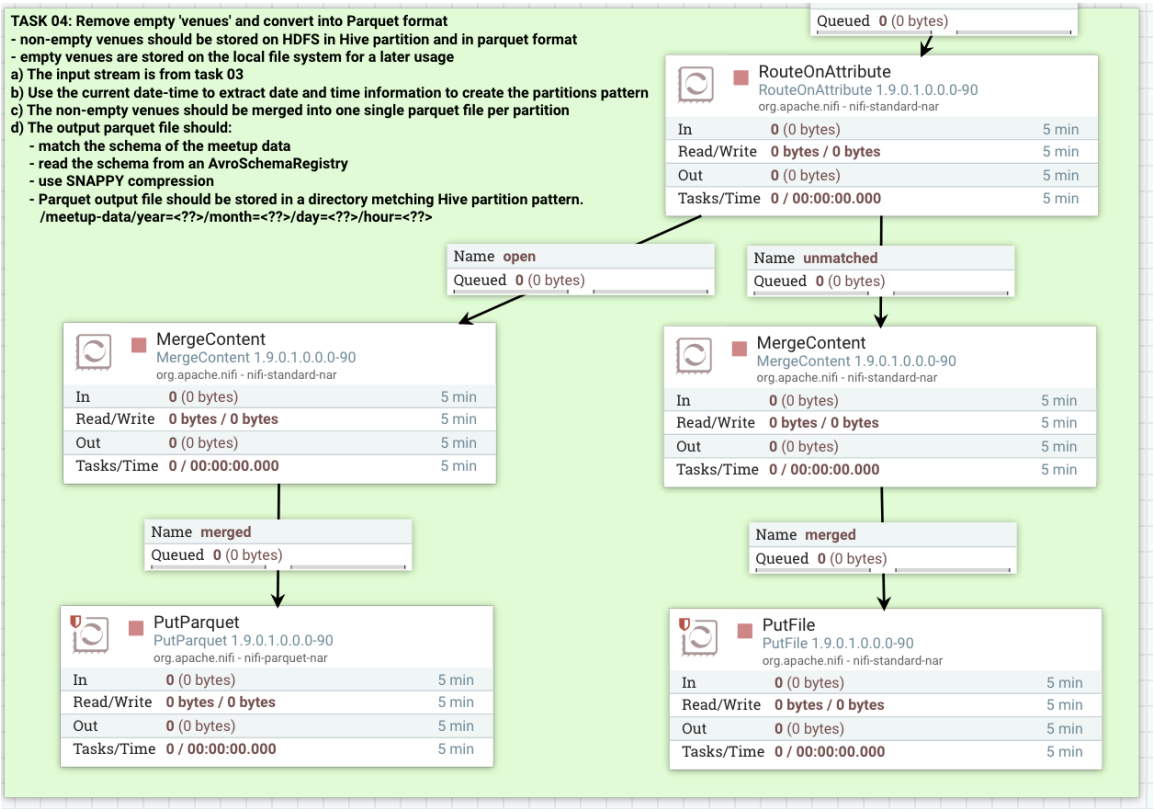
| Processor | Properties | |
|---|---|---|
| **ControlRate** | Rate Control Criteria | flowfile count |
| | Maximum Rate | 20 |
| | Time Duration | 1 min |

Link on success to:

| Processor | Properties | |
|---|---|---|
| **PublishKafka** | Kafka Brokers | localhost:9092 |
| | Security Protocol | PLAINTEXT |
| | Topic Name | meetup-agg |
| | Delivery Encoding | Best Effort |
| | Key Attribute Encoding | UTF-8 Encoded |
| | Max Request Size | 1 MB |
| | Acknowledgment Wait Time | 5 secs |
| | Max Metadata Wait Time | 5 sec |
| | Compression Type | none |

# STORING MEETUP DATA

## Task 04
## Remove empty ‘venues’ and convert into Parquet format




Link ReplaceText (task 03) on success to:

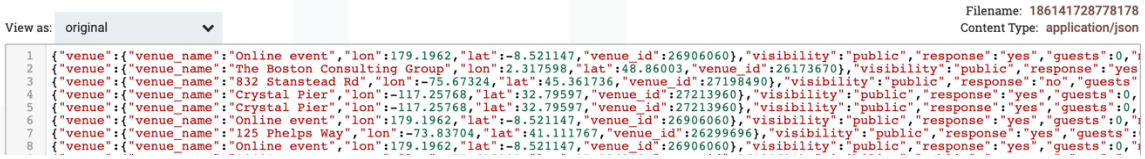
| Processor | Properties | |
|---|---|---|
| **RouteOnAttribute** | Routing Strategy | Route to Property name |
| | Add this custom attribute | |
| | open | ${venue_name:isEmpty():not()} |

- **Non-empty venues should be stored on HDFS in Hive partition and in parquet format.**

Link on open to:

| Processor | Properties | |
|---|---|---|
| **MergeContent**<br><br>Relationships:<br>failure<br>original<br>merged | Merge Strategy | Bin-Packing Algorithm |
| | Merge Format | Binary Concatenation |
| | Attribute Strategy | Keep Only Common Attributes |
| | Metadata Strategy | Do Not Merge Uncommon Metadata |
| | Minimum Number of Entries | 1 |
| | Maximum Number of Entries | 1000000 |
| | Minimum Group Size | 0 B |
| | Maximum number of Bins | 100 |
| | Delimiter Strategy | Text |
| | Demarcator | Shift + enter |
| | Compression Level | 1 |
| | Keep Path | false |
| | **Properties** | |
| | Run Schedule | 3600 sec |

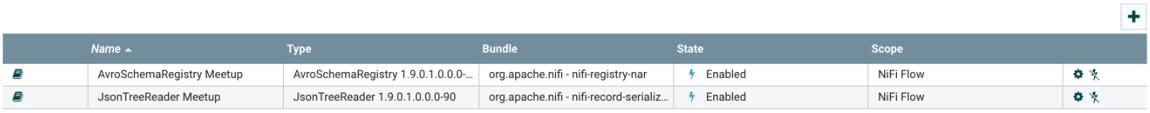
The data is merged in one file. We see the venue is not empty:




Link on merged to:

| Processor | Properties | |
|---|---|---|
| **PutParquet**<br><br>Relationships:<br>failure<br>retry<br>success | Hadoop Configuration Resources | /etc/hadoop/conf/core-site.xml,/etc/hadoop/conf/hdfs-site.xml |
| | Record Reader | JsonTreeReader Meetup |
| | Directory | /meetup-data/year=${now():format("yyyy", "GMT")}/month=${now():format("MM", "GMT")}/day=${now():format("dd", "GMT")}/hour=${now():format("HH", "GMT")} |
| | Compression Type | SNAPPY |
| | Overwrite Files | false |
| | Permission umask | 0 |

For the JSONTreeReader:



| | Name | Type | Bundle | State | Scope | |
|---|---|---|---|---|---|---|
| | AvroSchemaRegistry Meetup | AvroSchemaRegistry 1.9.0.1.0.0.0-... | org.apache.nifi - nifi-registry-nar | Enabled | NiFi Flow | |
| | JsonTreeReader Meetup | JsonTreeReader 1.9.0.1.0.0.0-90 | org.apache.nifi - nifi-record-serializ... | Enabled | NiFi Flow | |

We created a AvroSchemaRegistry and need the state Enabled

| **AvroSchemaRegistry Meetup** | |
|---|---|
| **Settings** | |
| Controller Services | JsonTreeReader Meetup |
| **Properties** | |
| Validate Field Name | true |
| meetup | We put our schema (see below) |

Sample of our Schema (Meetup.avsc)

```
{
  "type" : "record",
  "name" : "meetup_data",
  "namespace" :"nifi",
  "fields" : [ {
    "name" : "venue",
    "type" : {
      "type" : "record",
      "name" : "venue",
      "fields" : [ {
        "name" : "venue_name",
        "type" : "string"

      }, {
        "name" : "lon",
        "type" : "double",
        "doc" : "Type inferred from '1.241895'"
      }, {
        "name" : "lat",
        "type" : "double",
        "doc" : "Type inferred from '51.187927'"
      }, {
        "name" : "venue_id",
        "type" : "int",
        "doc" : "Type inferred from '24532832'"
      } ]
    },
    "doc" : "Type inferred from '{\"venue_name\":\"Chobham Academy \",\"lon\":1.241895,\"lat\":51.187927,\"venue_id\":24532832}'"
  },
  {
    "name" : "visibility",
    "type" : "string",
    "doc" : "Type inferred from '\"public\"'"
  }, {
    "name" : "response",
    "type" : "string",
    "doc" : "Type inferred from '\"yes\"'"
  }, {
    "name" : "guests",
    "type" : "int",
    "doc" : "Type inferred from '0'"
  }, {
```

We created a JsonTreeReader and need the state Enabled

| **AvroSchemaRegistry Meetup** | |
|---|---|
| **Settings** | |
| Controller Services | PutParquet |
| **Properties** | |
| Schema Access Strategy | Use 'Schema Name' Property |
| Schema Registry | AvroSchemaRegisty Meetup |
| Schema Name | ${schema.name} |
| Schema Text | ${avro.schema} |

The parquet files are created in the HDFS directory:

```
%sh
hdfs dfs -ls -R -h /meetup-data
```

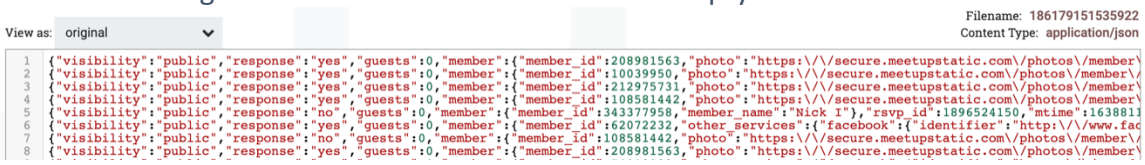
```
drwxrwxrwx   - root supergroup          0 2021-12-06 07:45 /meetup-data/year=2021
drwxrwxrwx   - root supergroup          0 2021-12-06 07:45 /meetup-data/year=2021/month=12
drwxrwxrwx   - root supergroup          0 2021-12-06 08:45 /meetup-data/year=2021/month=12/day=06
drwxrwxrwx   - root supergroup          0 2021-12-06 07:45 /meetup-data/year=2021/month=12/day=06/hour=15
-rw-rw-rw-   1 root supergroup     99.2 K 2021-12-06 07:45 /meetup-data/year=2021/month=12/day=06/hour=15/180579861485102
drwxrwxrwx   - root supergroup          0 2021-12-06 08:45 /meetup-data/year=2021/month=12/day=06/hour=16
-rw-rw-rw-   1 root supergroup     67.1 K 2021-12-06 08:45 /meetup-data/year=2021/month=12/day=06/hour=16/184179901283031
```

- **Empty venues are stored on the local file system for a later usage.**

Link RouteOnAttribute on unmatched to:

| **Processor** | **Properties** | |
|---|---|---|
| **MergeContent**<br><br>Relationships:<br>failure<br>original<br>merged | Merge Strategy | Bin-Packing Algorithm |
| | Merge Format | Binary Concatenation |
| | Attribute Strategy | Keep Only Common Attributes |
| | Metadata Strategy | Do Not Merge Uncommon Metadata |
| | Minimum Number of Entries | 1 |
| | Maximum Number of Entries | 1000000 |
| | Minimum Group Size | 0 B |
| | Maximum number of Bins | 100 |
| | Delimiter Strategy | Text |
| | Demarcator | Shift + enter |
| | Compression Level | 1 |
| | Keep Path | false |
| | **Properties** | |
| | Run Schedule | 3600 sec |

The data is merged in one file. We see the venue is empty:





```
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":208981563,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":10039950,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\/
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":212975731,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":108581442,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\
{"visibility":"public","response":"no","guests":0,"member":{"member_id":343377958,"member_name":"Nick I"},"rsvp_id":1896524150,"mtime":1638811
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":62072232,"other_services":{"facebook":{"identifier":"http:\/\/www.fac
{"visibility":"public","response":"no","guests":0,"member":{"member_id":108581442,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\/
{"visibility":"public","response":"yes","guests":0,"member":{"member_id":208981563,"photo":"https:\/\/secure.meetupstatic.com\/photos\/member\
```


Link on merged to:

| Processor | Properties | |
|---|---|---|
| **PutFile**<br><br>Relationships:<br>failure<br>success | Directory | /home/cloudera/Downloads/meetup-data/Empty/year=${now():format("yyyy", "GMT")}/month=${now():format("MM", "GMT")}/day=${now():format("dd", "GMT")}/hour=${now():format("HH", "GMT")} |
| | Conflict Resolution Strategy | replace |
| | Create Missing Directories | True |
| | Permissions | 777 |

The files are created in the local system:

```
[cloudera@quickstart Empty]$ cd /home/cloudera/Downloads/meetup-data/Empty/
[cloudera@quickstart Empty]$ ll -R
.:
total 4
drwxr-xr-x 3 root root 4096 Dec  6 07:45 year=2021

./year=2021:
total 4
drwxr-xr-x 3 root root 4096 Dec  6 07:45 month=12

./year=2021/month=12:
total 4
drwxr-xr-x 4 root root 4096 Dec  6 08:45 day=06

./year=2021/month=12/day=06:
total 8
drwxr-xr-x 2 root root 4096 Dec  6 07:45 hour=15
drwxr-xr-x 2 root root 4096 Dec  6 08:45 hour=16

./year=2021/month=12/day=06/hour=15:
total 36
-rwxrwxrwx 1 root root 34044 Dec  6 07:45 180582128126002

./year=2021/month=12/day=06/hour=16:
total 24
-rwxrwxrwx 1 root root 23143 Dec  6 08:45 184182133589306
```

# DATA MODELING

## Task 05
## Impala Partitioned Tables

**Create a new Impala user-managed partitioned table named meetup.**

Create directory for the schema
```
%sh
hdfs dfs -mkdir /schema
```

Put the meetup schema in the directory
```
%sh
hdfs dfs -put /home/cloudera/Downloads/Meetup.avsc /schema
```

Create a database
```
%hive
create database meetup
```

Use the database meetup
```
%hive
use meetup
```

Create a table with the schema Meetup.avsc
```
%hive
CREATE EXTERNAL TABLE avro_meetup
 ROW FORMAT SERDE
 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
 STORED AS INPUTFORMAT
 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
 OUTPUTFORMAT
 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
 TBLPROPERTIES ( 'avro.schema.url'='hdfs:///schema/Meetup.avsc')
```

Describe the table created
```
%hive
DESCRIBE avro_meetup
```

| col_name | data_type | comment |
|---|---|---|
| venue | struct<venue_name:string,lon:double,lat:double,venue_id:int> | Type inferred from '{"venue_name":"Chobham Academy ","lon":1.241895,"lat":51.187927,"venue_id":24532832}' |
| visibility | string | Type inferred from '"public"' |
| response | string | Type inferred from '"yes"' |
| guests | int | Type inferred from '0' |
| member | struct<member_id:int,photo:string,member_name:string> | Type inferred from '{"member_id":161222912,"photo":"https://secure.meetupstatic.com/photos/member/9/7/f/f/thumb_263498911.jpeg","member_name":"aj"}' |
| rsvp_id | int | Type inferred from '1887269644' |

Create the table meetup with the parquet file

```
%hive
-- Create the table meetup with the parquet file
CREATE EXTERNAL TABLE if not exists meetup
LIKE avro_meetup
STORED AS PARQUET
LOCATION '/meetup-data/year=2021/month=12/day=06/hour=16/'
TBLPROPERTIES ("parquet.compression"="SNAPPY")
```

Show 5 lines of the table

```
%hive
select * from meetup limit 5
```

| meetup.venue | meetup.visibility | meetup.response | meetup.guests | meetup.member | meetup.rsvp_id | meetup.mtime | meetup.event | m |
|---|---|---|---|---|---|---|---|---|
| {"venue_name":"OHenry's Coffee","lon":-86.77303,"lat":33.468494,"venue_id":27231081} | public | yes | 0 | {"member_id":334391720,"photo":"https://secure.meetupstatic.com/photos/member/b/2/1/6/thumb_305505590.jpeg","member_name":"Lexi "} | 1896543150 | 1638805512638 | {"event_name":"The Hunting Party" Book Club Meeting ","event_id":"282527188","time":1641654000000,"event_url":"https://www.meetup.com/mocha-girls-read-birmingham/events/282527188/"} | [{ o B {' d m D {' p , {' n |

Refresh the metadata in impala

```
%impala
invalidate metadata
```

Use database meetup in impala

```
%impala
use meetup
```

Compute stats of the table

```
%impala
compute stats meetup
```

| summary |
|---|
| Updated 1 partition(s) and 5 column(s). |

Show the stats of the table to see the number of rows and the size

```
%impala
show table stats meetup
```

| #Rows | #Files | Size | Bytes Cached | Cache Replication | Format | Incremental stats | Location |
|---|---|---|---|---|---|---|---|
| 193 | 1 | 67.11KB | NOT CACHED | NOT CACHED | PARQUET | false | hdfs://quickstart.cloudera:8020/meetup-data/year=2021/month=12/day=06/hour=16 |

Rows: 193
Size: 67.11KB

# DATA ANALYSIS

## Task 06
## Basic meetup rsvps analysis

venues rows count

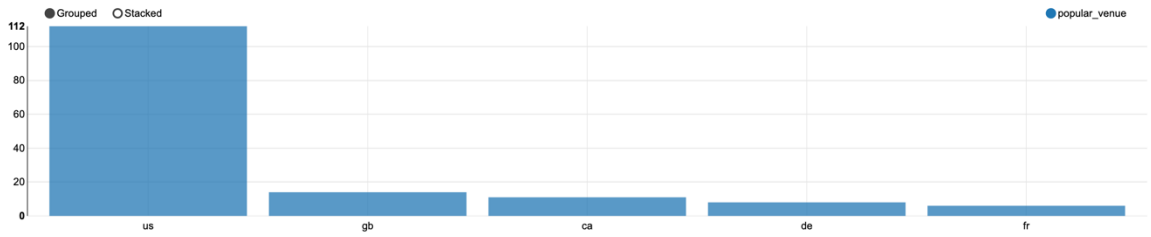
```
%impala
select count(venue.venue_name) from meetup
```

| count(venue.venue_name) |
|---|
| 193 |

Most popular venue by country location

```
%impala
select  `group`.group_country, count(*) as popular_venue
from meetup
group by `group`.group_country
order by popular_venue desc
```

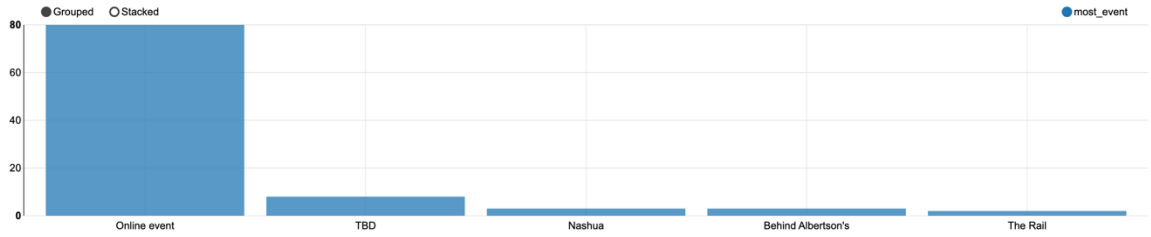
| group.group_country | popular_venue |
|---|---|
| us | 112 |
| gb | 14 |
| ca | 11 |
| de | 8 |
| fr | 6 |
| cz | 5 |
| es | 3 |
| nl | 3 |




venue has the most event

```
%impala
select  venue.venue_name, count(event.event_id) as most_event
from meetup
group by venue.venue_name
order by most_event desc
```

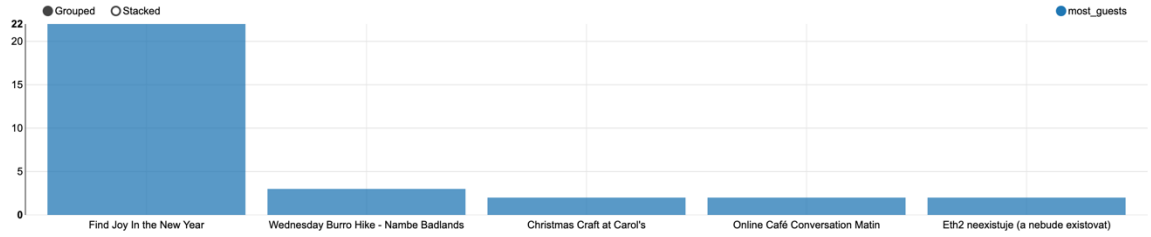
| venue.venue_name | most_event |
|---|---|
| Online event | 80 |
| TBD | 8 |
| Nashua | 3 |
| Behind Albertson's | 3 |
| Online with Zoom! | 2 |
| Ace Hotel Downtown Los Angeles | 2 |
| Paralelní Polis | 2 |
| Laguna Niguel Regional Park | 2 |




Event has the most guest

```
%impala
select  event.event_id as event_id,event.event_name as event_name,
        count(guests) as most_guests
from meetup
group by event.event_id,event.event_name
order by most_guests desc
```

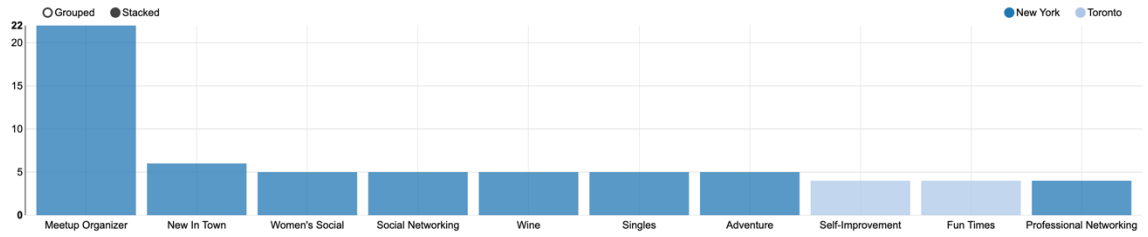
| event_id | event_name | most_guests |
|---|---|---|
| 281509839 | Find Joy In the New Year | 22 |
| 282517856 | Wednesday Burro Hike - Nambe Badlands | 3 |
| 282526332 | Tools voor learning experience design | 2 |
| 272057281 | IN-PERSON Full Day Workshop: Systemic & Family Constellations with Illi Adato | 2 |
| 282344376 | Lean Coffee São Paulo #113 (ONLINE) | 2 |
| 282525401 | Eth2 neexistuje (a nebude existovat) | 2 |



hotest topics in the given city

```
%impala
select `group`.group_city as group_city,
        grouptopics.topic_name as topic_name,
        count(grouptopics.topic_name) hottest_topics_city
from meetup,
    meetup.`group`.group_topics  as grouptopics
group by `group`.group_city, grouptopics.topic_name
order by hottest_topics_city desc
```

| group_city | topic_name | hottest_topics_city |
|---|---|---|
| New York | Meetup Organizer | 22 |
| New York | New In Town | 6 |
| New York | Women's Social | 5 |
| New York | Singles | 5 |
| New York | Adventure | 5 |
| New York | Social Networking | 5 |
| New York | Wine | 5 |
| London | New In Town | 4 |



# Task 07
# a) Basic SparkSQL meetup rsvps analysis

- **Load the meetup parquet file**

Generate SQLContext using the following command
```
%spark
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
```

Create an RDD DataFrame by reading a data from the parquet file
```
%spark
val parqfile =
sqlContext.read.parquet("/meetupdata/year=2021/month=12/day=06/hour=16/")
```

Use the following command for storing the DataFrame data into a table named meetup
```
%spark
parqfile.registerTempTable("meetup")
```

View 5 lines of the table
```
%spark
parqfile.show(5)
```

```
+--------------------+----------+--------+------+--------------------+----------+------------+--------------------+--------------------+
|               venue|visibility|response|guests|              member|   rsvp_id|       mtime|               event|               group|
+--------------------+----------+--------+------+--------------------+----------+------------+--------------------+--------------------+
|[OHenry's Coffee,...|    public|     yes|     0|[334391720,https:...|1896543150|1638805512638|[“The Hunting Par...|[WrappedArray([cl...|
|[Calle de Génova;...|    public|     yes|     0|[187601992,https:...|1896543151|1638805513902|[DE MERIENDA,2825...|[WrappedArray([si...|
|[Online event,179...|    public|     yes|     0|[294755969,https:...|1896543152|1638805515133|[[Online] How to ...|[WrappedArray([ph...|
|[Bouldin Acres,-9...|    public|     yes|     0|[333035850,https:...|1896543154|1638805515492|[WOMEN ONLY! Spee...|[WrappedArray([fi...|
|[Painting With a ...|    public|      no|     0|[332571908,https:...|1895193035|1638805515788|[Black Women Holi...|[WrappedArray([ad...|
+--------------------+----------+--------+------+--------------------+----------+------------+--------------------+--------------------+
only showing top 5 rows
```

- **Report the number of unique topics**

Use the following command for storing the distinct count of group_topics
```
%spark
val allrecords = sqlContext.sql("SELECT count(distinct group.group_topics) as topics
FROM meetup")
```

Show the result of count
```
%spark
allrecords.show()
```

```
+------+
|topics|
+------+
|   143|
+------+
```

# b) Basic Spark Structured Streaming analysis

- **Read data from meetup-agg Kafka topic**

install Spark Streaming dependencies
1 - upload the jars files to the VM (e.g Downloads)
2 - run the following commands on this directory

```
sudo cp spark-streaming-kafka-0-10_2.11-2.2.1.jar /usr/lib/zeppelin/interpreter/spark/dep/
sudo chmod 777 /usr/lib/zeppelin/interpreter/spark/dep/spark-streaming-kafka-0-10_2.11-2.2.1.jar

sudo cp spark-sql-kafka-0-10_2.11-2.1.1.jar /usr/lib/zeppelin/interpreter/spark/dep/
sudo chmod 777 /usr/lib/zeppelin/interpreter/spark/dep/spark-sql-kafka-0-10_2.11-2.1.1.jar

sudo cp kafka-clients-0.11.0.1.jar /usr/lib/zeppelin/interpreter/spark/dep/
sudo chmod 777 /usr/lib/zeppelin/interpreter/spark/dep/kafka-clients-0.11.0.1.jar
```

Load all the dependencies in zeppelin (need run this code at the beginning)
This will let us connect Spark Streaming to kafka topic

```
%dep
z.load("spark-streaming-kafka-0-10_2.11-2.2.1.jar")
z.load("spark-sql-kafka-0-10_2.11-2.1.1.jar")
z.load("kafka-clients-0.11.0.1.jar")
```

Import Kafka and Spark Streaming libraries

```
%spark
import org.apache.spark.streaming._
import org.apache.spark.sql.types._
```

Define a case class to hold meetup-agg

```
%spark
case class meetup
  (
  venue_name: String,
  guest: Int,
  response: String
  )
```

Create the kafka Consumer
The consumer will read from kafka movie topic

```
%spark
val kafkaStream = spark
   .readStream
   .format("kafka")
   .option("kafka.bootstrap.servers","localhost:9092")
   .option("subscribe","meetup-agg")
   .load()
```

Print the Schema
The 'value' column contains the meetup data in binary format

```
%spark
kafkaStream.printSchema
```

```
root
 |-- key: binary (nullable = true)
 |-- value: binary (nullable = true)
 |-- topic: string (nullable = true)
 |-- partition: integer (nullable = true)
 |-- offset: long (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- timestampType: integer (nullable = true)
```

We are interested by the 'value' column
Let's cast this as String

```
%spark
val dataStream = kafkaStream.selectExpr("CAST(value AS STRING)").as[String]

%spark
val meetupDataStream = dataStream.map(row => row.split(","))
                        .map(
                                row => meetup(
                                        row(0),                 //venue_name
                                        row(1).trim.toInt,      //guest
                                        row(2)                  //response
                                )
                        )
```

Create a Tempory View to run your SparkSQL queries on the data

```
%spark
meetupDataStream.createOrReplaceTempView("meetup")
```

AGGREGATION

```
%spark
val meetupCount = meetupDataStream
.groupBy($"venue_name")
.agg(sum("guest").alias("total_guest"), count("response").alias("response"))
.sort("venue_name")
```

- **Report (on the console) the guest count and response per venue**

Console

********** AGGREGATION **********

```
%spark
meetupCount
  .writeStream
    .format("console")
    .outputMode("complete")
    .start()
    .awaitTermination()
```

```
-------------------------------------------
Batch: 1
-------------------------------------------
+--------------------+-----------+--------+
|          venue_name|total_guest|response|
+--------------------+-----------+--------+
|"Wilson Road Lot ...|          0|       1|
|14895 Little Tuju...|          0|       1|
|          240 Elm St|          0|       1|
|               Agora|          0|       2|
|AnneMarie Tennis ...|          0|       1|
|Broadway Comedy Club|         25|       1|
|Broadway Performi...|          0|       1|
|Classic Cinemas T...|          0|       1|
|Dorking Railway S...|          0|       1|
|Farmington Commun...|          0|       1|
```

# PLATFORME INTEGRATION

## Task 08
## Show meetup data in MS Excel

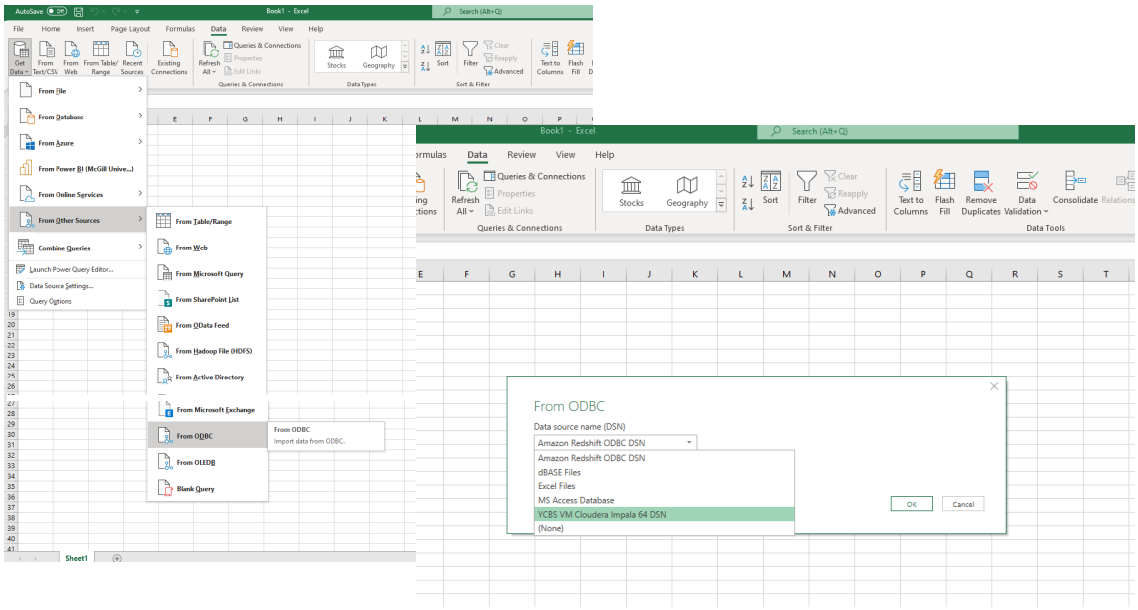
Create view (meetup_view) based on meetup table

```
%impala
create view meetup_view as
select meetup.`group`.group_country as country,
       meetup.venue.venue_name as venue_name, grouptopics.topic_name as topic_name,
       meetup.`group`.group_name as group_name
from meetup,
   meetup.`group`.group_topics  as grouptopics
   limit 50
```

Create view (meetup_view_count) based on meetup table

```
%impala
create view meetup_view_count as
select meetup.`group`.group_country as country,
       count(meetup.venue.venue_name) as
       cnt_venue_name,count(grouptopics.topic_name) as
       cnt_topic_name,count(meetup.`group`.group_name) as cnt_group_name
from meetup,
   meetup.`group`.group_topics  as grouptopics
   group by meetup.`group`.group_country
   limit 50
```

ODBC connection



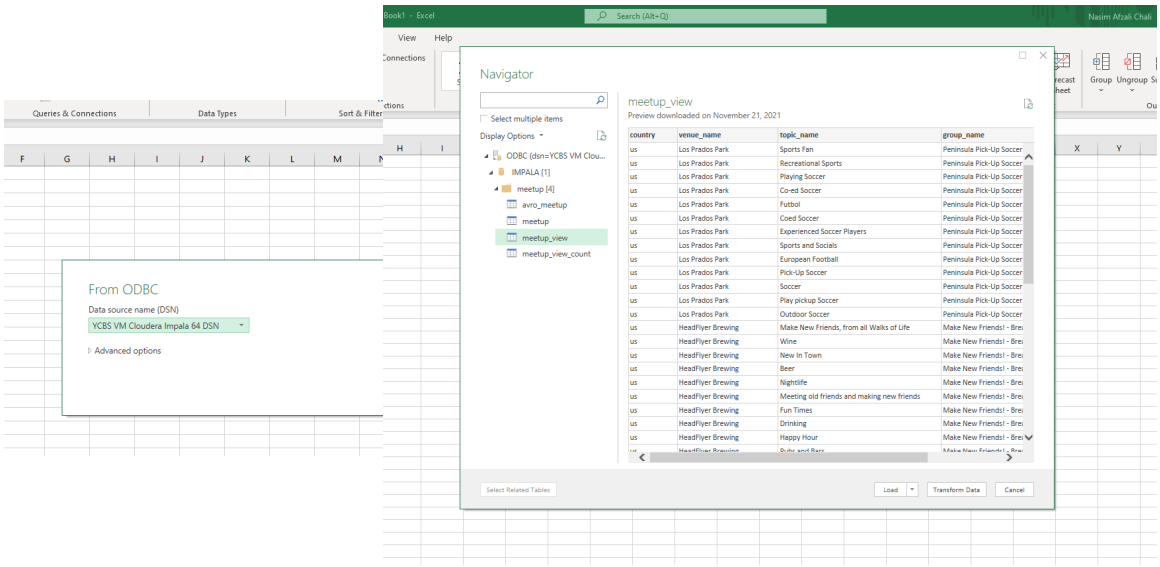
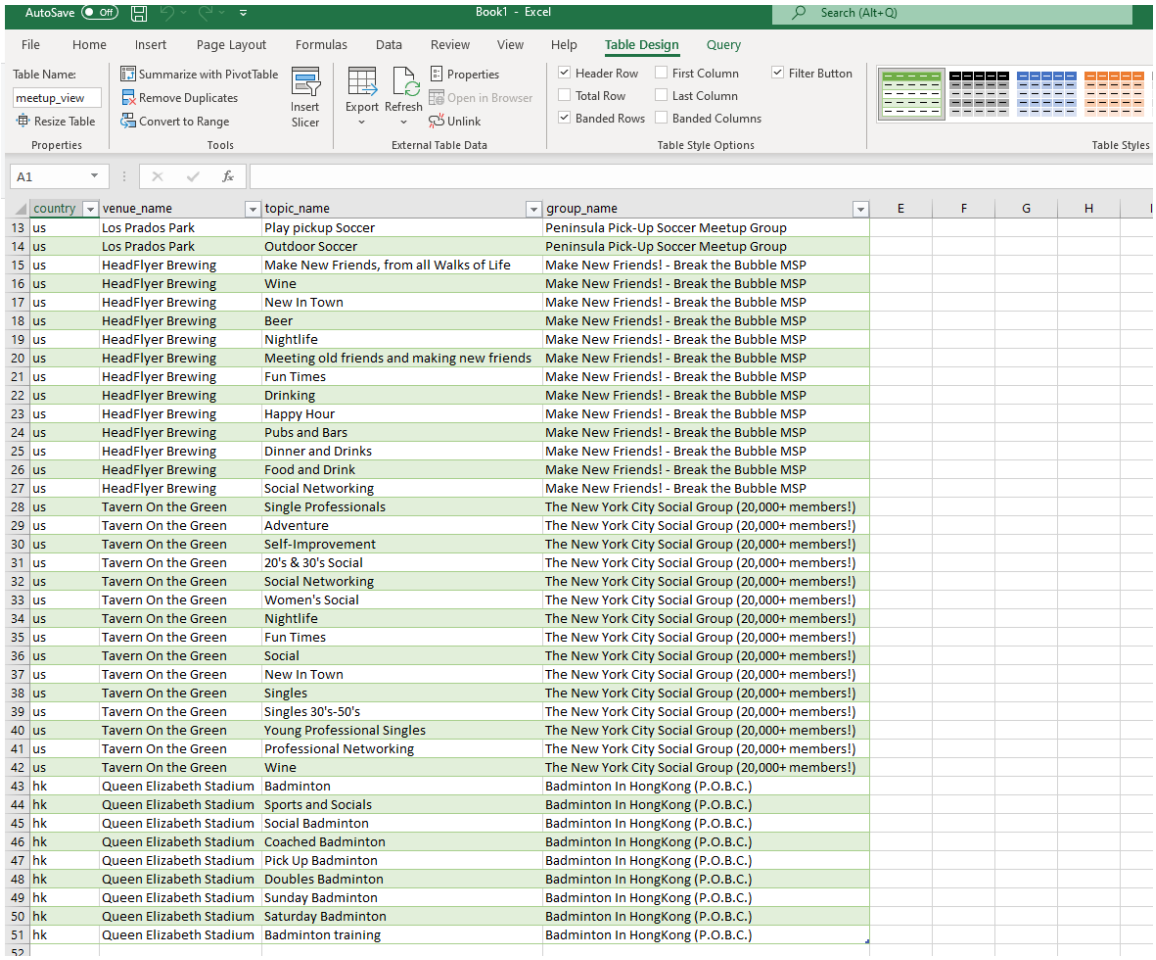
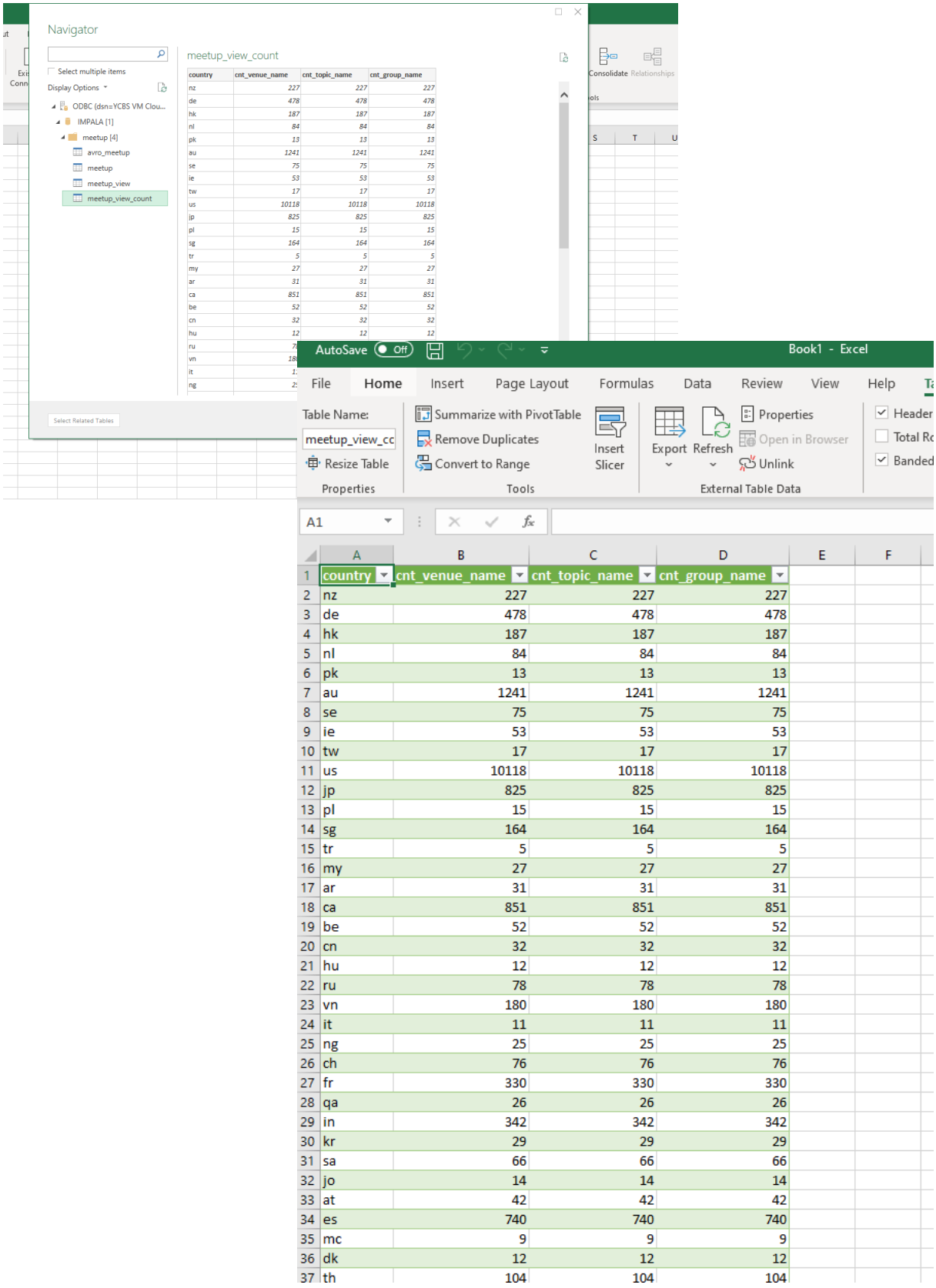



Table meetup_view:



| | country | venue_name | topic_name | group_name |
|---|---|---|---|---|
| 13 | us | Los Prados Park | Play pickup Soccer | Peninsula Pick-Up Soccer Meetup Group |
| 14 | us | Los Prados Park | Outdoor Soccer | Peninsula Pick-Up Soccer Meetup Group |
| 15 | us | HeadFlyer Brewing | Make New Friends, from all Walks of Life | Make New Friends! - Break the Bubble MSP |
| 16 | us | HeadFlyer Brewing | Wine | Make New Friends! - Break the Bubble MSP |
| 17 | us | HeadFlyer Brewing | New In Town | Make New Friends! - Break the Bubble MSP |
| 18 | us | HeadFlyer Brewing | Beer | Make New Friends! - Break the Bubble MSP |
| 19 | us | HeadFlyer Brewing | Nightlife | Make New Friends! - Break the Bubble MSP |
| 20 | us | HeadFlyer Brewing | Meeting old friends and making new friends | Make New Friends! - Break the Bubble MSP |
| 21 | us | HeadFlyer Brewing | Fun Times | Make New Friends! - Break the Bubble MSP |
| 22 | us | HeadFlyer Brewing | Drinking | Make New Friends! - Break the Bubble MSP |
| 23 | us | HeadFlyer Brewing | Happy Hour | Make New Friends! - Break the Bubble MSP |
| 24 | us | HeadFlyer Brewing | Pubs and Bars | Make New Friends! - Break the Bubble MSP |
| 25 | us | HeadFlyer Brewing | Dinner and Drinks | Make New Friends! - Break the Bubble MSP |
| 26 | us | HeadFlyer Brewing | Food and Drink | Make New Friends! - Break the Bubble MSP |
| 27 | us | HeadFlyer Brewing | Social Networking | Make New Friends! - Break the Bubble MSP |
| 28 | us | Tavern On the Green | Single Professionals | The New York City Social Group (20,000+ members!) |
| 29 | us | Tavern On the Green | Adventure | The New York City Social Group (20,000+ members!) |
| 30 | us | Tavern On the Green | Self-Improvement | The New York City Social Group (20,000+ members!) |
| 31 | us | Tavern On the Green | 20's & 30's Social | The New York City Social Group (20,000+ members!) |
| 32 | us | Tavern On the Green | Social Networking | The New York City Social Group (20,000+ members!) |
| 33 | us | Tavern On the Green | Women's Social | The New York City Social Group (20,000+ members!) |
| 34 | us | Tavern On the Green | Nightlife | The New York City Social Group (20,000+ members!) |
| 35 | us | Tavern On the Green | Fun Times | The New York City Social Group (20,000+ members!) |
| 36 | us | Tavern On the Green | Social | The New York City Social Group (20,000+ members!) |
| 37 | us | Tavern On the Green | New In Town | The New York City Social Group (20,000+ members!) |
| 38 | us | Tavern On the Green | Singles | The New York City Social Group (20,000+ members!) |
| 39 | us | Tavern On the Green | Singles 30's-50's | The New York City Social Group (20,000+ members!) |
| 40 | us | Tavern On the Green | Young Professional Singles | The New York City Social Group (20,000+ members!) |
| 41 | us | Tavern On the Green | Professional Networking | The New York City Social Group (20,000+ members!) |
| 42 | us | Tavern On the Green | Wine | The New York City Social Group (20,000+ members!) |
| 43 | hk | Queen Elizabeth Stadium | Badminton | Badminton In HongKong (P.O.B.C.) |
| 44 | hk | Queen Elizabeth Stadium | Sports and Socials | Badminton In HongKong (P.O.B.C.) |
| 45 | hk | Queen Elizabeth Stadium | Social Badminton | Badminton In HongKong (P.O.B.C.) |
| 46 | hk | Queen Elizabeth Stadium | Coached Badminton | Badminton In HongKong (P.O.B.C.) |
| 47 | hk | Queen Elizabeth Stadium | Pick Up Badminton | Badminton In HongKong (P.O.B.C.) |
| 48 | hk | Queen Elizabeth Stadium | Doubles Badminton | Badminton In HongKong (P.O.B.C.) |
| 49 | hk | Queen Elizabeth Stadium | Sunday Badminton | Badminton In HongKong (P.O.B.C.) |
| 50 | hk | Queen Elizabeth Stadium | Saturday Badminton | Badminton In HongKong (P.O.B.C.) |
| 51 | hk | Queen Elizabeth Stadium | Badminton training | Badminton In HongKong (P.O.B.C.) |

meetup-view_count (Per country / venues, group, topic)




| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | country | cnt_venue_name | cnt_topic_name | cnt_group_name | | |
| 2 | nz | 227 | 227 | 227 | | |
| 3 | de | 478 | 478 | 478 | | |
| 4 | hk | 187 | 187 | 187 | | |
| 5 | nl | 84 | 84 | 84 | | |
| 6 | pk | 13 | 13 | 13 | | |
| 7 | au | 1241 | 1241 | 1241 | | |
| 8 | se | 75 | 75 | 75 | | |
| 9 | ie | 53 | 53 | 53 | | |
| 10 | tw | 17 | 17 | 17 | | |
| 11 | us | 10118 | 10118 | 10118 | | |
| 12 | jp | 825 | 825 | 825 | | |
| 13 | pl | 15 | 15 | 15 | | |
| 14 | sg | 164 | 164 | 164 | | |
| 15 | tr | 5 | 5 | 5 | | |
| 16 | my | 27 | 27 | 27 | | |
| 17 | ar | 31 | 31 | 31 | | |
| 18 | ca | 851 | 851 | 851 | | |
| 19 | be | 52 | 52 | 52 | | |
| 20 | cn | 32 | 32 | 32 | | |
| 21 | hu | 12 | 12 | 12 | | |
| 22 | ru | 78 | 78 | 78 | | |
| 23 | vn | 180 | 180 | 180 | | |
| 24 | it | 11 | 11 | 11 | | |
| 25 | ng | 25 | 25 | 25 | | |
| 26 | ch | 76 | 76 | 76 | | |
| 27 | fr | 330 | 330 | 330 | | |
| 28 | qa | 26 | 26 | 26 | | |
| 29 | in | 342 | 342 | 342 | | |
| 30 | kr | 29 | 29 | 29 | | |
| 31 | sa | 66 | 66 | 66 | | |
| 32 | jo | 14 | 14 | 14 | | |
| 33 | at | 42 | 42 | 42 | | |
| 34 | es | 740 | 740 | 740 | | |
| 35 | mc | 9 | 9 | 9 | | |
| 36 | dk | 12 | 12 | 12 | | |
| 37 | th | 104 | 104 | 104 | | |

# Task 09
# Sqoop Export

**Export data into a MySQL.**
We use an existing database (mysql)

Use sqoop to list all databases on the running MySQL local instance
```
%sh
sqoop list-databases --connect jdbc:mysql://localhost/ --username root --password
cloudera
```

```
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/12/05 15:23:30 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
21/12/05 15:23:30 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
21/12/05 15:23:31 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
```

Use mysql database for create the table
```
%mysql
use mysql
```

**Write a Sqoop command to export all venues in the US from the meetup table**

Create database sqoop
```
%hive
create database sqoop
```

Use the database sqoop
```
%hive
use sqoop
```

Create table meetup_us in sqoop database
```
%hive
create table sqoop.meetup_us
row format delimited fields terminated by ','
as
select cast(venue.venue_name as varchar(100)) as venue_name,
      cast(venue.lon as double) as longitude,
      cast(venue.lat as double) as lattitude,
      cast(`group`.group_country as varchar(5)) as group_country,
      cast(`group`.group_city as varchar(35)) as group_city,
```

```
        cast(guests as int) as guest,
        cast(mtime as bigint) as mtime
from meetup
where group.group_country ='us'
```

View 5 lines on the table meetup_us

```
%hive
select * from sqoop.meetup_us limit 5
```

| meetup_us.venue_name | meetup_us.longitude | meetup_us.lattitude | meetup_us.group_country | meetup_us.group_city | meetup_us.guest | meetup_us.mtime |
|---|---|---|---|---|---|---|
| OHenry's Coffee | -86.77303 | 33.468494 | us | Birmingham | 0 | 1638805512638 |
| Online event | 179.1962 | -8.521147 | us | Las Vegas | 0 | 1638805515133 |
| Bouldin Acres | -97.76865 | 30.248215 | us | Austin | 0 | 1638805515492 |
| Painting With a Twist | -95.411377 | 29.80267 | us | Houston | 0 | 1638805515788 |
| Online event | 179.1962 | -8.521147 | us | New York | 0 | 1638805518106 |

Count the number in line from meetup_us created in hive

```
%hive
select count(*) from sqoop.meetup_us
```

| cnt |
|---|
| 112 |

The target MySQL table must be existed in the target database prior to run the Sqoop export command

```
%mysql
CREATE TABLE mysql.meetup (
        venue_name varchar(100),
        longitude double,
        latitude double,
        country varchar(5),
        city varchar(35),
        guests int,
        mtime bigint)
```

Check recursively the directory of the warehouse table meetup_us

```
%sh
hdfs dfs -ls -R /user/hive/warehouse/sqoop.db/
```

```
drwxrwxrwx   - hive supergroup          0 2021-12-05 15:24 /user/hive/warehouse/sqoop.db/avro_meetup
drwxrwxrwx   - hive supergroup          0 2021-12-06 11:39 /user/hive/warehouse/sqoop.db/meetup_us
-rwxrwxrwx   1 hive supergroup       7130 2021-12-06 11:39 /user/hive/warehouse/sqoop.db/meetup_us/000000_0
```

Export the table with sqoop

```
%sh
sqoop export --connect jdbc:mysql://localhost/mysql --username root --password
cloudera --table meetup --export-dir /user/hive/warehouse/sqoop.db/meetup_us --input-
fields-terminated-by ","
```

Show 5 lines of the table meetup in mysql
%mysql
select * from mysql.meetup limit 5

| venue_name | longitude | latitude | country | city | guests | mtime |
|---|---|---|---|---|---|---|
| OHenry's Coffee | -86.77303 | 33.468494 | us | Birmingham | 0 | 1638805512638 |
| Online event | 179.1962 | -8.521147 | us | Las Vegas | 0 | 1638805515133 |
| Bouldin Acres | -97.76865 | 30.248215 | us | Austin | 0 | 1638805515492 |
| Painting With a Twist | -95.411377 | 29.80267 | us | Houston | 0 | 1638805515788 |
| Online event | 179.1962 | -8.521147 | us | New York | 0 | 1638805518106 |

**Report rows count from the target table**

Count the number of lines (same number of lines of the table created in hive)
%mysql
select count(*) as cnt from mysql.meetup

| cnt |
|---|
| 112 |

View the list of all tables in mysql database
%mysql
sqoop list-tables --connect jdbc:mysql://localhost/mysql --username root --password cloudera

```
func
general_log
help_category
help_keyword
help_relation
help_topic
host
meetup
```