*Alka Patel*

*Rodrigo Pedro Batista de Oliveira*

**Travel Data Analysis**

# Database Design Document

**Version 3.0**

**03/22/2019**

**Table of Contents**

[**Database Design Document**](#_15g0kxf0nq9v) **1**

[**Assumptions/Constraints/Risks**](#_2et92p0) **3**

[Assumptions](#_tyjcwt) 3

[Constraints](#_3dy6vkm) 3

[Risks](#_1t3h5sf) 4

[**Design Decisions**](#_4d34og8) **4**

[Key Factors Influencing Design](#_2s8eyo1) 5

[Security and Privacy Design Decisions](#_26in1rg) 6

[**Detailed Database Design**](#_35nkun2) **6**

[Roles and Responsibilities](#_1ksv4uv) 9

[Performance Monitoring and Database Efficiency](#_44sinio) 11

[Data Transfer Requirements](#_z337ya) 11

[Data Formats](#_3j2qqm3) 11

**List of Tables**

Appendix A [- Acronyms](#qsh70q) 13

Appendix B - Improvements**Overview**

## 

**Context:**

We are an Airline company and want to get a grasp on the factors that can cause an increase or decrease in ticket sales. We would like to follow international trends to analyze how they promote or adversely affect travel. A current example would be to stream live twitter (travel-related) data on China and Canada relationships as an analysis of the recent happenings of Canada arresting the Chinese Huawei executive, Meng Wanzhou, in Vancouver in early December. We can see how this has impacted the Canadian-Chinese relationships in relation to travel behaviours (business and pleasure).

From twitter, we can extract comments, and do a sentiment analysis of people commenting on the situation in China and Canada specifically in regards to “travel”. We can extract additional information like where the tweets are coming from, how many likes they have, how many retweets there were etc.

## Assumptions/Constraints/Risks

### Assumptions

Our project constitutes an efficient (but not extravagant) infrastructure set up to optimize the extraction, storage, cleaning, and analysis of warm temperature data. The exact technologies are described in the document below.

Below are the basic, main steps:

Source

Extract

Data to analyze

Analyze

### Constraints

* **Need for talent:** Data scientists and big data experts are among the most highly coveted —and highly paid — workers in the IT field. Hiring or training staff can increase costs considerably.
* **Data quality:** Data scientists and analysts need to ensure that the information they are using is accurate, relevant and in the proper format for analysis.
* **Hardware needs:** We need to have theIT infrastructure necessary to support Travel Data Analysis initiatives. Storage space to house the data, networking bandwidth to transfer it, and compute resources to perform those analytics are expensive to purchase and maintain.
* **Costs:** We are going to useopen source technology, which dramatically reduces software costs, but we still face significant expenses related to staffing, hardware, maintenance and related services.

### Risks

We need to take in consideration that a data model for a columnar-oriented database is different from an analogous model designed for an DBMS.

In order to achieve the same capabilities that a relational database provides on tables, we need to model our data in a method that will allow us to query and analyze the data as efficiently.

## Design Decisions

For the purposes of this project, we will focus on a SSD server based approach.

* As we will be handling Live Stream Tweets, our set up for the Cassandra database will be the following:
  + Data Temperature: **Warm Data**
    - While the tweet data does not need to be monitored consistently, it would be analyzed on a biweekly (twice a week) basis for the marketing team to plan marketing strategies for the upcoming days.
    - Additionally, all the trends will be compared on quarterly and annual basis. So the data needs to be easily and readily available.
  + Type of Storage: **SSD**

Because of our option for **Warm Data**, we will set up our storage as an **SSD** type. We want to frequently access this data for analysis purposes. Moreover, we would want to see the trends over time. (Putting it on RAM would mean we lose the data and on HDD, it would be too far away and cumbersome to retrieve.)

Lastly, the business decisions will be made by the marketing team, and we would provide a tableau report for them that would have the following columns below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Topic | No. of Tweets | No. of Likes | No. of Retweets | Location of Tweet | Travel Origin Country | Travel Destination Country | Sentiment: Yes / No to Travel | Date |
|  |  |  |  |  |  |  |  |  |

### Key Factors Influencing Design

Currently, we will use the **tweepy API for the authentication and access to the twitter endpoints** (methods called on a spark script) and the logic of our extraction will be done in **Spark for the ETL to Cassandra**. We would also want to set up a tableau view report for our marketing team. The **near real time streaming** will be orchestrated using SPARK.

* **Our DB setup would work best with BASE properties.**
  + Because we want the latest and the greatest of trends on twitter, we care more about the **speed**, and **throughput** rather than consistency. We also want to keep things simple and **avoid locks**. Twitter data may not always be consistent, and we care more about the **semantic trend analysis** than a numerical hard calculation. Therefore, we believe the **BASE** set up is best.

* **Cassandra:**
  + On **Eric Brewer’s CAP**, our project falls into the **AP category**. We care most about **availability and partition-tolerance**, as opposed to consistency. We need the system to continue operating despite node failures. Additionally, because we want to **avoid bottlenecks** and insure continuous functionality despite network failures, we also need partition tolerance. Basically, **no downtime** should be felt! Therefore, we believe **Cassandra** is the best set up. It will allow us the columnar set up we need to do analysis over time. Additionally, Cassandra is also **scalable;** if business grows, or there a multiple social/political trends affecting the market at a given time, we know we can scale using this architectural choices.

### Security and Privacy Design Decisions

The types of access we can have are:

* **Full:** Complete access to delete, update and create tables
  + Executives, DBA team
* **Intermediate:** Querying and Viewing: Manipulate the data from the ssd, and view everything on the RAM.
  + Data Analysts/ Data Scientists
* **Limited:** View access of Tabulated and Queried Data
  + Marketing Team

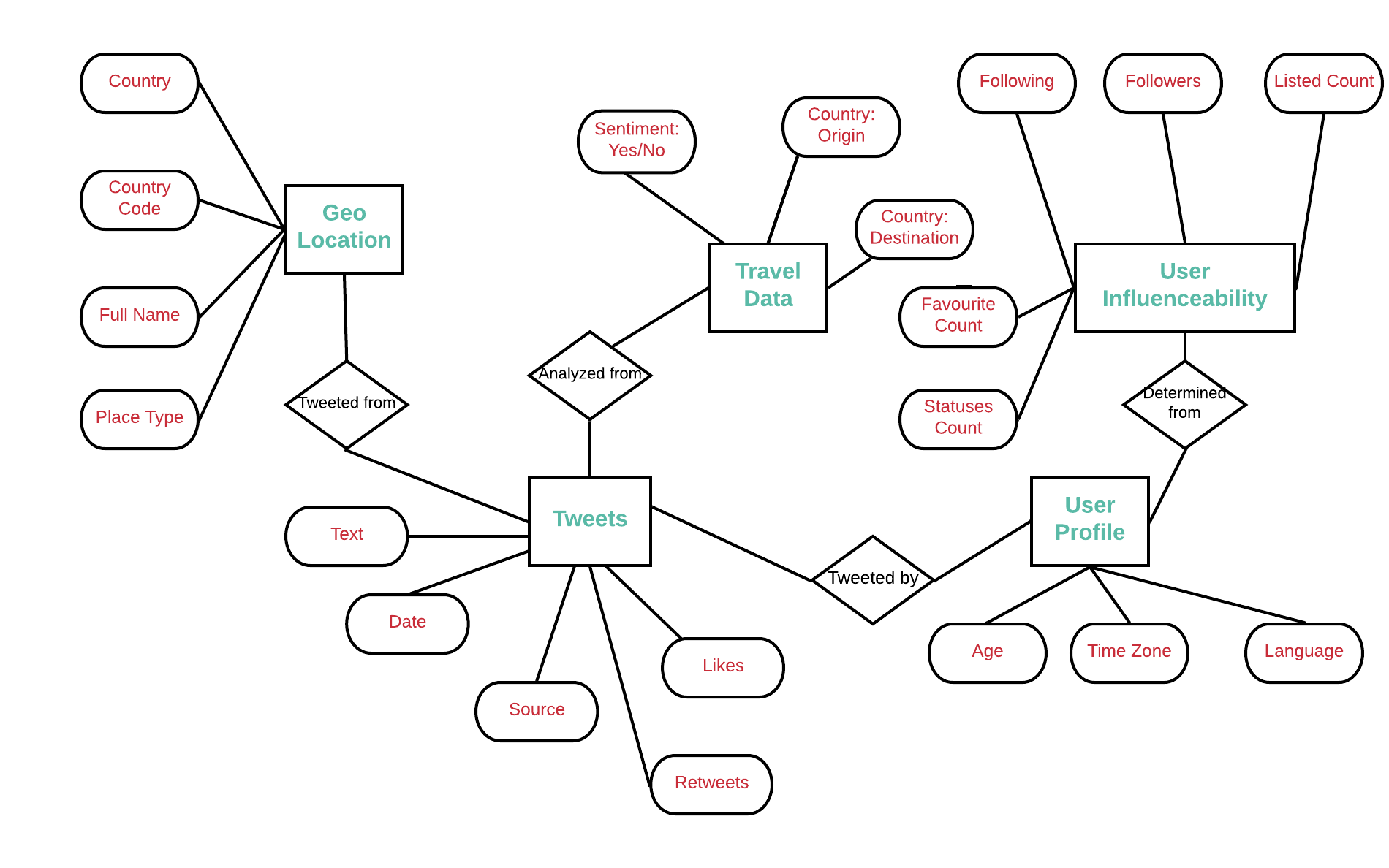
**Ref:** <http://cassandra.apache.org/doc/latest/operating/security.html?highlight=security>

The following security features are provided from Cassandra, and we deem that as satisfactory for our purposes:

* **TLS/SSL Encryption**
  + Ensures data in flight is not compromised and is transferred securely.
* **Roles**
  + We can create different user (or group) roles using this feature
* **Password Authentication**
  + When a server is launched, our database will have authentication enabled for additional safety.

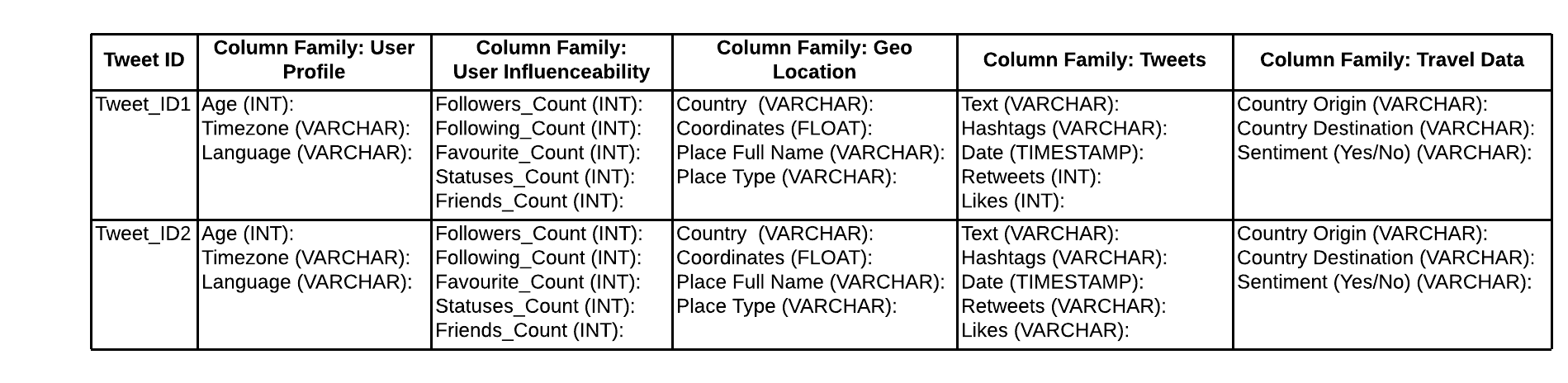
## Detailed Database Design

* **Conceptual Data Model (CDM)**
  + Our project is based on Tweets feed, the concept design for it would be the following (as per our endpoint file):



* **Logical Data Model (LDM) and LDM Entity Relationship Diagram (ERD)**.
  + Our database will be **Columnar based NoSql with Cassandra**

-ERD would be something like the figure below:

**

* **Physical Data Model (PDM) with a description of the DBMS schemas, sub-schemas, records, sets, tables.**
  + **DDLs for the columnar Database:**

CREATE TYPE user\_profile (

age INT,

timezone VARCHAR,

language VARCHAR

);

CREATE TYPE user\_influenceability (

followers\_count INT,

following\_count INT,

favourite\_count INT,

statuses\_count INT,

friends\_count INT

);

CREATE TYPE geo\_location (

country VARCHAR,

coordinates FLOAT,

place\_full\_name VARCHAR,

place\_type VARCHAR

);

CREATE TYPE tweets (

text VARCHAR,

hashtags VARCHAR,

date TIMESTAMP,

retweets INT,

likes INT

);

CREATE TYPE travel\_data (

country\_origin VARCHAR,

country\_destination VARCHAR,

travel\_sentiment VARCHAR

);

CREATE TABLE tweet\_travel\_analysis (

tweet\_id INT,

user\_profile frozen<user\_profile>

user\_influenceability frozen<user\_influenceability>

geo\_location frozen<geo\_location>

tweets frozen<tweets>

travel\_data frozen<travel\_data>

);

**To Orchestrate and execute the process we will be using a Spark script that will call methods to authenticate Tweepy, and extract the data into our Cassandra database at 1 minute intervals.**

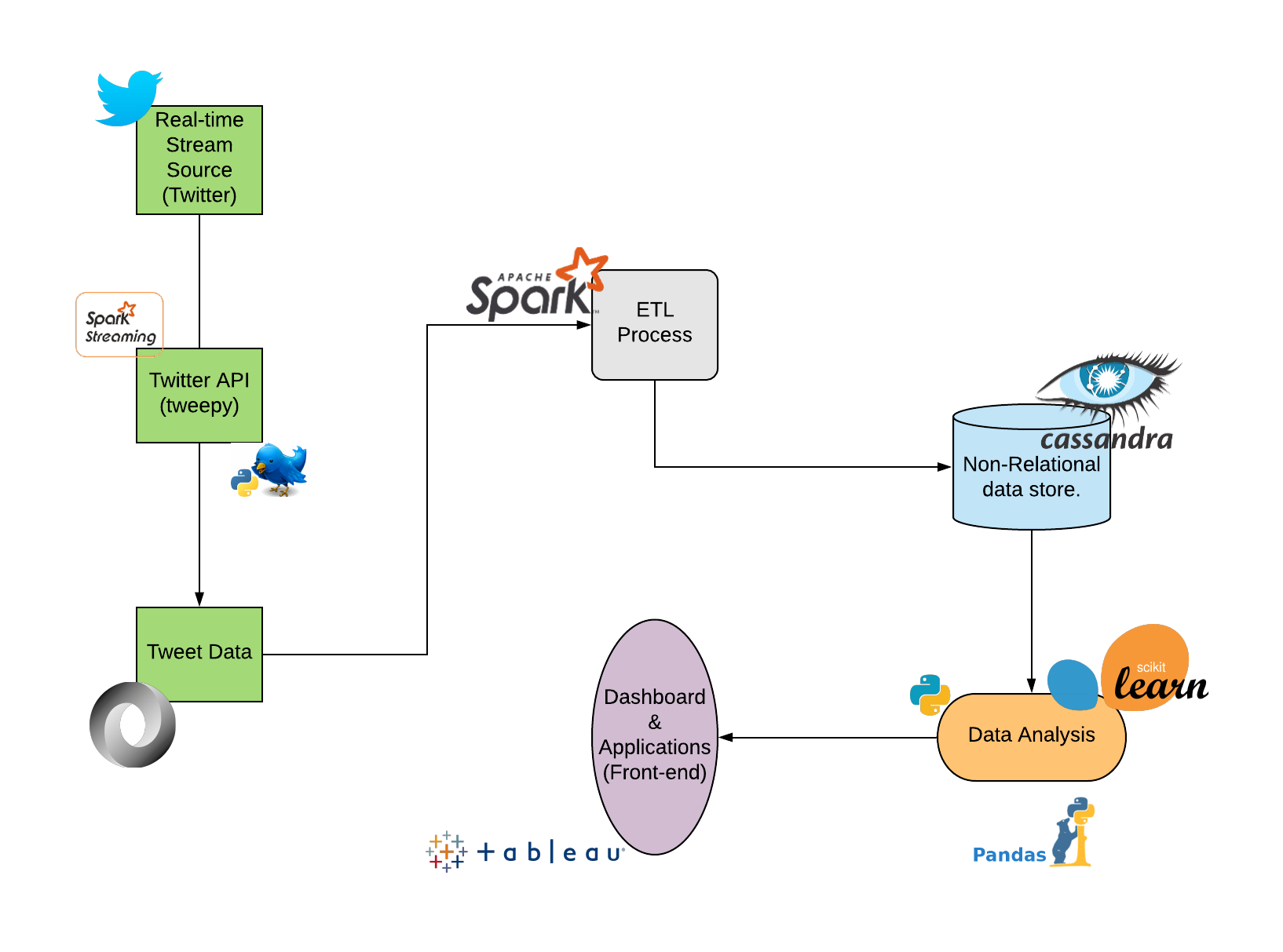
### Roles and Responsibilities

For this project, we will need the following personnel to be able to maintain the environment:

* Role: **DB administrators**
  + Responsibilities: capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recover on Cassandra DB.
* Role: **Network Administrator**
  + Responsibilities: installing and configuring computer networks and systems, identifying and solving any problems that arise with computer networks and systems, budgeting for equipment and assembly costs, assembling new systems, monitoring computer networks and systems to identify how performance can be improved
* Role: **System Administrator**
  + Responsibilities: user administration (setup and maintaining account), maintaining system, verify that peripherals are working properly, monitor system performance, create file systems, install software, create a backup and recovery policy, setup security policies for users, documentation, automation software such as puppet, chef, etc.
* Role: **Python developer**
  + Responsibilities: Write effective, scalable code, develop back-end components to improve responsiveness and overall performance, integrate user-facing elements into applications, test and debug programs, improve functionality of existing systems, implement security and data protection solutions, assess and prioritize feature requests, expertise in pandas and in the following API: tweepy.
* Role: **Data Analysts**
  + Responsibilities: Organize and query the data to provide useful Business intelligence insights. They will also be responsible for setting up the Tableau reports needed for the marketing team. The marketing team will in turn use these reports to make business decisions in regards to where and how to executive different marketing tactics to ensure growth of ticket sales.

### Performance Monitoring and Database Efficiency

#### Data Transfer Requirements

**

#### Data Formats

* Type of File: **JSON**

Using JSON over XML is better because it is lighter and also JSON data is formatted serially and contain no tags like XML, which makes it easier to read. Because it contains no tags it makes it easier and faster to parse also, and takes less character to represent data. Twitter data is naturally in JSON format, so it just works better.

Example of Data format:

**Endpoint snippet:**

{

"statuses": [

{

"created\_at":

"id":

"id\_str":

"text":

"truncated":

"entities": {

"hashtags": [],

"symbols": [],

"user\_mentions": [],

"protected":,

"followers\_count":

"friends\_count":,

"listed\_count":,

"created\_at": "

"favourites\_count":

"utc\_offset":

"time\_zone":

"geo\_enabled":

"verified":

"statuses\_count":

"lang":

* Type of Compression: **LZ4**

Compression maximizes the storage capacity of Cassandra nodes by reducing the volume of data on disk and disk I/O, particularly for read-dominated workloads. We will use LZ4 because is fastest to decompress.

**Appendix A: Acronyms**

**Table 1 - Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Literal Translation** |
| AP | Available and Partition Tolerant |
| API | Application Programming Interface |
| AWS | Amazon Web Services |
| AWS S3 | Amazon Simple Storage Service |
| CAP | Consistency, Availability and Partition tolerance |
| CDM | Conceptual Data Model |
| DB | Database |
| DBA | Database Administrator |
| DBMS | Database Management System |
| ERD | Entity Relationship Diagram |
| Geo | Geostationary Earth orbit |
| HDD | Hard Disk Drive |
| I/O | Input/Output |
| INT | Integer |
| LDM | Logical Data Model |
| LZ4 | Lossless Data Compression Algorithm |
| PDM | Physical Data Model |
| RAM | Random Access Memory |
| SSD | Solid State Disk |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| TXT | Text |
| XML | eXtensible Markup Language |

**Appendix B: Improvements**

**IMPROVEMENTS:**

In real time, to do an accurate sentimental analysis, we would require a column for retweets. These would have much more text and the semantics process would have more data to work with.

So the following column could have been added to the other column types:

CREATE TYPE retweets (

tweet\_id INT,

text TXT

);

Instead of VARCHAR we would use TXT as this parameter would contain much more text.