



## DATA WAREHOUSE AND BUSINESS INTELLIGENCE- PROJECT

Samridh Sharma

Student id : X17157846


email:x17157846@student.ncirl.ie


TOPIC: TEDx Data Analysis

## Introduction

TEDx is an international community which focuses to organise TED-style events with the main object of celebrating the local driven ideas and giving them a global audience. TED is owned by non-profit foundation with core objective of making great ideas accessible on a global platform. In the year 2017 when I was in my final year of college I experienced the golden opportunity to work as a volunteer for organising team of TEDx MITP and from that day I was intrigued by the methodology of organisation of TEDx event and their business approach. This is the main motivation for me to choose this topic for my Data Warehouse and Business Intelligence project. During my research in this topic I was amazed to read that in the year 2006 the director of TED Media pitched in the idea of TV show dedicated to the TED lectures which was rejected by several Networks, then some videos were posted on TED website and YouTube website. Surprisingly the viewership grew exponentially within one year. On April 2007, a new TED.com was launched which received critical, economic and social recognition till present day scenario. From being rejected by several media networks in the year 2006 to acceptance of a multimillion dollar deal by Netflix in the year 2012 it has been a beautiful success story in every area of concern where the community is also profited by getting exposed to new ideas and providing a global platform to share the ideas regarding every aspect of life and technology.

## Sources of Data:-

1. Kaggle is the source of the structured data of 2550 including the TEDMED, TEDWomen, TED ED, TEDFellows, TEDx, TEDGLOBAL, TEDYouth and TED Salon. As my scope was to focus on TEDx events because of my motivation so I concentrated on 471 TEDx Talks in the data. The dataset provided variables of given talk, such as the number of comments, number of languages translated, duration of the talk, number of tags, or day it was published online— are strong predictor of the popularity of the talk, measured in number of views. The dataset also includes variables like name, title, description and URL of each of the 471 talks, name and occupation of the speakers, duration of the talk, the TED event and date it was filmed at, date it was published online, number of comments, languages translated and views. We cannot take any of these observations as causal inference. The striking features which interested my approach for business intelligence and creating a Data Warehouse in the dataset was the number of TED views, TEDx event name, occupation of each of the speakers. This structured dataset helped me formulate the foundation for my approach for Data warehouse and Business Intelligence. 
2. As I mentioned before in the introduction that how YouTube website and relaunching of ted website changed the entire business scenario for TEDx events making it one of the most successful business model not only in terms of financial numbers but also in terms social good and community service as ted is owned by non-profit organisation. So instead of scraping data from twitter for analysing the sentiment of people for each TEDx talks in terms of sentiment score by using REST API. YouTube was used as the source on unstructured data which is YouTube views, YouTube likes and YouTube dislikes for each TEDx video. The reason for this shift of source was the limitation of REST API which only provides tweets for last 6-9 days. For more accurate analysis

scraped data from YouTube is more reliable as the TEDx videos are deployed on TED website as well as YouTube website. The YouTube data scraping for video related views, YouTube likes and YouTube dislikes was orchestrated by using python code and by deploying beautiful soup library to process the scraping. Hence YouTube act as a source of unstructured data for our Data warehouse and Business Intelligence. 

3. As TEDx events are independently organized events licensed by the organization in charge Ted conference. TEDx events have spread massively around the globe so that ideas can have global platform. It is imperative to orchestrate data regarding the location of various TEDx event as this data was not available in the structured data provided by Kaggle. But the structured data consist of a column of TEDx Events urls which was used as an input csv for scraping the locations from the TED website using r script by importing libraries like (htmltab), (rvest),(stringr) for dealing with regular expression and (XML2).The retrieval of all the locations of the TEDx Events we can make a lot of inferences regarding the demographic distribution of the locally organised TED events. **TED<sup>x</sup>**

### Architecture and Implementation

Both Inmon and Kimball[1]approach works perfectly fine to deliver Data warehouse. The Kimball approach starts with staging the answer to key business queries by data warehouse and uses bottom up approach which is creation of dimension tables which surrounds the fact table. On the other hand, Inmon approach is top-down approach which means the construction of data warehouse is first process and then the dimensions are created. Inmon approach advocates normalization of dimensions and Kimball approach advocates denormalization of dimensions[2]. Yet, the deciding factors for choosing one of the approach are reporting, urgency to complete project, future staff plan, frequency of changes in reporting requirement and the organization culture[2]. Kimball multidimensional style design executes a star schema in which all the dimensions are arranged around the fact table which in future provides possibility of simpler join and fast query processes to fetch the data using the OLAP technologies. As in our data warehouse the end user is involved in the process at very early stage which resolutely support Kimball approach which is also end user driven and hence we choose Kimball approach over Inmon's approach.

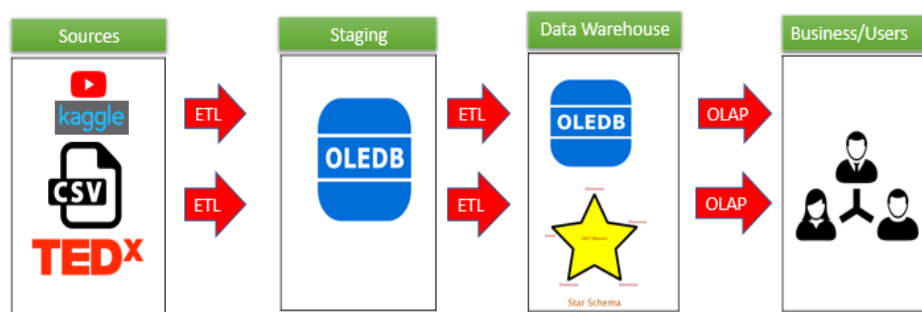


Figure: Overall Architecture of Data warehouse

The truncation and loading approach will be used to occupy the source data in staging area. The staging area will occupy the tables that are truncated each time the ETL process executes and the attributes are having 1:1 mapping with its source. The data is imported from Source Data to the Raw Data tables, perform the necessary cleansing and then feeding Dimensions table and Fact table the same pulled data.

The following steps are taken in consideration while orchestrating implementation:

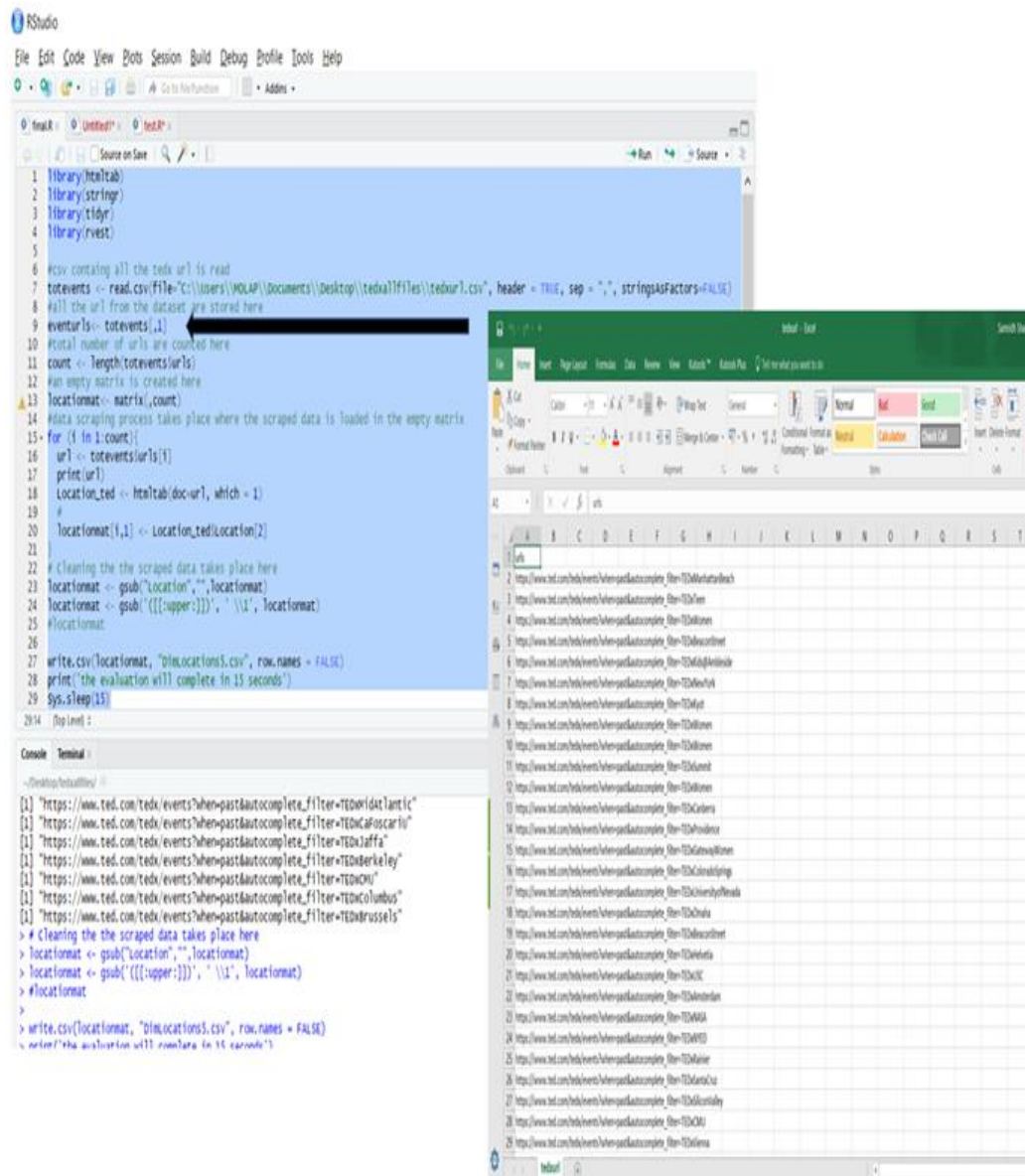
1. The first step while designing the implementation is to figure out a business objective on which we desire to run business intelligence.
2. Next step is to collect the data from variety of sources as in this case the three sources of data were Kaggle.com, TED.com, Youtube.com. The kaggle.com provided the structured data, TED.com provided the semi structured data and YouTube was the source of unstructured data.
3. Extraction of transactional data from the source into the staging area using truncate and load approach. The dataset from Kaggle formed blueprint for the business intelligence by providing the required measures like TED views and attributes like Speaker's Occupation, TEDxevent names etc. Various scrapping techniques were involved to scrap the locations of various TEDx events from the TED website and the relevant YouTube attributes from YouTube website.
4. Cleaning of extracted data is the most important step while orchestrating the implementation process. The data retrieved from the Kaggle website was structured cleaned data from which required attributes were selected which suits the approach and load it to the new csv files. The semi structured data retrieved from TED website which where the scraped locations was cleaned replacing the regular expression and then separating the city name from country name using libraries like stringr.
5. Now the entire scraped data from TED website and YouTube website is transported as flat files which now are loaded in the staging area to Raw data tables, measures and dimension table for processing.
6. Feeding the fact table using the dimensions with the help of relevant join Sequel query.
7. One of the most important step is deployment of cube using analysis services which is to be integrated with the reporting mechanism allowing the analysis of data by Business objective.
8. This is the last step is the business intelligence analysis where data is displayed and visualized using various tools like tableau, excel, power BI etc.

### Data model:

Before proceeding to dimension data model, we observe how the data is append from the source to the staging arena. The source which will act as the Mart data in this particular case is csv, TED scrapped locations, YouTube views, YouTube likes and YouTube dislikes which will be loaded to the measures of our data warehouse.

The sources of our data warehouse are as follows:-

1. The R scraped locations from the TED website which will be transported to a csv and that particular csv will be fed to the raw data table or stage table.



- The python code is used to scrape the YouTube likes, YouTube dislikes and YouTube views and these scraped data is loaded to the measures table.

```
>>> import re, requests
>>> from bs4 import BeautifulSoup
>>> import pandas as pd
>>>
>>> class YoutubeScrape(object):
...     def __init__(self, soup):
...         self.soup = soup
...         self.title = self.parse_string('.watch-title')
...         self.poster = self.parse_string('.yt-user-info')
...         self.views = self.parse_int('.watch-view-count')
...         self.published = self.parse_string('.watch-time-text')
...         self.published = re.sub(r'(\d+)', ' ', self.published).strip()
...         self.like = int(self.soup.find_all("button", class_="like-button-renderer-like-button-unclicked")[0].text.replace(', ', ''))
...         self.dislike = int(self.soup.find_all("button", class_="like-button-renderer-dislike-button-unclicked")[0].text.replace(', ', ''))
...     def parse_int(self, selector):
...         return int(re.sub('[^0-9]', '', self.parse_string(selector)))
...     def parse_string(self, selector):
...         return self.soup.select(selector)[0].get_text().strip()
>>> def scrape_html(html):
...     return YoutubeScrape(BeautifulSoup(html, "lxml"))
>>> def scrape_url(url):
...     html = requests.get(url).text
...     return scrape_html(html)
>>>
>>> urls = pd.read_csv('youtubeurl.csv', header=None)
>>>
>>> views_arr = []
>>> for i in a[0]:
...     temp = scrape_url(i)
...     print(str(temp.views) + ", " + str(temp.like) + ", " + str(temp.dislike))
...
4542265, 36608, 790
7688018, 83990, 1748
14113383, 104323, 7391
9119277, 102298, 2210
2259278, 24702, 312
9380479, 157490, 4061
4395839, 93195, 1295
7684298, 62237, 4109
2854620, 72885, 949
2111547, 29233, 709
5238872, 60499, 4673
3458330, 46879, 1852
573721, 6665, 102
951008, 26645, 230
903402, 7806, 296
4905649, 49941, 1714
```

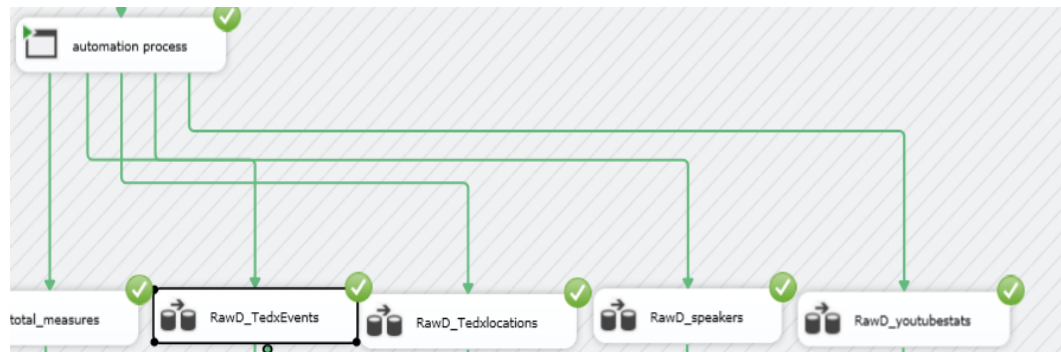
|    | A  | B | C | D | E | F |
|----|--|---|---|---|---|---|
| 1  | https://www.youtube.com/watch?v=4ZuIKF_VuA&t=30s |   |   |   |   |   |
| 2  | https://www.youtube.com/watch?v=iCvmsMjIf7o      |   |   |   |   |   |
| 3  | https://www.youtube.com/watch?v=KM4XeDlqDY       |   |   |   |   |   |
| 4  | https://www.youtube.com/watch?v=8K6uTCrVn        |   |   |   |   |   |
| 5  | https://www.youtube.com/watch?v=flJsdqnzB0       |   |   |   |   |   |
| 6  | https://www.youtube.com/watch?v=DfJi2hxf0        |   |   |   |   |   |
| 7  | https://www.youtube.com/watch?v=K1vskVdwl4       |   |   |   |   |   |
| 8  | https://www.youtube.com/watch?v=8K4l0C46VM       |   |   |   |   |   |
| 9  | https://www.youtube.com/watch?v=eBUcBfKVC0       |   |   |   |   |   |
| 10 | https://www.youtube.com/watch?v=XDmgOQSpLU       |   |   |   |   |   |
| 11 | https://www.youtube.com/watch?v=8u9TWUfnd0       |   |   |   |   |   |
| 12 | https://www.youtube.com/watch?v=IKH1awgKXWQ      |   |   |   |   |   |
| 13 | https://www.youtube.com/watch?v=5ahI2pajcho      |   |   |   |   |   |
| 14 | https://www.youtube.com/watch?v=2hlc2FL0dhl      |   |   |   |   |   |
| 15 | https://www.youtube.com/watch?v=XDXD3Nep5U4      |   |   |   |   |   |
| 16 | https://www.youtube.com/watch?v=jipe-LKn-4gM     |   |   |   |   |   |
| 17 | https://www.youtube.com/watch?v=fktsfcootG8      |   |   |   |   |   |
| 18 | https://www.youtube.com/watch?v=QJ0R05V013F      |   |   |   |   |   |
| 19 | https://www.youtube.com/watch?v=8u9TWUfnd0       |   |   |   |   |   |
| 20 | https://www.youtube.com/watch?v=yVXqyVdmY        |   |   |   |   |   |
| 21 | https://www.youtube.com/watch?v=I5wEYDaR-0       |   |   |   |   |   |
| 22 | https://www.youtube.com/watch?v=ueQqYebVhtc      |   |   |   |   |   |
| 23 | https://www.youtube.com/watch?v=yVXqyVdmY        |   |   |   |   |   |
| 24 | https://www.youtube.com/watch?v=dlqY8BD2VA       |   |   |   |   |   |
| 25 | https://www.youtube.com/watch?v=jdpKOLLYM        |   |   |   |   |   |
| 26 | https://www.youtube.com/watch?v=80X72630N        |   |   |   |   |   |
| 27 | https://www.youtube.com/watch?v=46w99823W_M      |   |   |   |   |   |
| 28 | https://www.youtube.com/watch?v=GcJalqT5nK       |   |   |   |   |   |
| 29 | https://www.youtube.com/watch?v=PiHMFWD3Y        |   |   |   |   |   |

- The csv files which are formed with the help of the structured, semi structured and unstructured data which are fed to the raw tables or staging tables through the automation process of ssis.

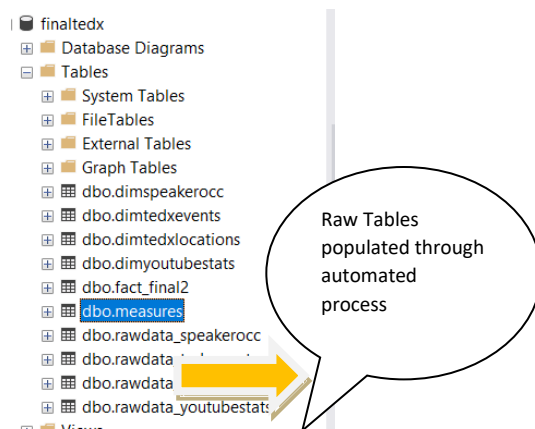
| Name              | Date modified    | Type                  | Size  |
|-------------------|------------------|-----------------------|-------|
| .Rhistry          | 24/04/2018 03:33 | RHISTORY File         | 7 KB  |
| Measures          | 26/04/2018 13:40 | Microsoft Excel Co... | 18 KB |
| Occupationmanbual | 26/04/2018 13:40 | Microsoft Excel Co... | 5 KB  |
| tedxEvents        | 26/04/2018 13:40 | Microsoft Excel Co... | 35 KB |
| tedxLocations     | 26/04/2018 13:40 | Microsoft Excel Co... | 5 KB  |
| try               | 26/04/2018 13:40 | Microsoft Excel Co... | 21 KB |



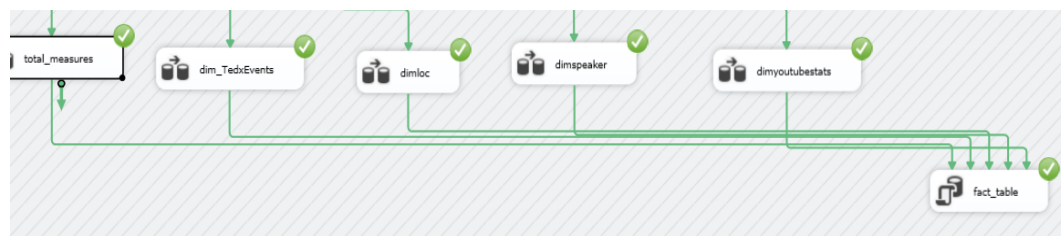
All these CSVs are dropped to the Raw Tables using the automation process by running the required R script in the execution process task and hence the tables are populated.



Below we have our source to raw or staging tables which is 1:1 in this scenario.



Proceeding to dimension data model where the dimensions are loaded and facts are generated. The dimensions and facts are formulated in accordance of our business queries. Below are 5 four dimension one fact table and one measures table.



1. dimtedxevents: This dimension comprises of TEDx event ID's, the description of the title of the TEDx talk and the list of all the nomenclature of TEDx events. This list of TEDx events are further connected to the locations of the events and measure by YouTube views and ted views. Hierarchy followed here: - EventTedx\_Name -> Tedx\_TitleName
2. dimtedxlocations: This dimension comprises of unique location ID's. The id of specific locations is glued with respective events and are situated in the fact table. The location

will help to analyse the regions with more number to TEDx events held and regions where significantly less number of TEDx events are held which further deduce which region of the globe needs to be framed for the future events. Hierarchy followed here:  
- country -> city

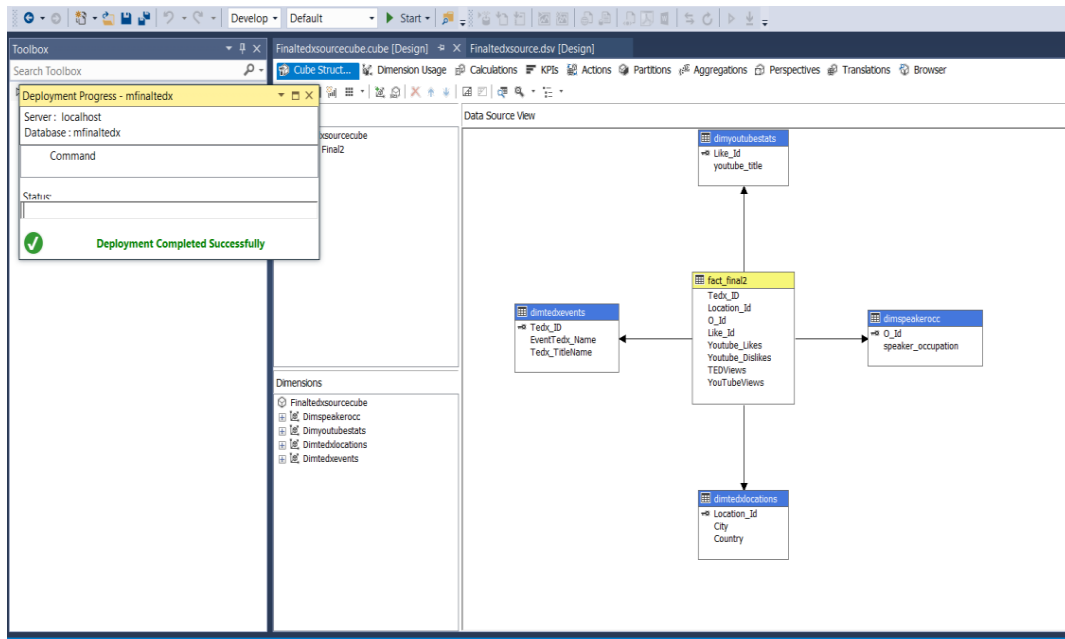
3. **dimspeakerocc**: This dimension consists of speaker's unique occupation id and the occupation of speaker. As we know a TED speaker come from a wide diversity of occupation background and we have tried to retrieve the analysis of the preference of occupation background of the speaker according to the YouTube likes and dislikes which further infer which speaker from which occupation background should be invited to deliver a TEDx talk so that viewership of the talks increases.
4. **dimyoutubestats**: This dimension consists of a like\_id and the YouTube title of each TEDx video. This dimension contributes to assist platform analysis and comparison between TED website viewership and YouTube website viewership as it provides command TEDx talk for both the platform. Hence analysis can be made to infer which platform is better for which kind of video to be deployed.
5. **Fact\_final2**: the fact table consists of foreign keys such as Tedx\_ID, Location\_Id, O\_Id, Like\_Id and measures Youtube\_likes, Youtube\_dislikes, TEDViews, YouTubeViews etc.

finaltedx  
Database Diagrams  
Tables  
System Tables  
FileTables  
External Tables  
Graph Tables  
dbo.dimspeakerocc  
dbo.dimtedxevents  
dbo.dimtedxlocations  
dbo.dimyoutubestats  
dbo.fact\_final2  
dbo.measures  
dbo.rawdata\_speakerocc  
dbo.rawdata\_tedxevents  
dbo.rawdata\_tedxlocations  
dbo.rawdata\_youtubestats  
Views

GO

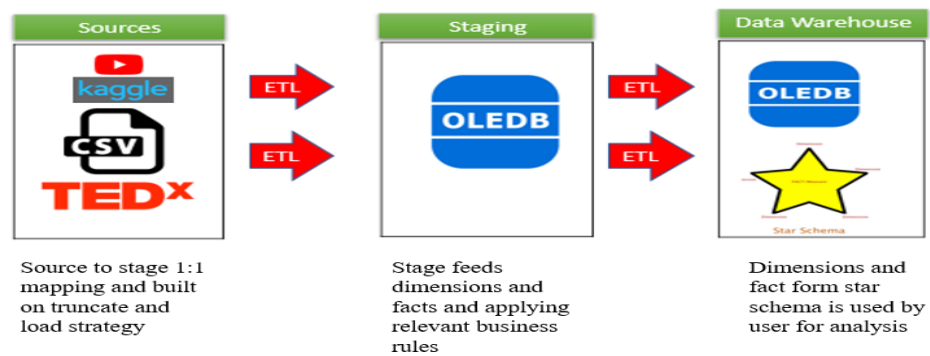
```
CREATE TABLE [dbo].[fact_final2](
    [Tedx_ID] [varchar](50) NULL,
    [Location_Id] [varchar](50) NULL,
    [O_Id] [varchar](50) NULL,
    [Like_Id] [varchar](50) NULL,
    [Youtube_Likes] [varchar](255) NULL,
    [Youtube_Dislikes] [varchar](255) NULL,
    [TEDViews] [varchar](255) NULL,
    [YouTubeViews] [varchar](255) NULL
) ON [PRIMARY]
GO
```



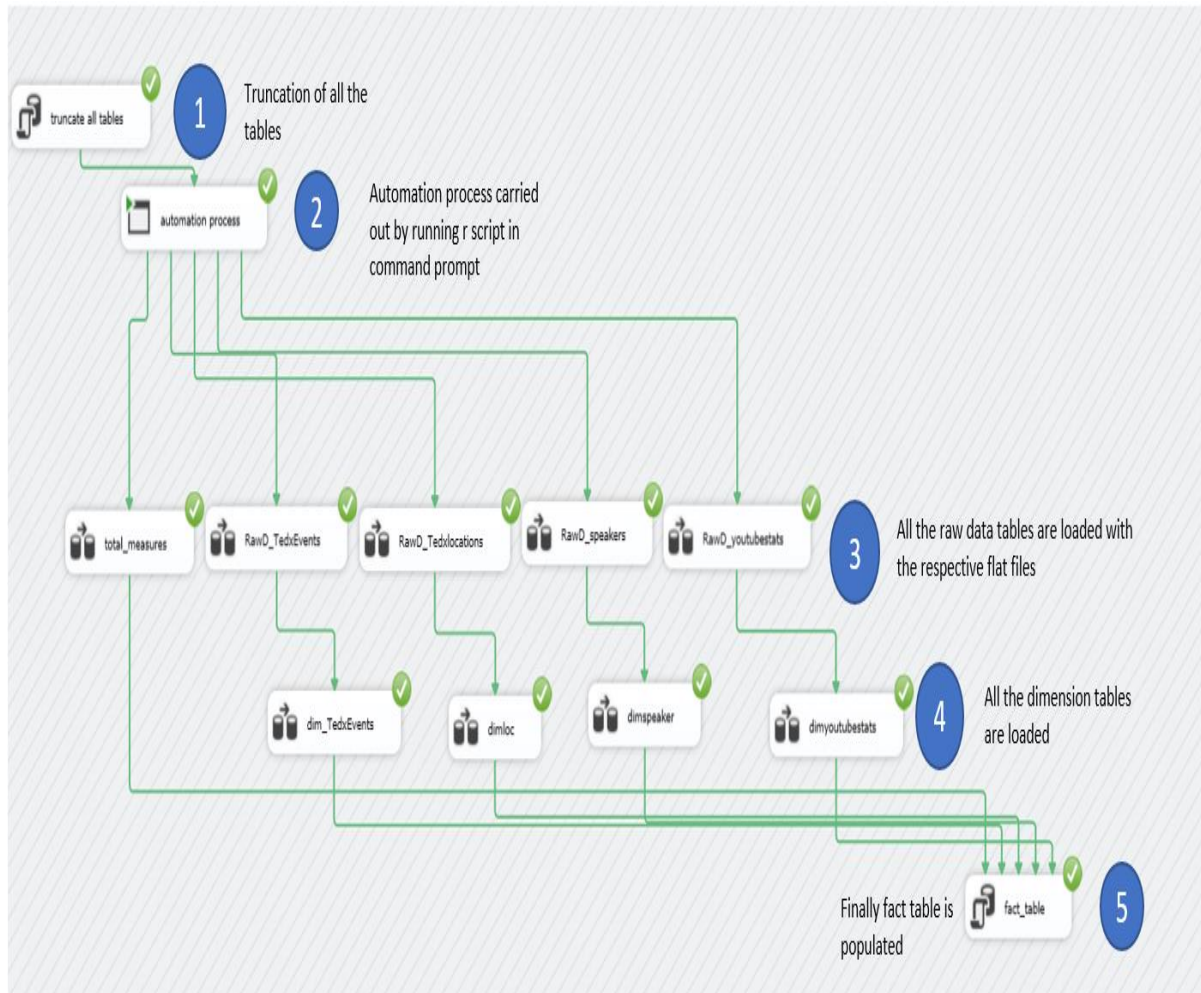


## Extraction,Transformation and Load:

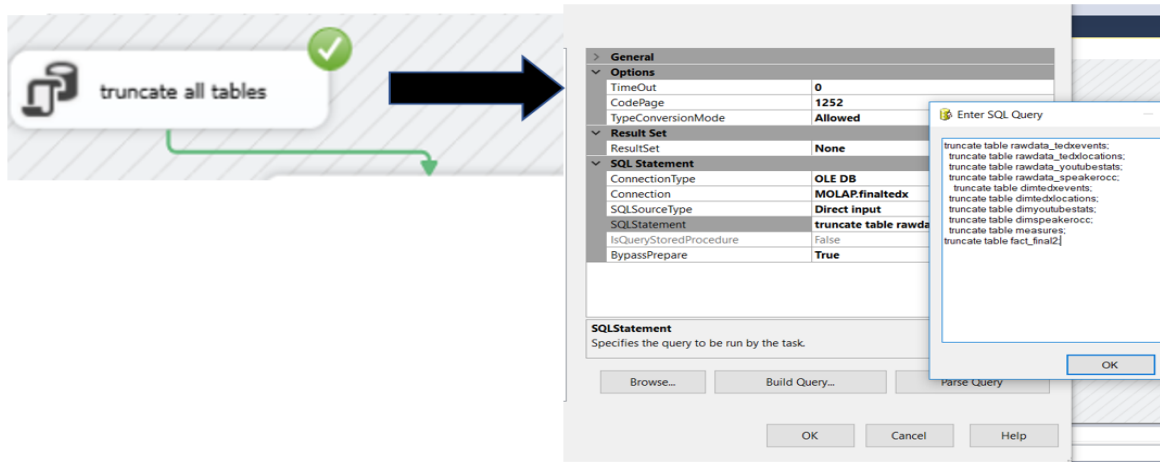
Logical data flow of ETL:



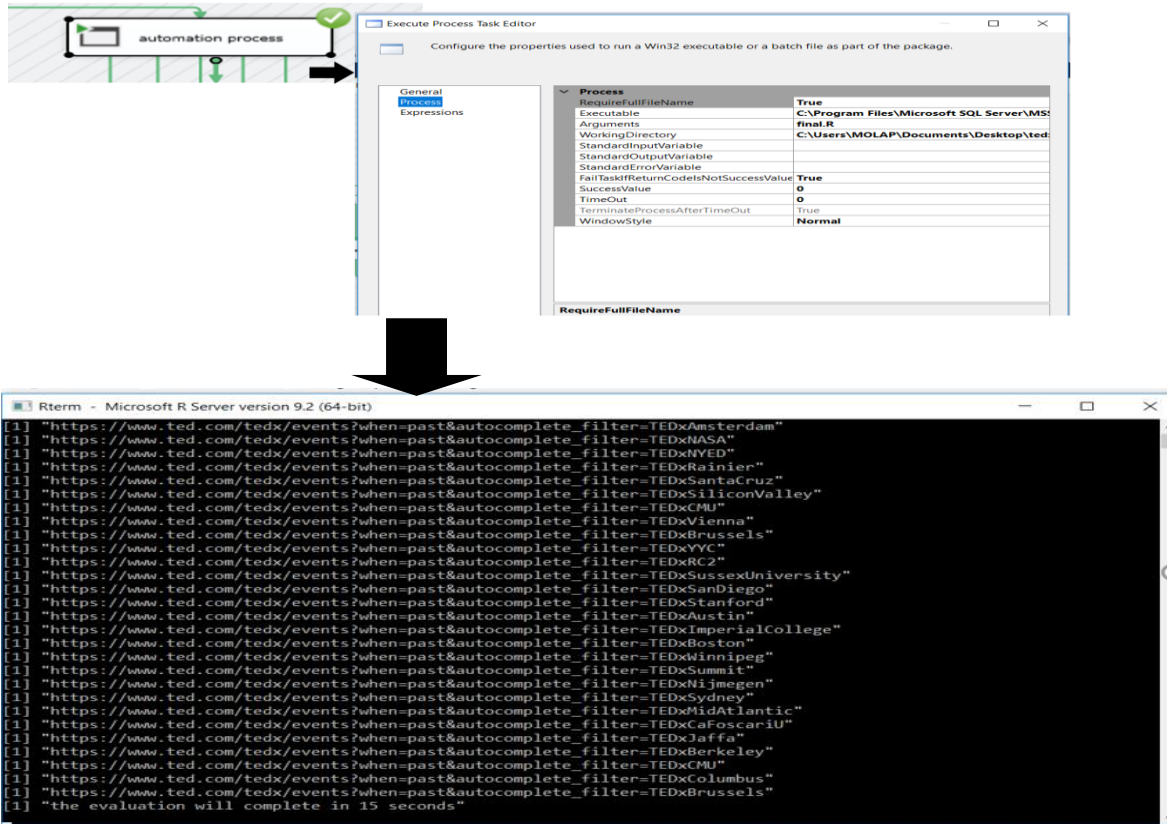
ETL process automated in SSIS:



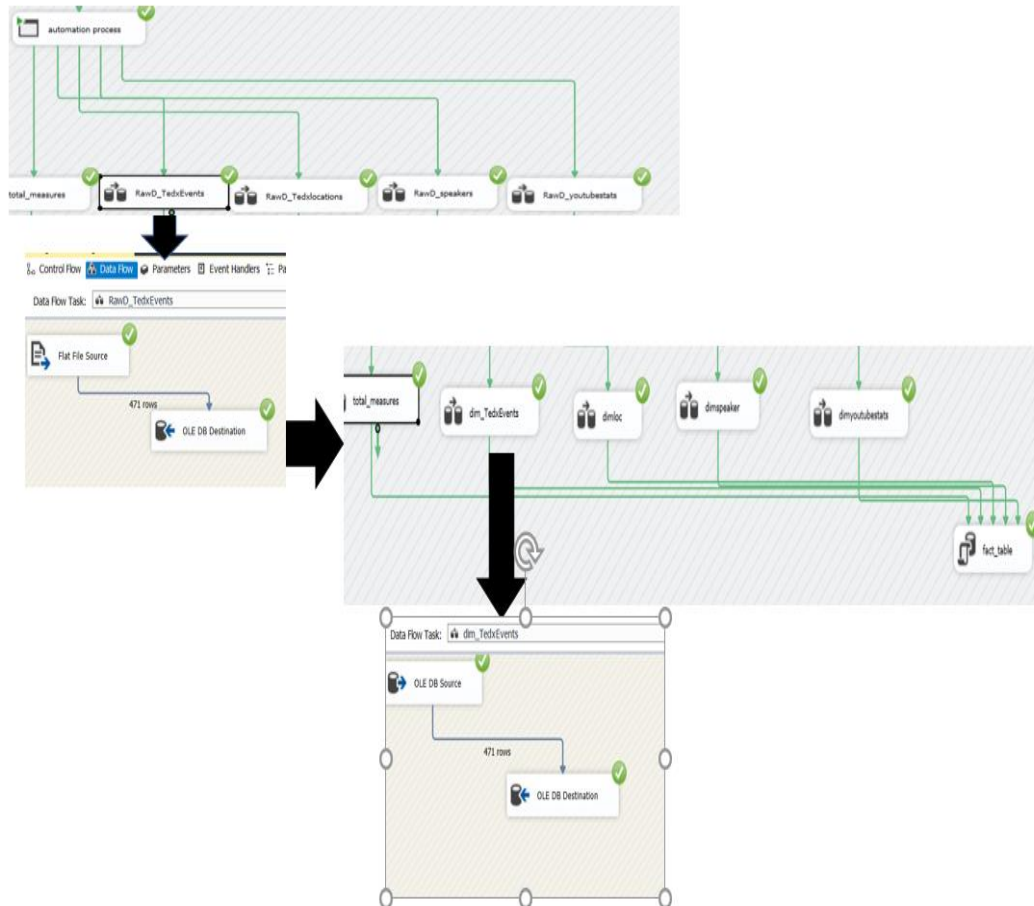
1. Truncate all tables: It is the first process in which all the table which were generated are truncated each time the ETL process is run. The tables gets devoid of its values each time ETL executes.



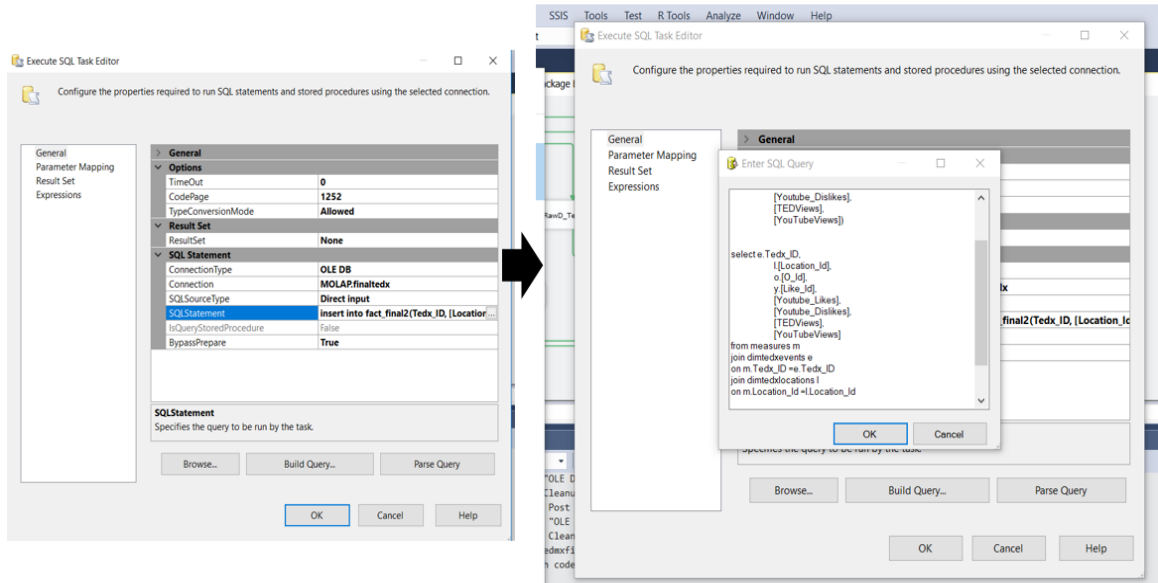
- The R script runs in the the execute process task which carries the entire automation of the entire ETL process which scraps the data from the source and then load the csv files at the required destination. On successful execution of R script, csv files with raw data are generated and then saved at the designated working directory on the hard disk.



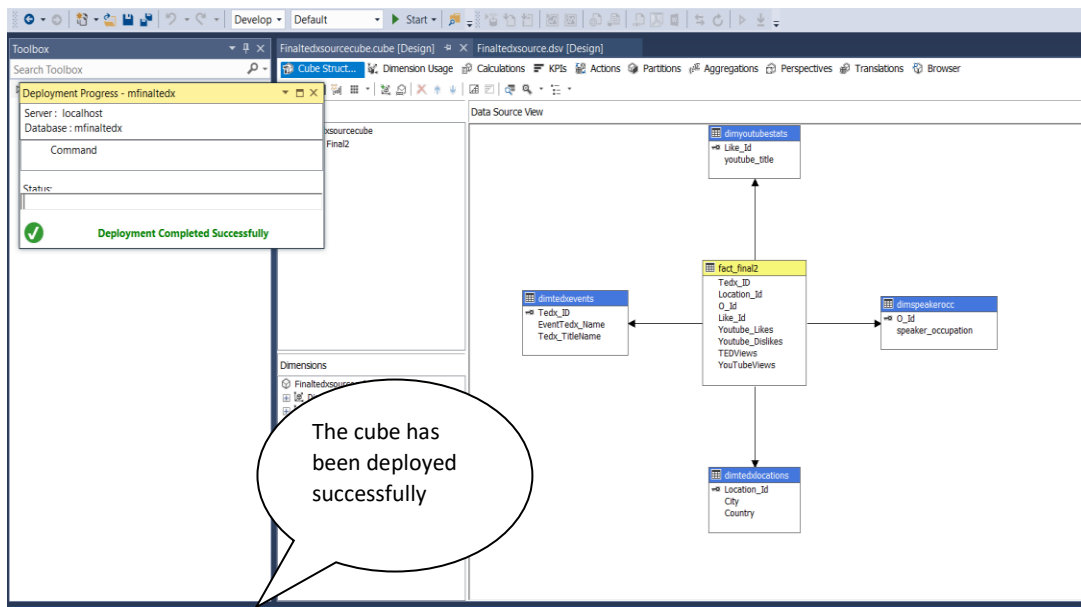
- The successful execution of r script leads to the parallel processing of further processes in SSIS. The raw tables are populated after the data is scrapped by using r programming and respective flat files formed from structured and the unstructured data are loaded. Hence the RawD\_tedxevents, RawD\_tedxlocations, RawD\_youtubestats, RawD\_speakers and measure are loaded with the respected flat files which are constructed in structured format by the help the R script which takes its source from kaggle.com for structured data, semi struted data from TED.com and unstructured data from youtube.com.



4. After raw tables are loaded the next task is proceeded in which dimension tables are loaded with the final data using OLEDB source and OLEDB destination. By segregating the raw tables and dimension tables we orchestrate a system in which if any modification or changes in the data needs to be implimented we can do it at staging level with the raw tables and hence the dimension tables get updated parallely we donot need to involve in manual modification.
  
5. On successful loading of dimension tables and measures a join query is imperative to execute the loading of fact table for further processing of the cube.



- After the fact table is successfully loaded then the Sequel Server Analysis Services comes to the process where it generate cube based on dimensions tables and the fact table and creates a star schema with dimension tables surrounding the Fact table.

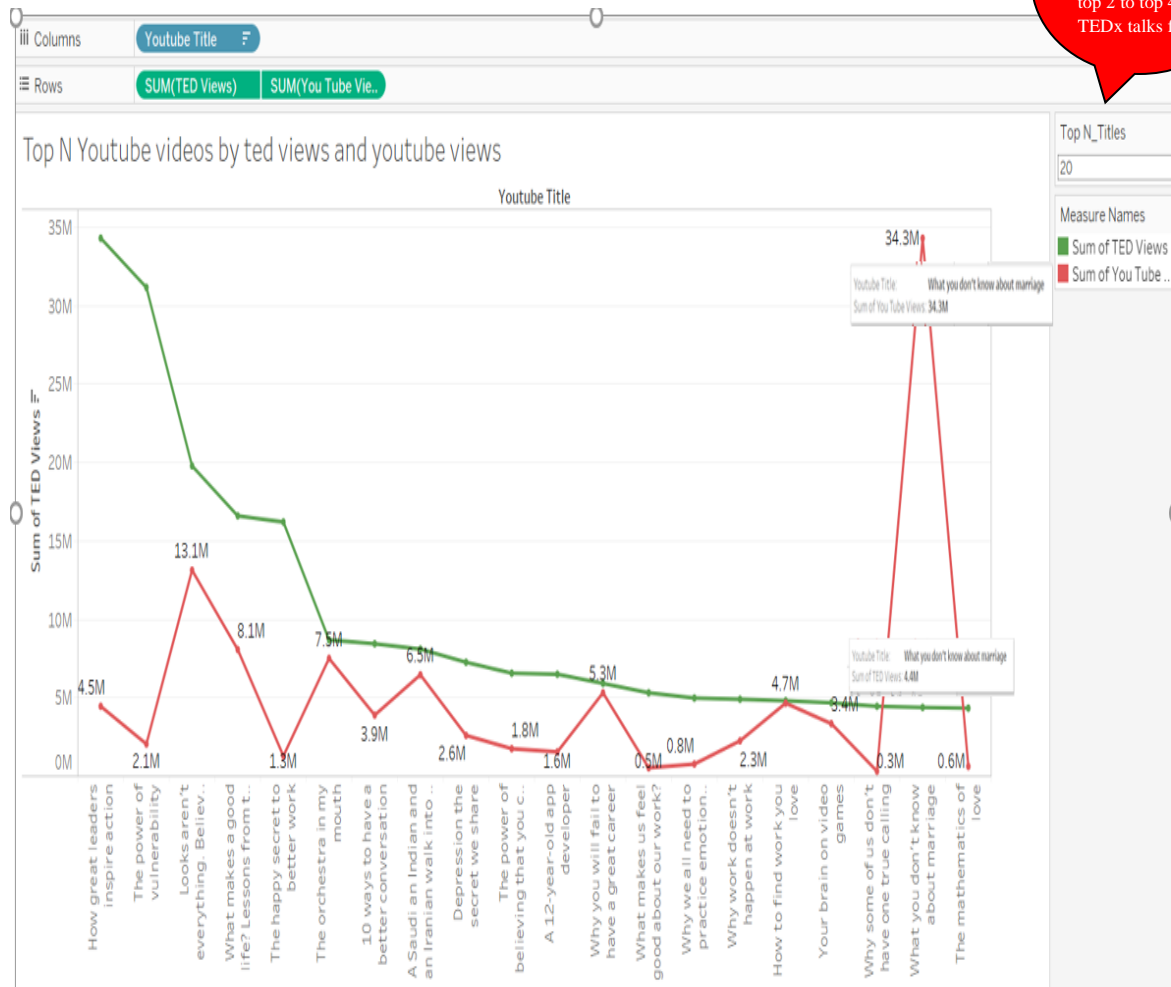


- After the successful deployment and processing of the cube , the cube is loaded to tableau application for the visualisation of business intelligence on dimensions and fact tables.

## Application of data warehouse and business intelligence

**Case 1:** Youtube views vs TED views for analysing better platform based on number of respective views for each TEDx talk.

The visualization is made scalable top 2 to top 471 TEDx talks for

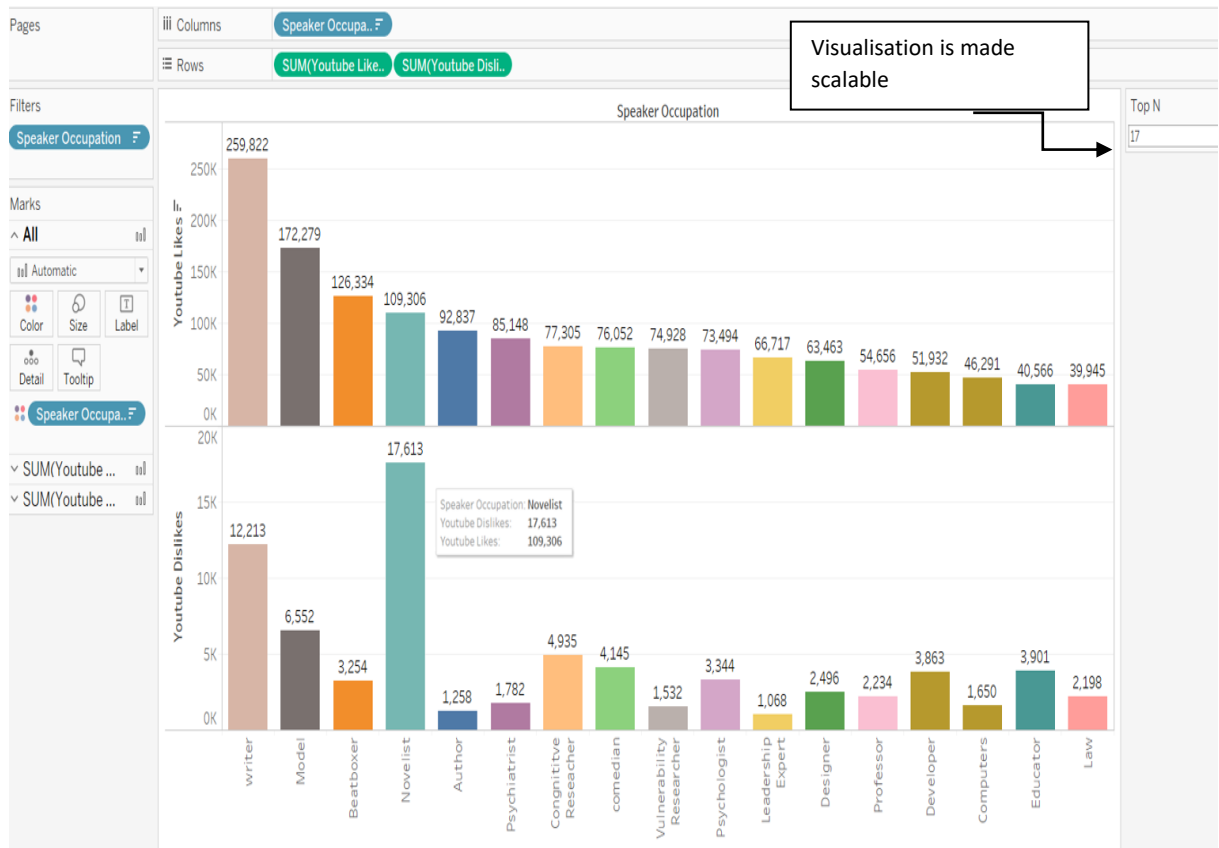


- As the TEDx videos are deployed on both the platform youtube website and TED website. Both the website has a loyal viewership. Hence an analysis could be set where which platform cater more views and audience on the bases of there views recorded respectively.
- Now if we observe the visualization above we can notice that the TED views per video is more if we have a general outlook towards the entire visual description. The difference between the viewership between the TED website and youtube website per video is significantly low.
- Also we can notice that the video which have comparatively very high viewership on TED website has significantly very low viewership on youtube platform. Alternatively the videos which have a very high viewership from youtube website has very low viewership on TED website.



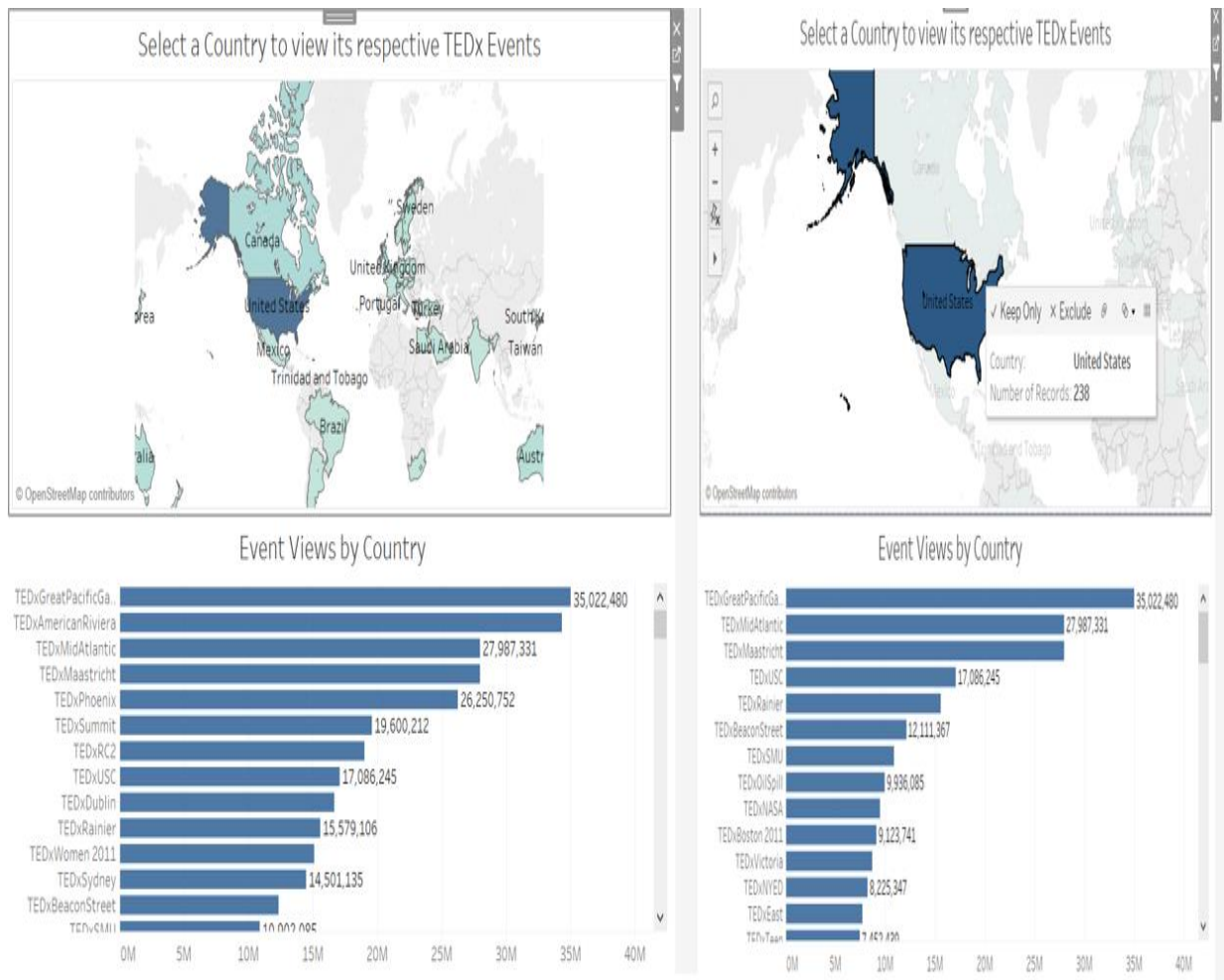
- As mentioned in the visualization that the visualization is designed to be scaleable as it can show analysis of 471 TEDx videos but for space constrain sake we displayed top 20 videos with there views on TED website and youtube but if we have to be binary and infer a comparitively better platform for deployment of videos then TED website caters to comparatively more viewership than Youtube website according to respective number of views.

**Case 2:** The prefered speaker's occupation according to the youtube likes and youtube dislikes



- Each TEDx talk is delivered by speakers who come from diversity of occupational background irrespective of social,economic or poltital strata. Hence each video is deployed on youtube where the viewer has option to like or dislike the video. Hence an approach is pitched where we analyse that speakers from which occupation background fetches comman interest of people because majority of time the TEDx talk delivered by the speakers is somewhere related to the occupation or the occupational experience of the speakers.
- Hence by running this business intelligence query we can concentrate on making room for more speakers coming from prefered background according to the viewers like and hence this will increase in gathering more views for TEDx talks and videos.
- In this visualization we can observe that writer,model,author,Psychiatrist are prefered more by the viewers as these occupation background collected comparatively high number of youtube likes.

### Case 3: Countries where more number of TEDx events are organised around the Globe



- The TEDx events are hosted throughout the globe every year. TEDx events are locally organised in more than 150 countries. Hence an analysis will be insightful to find the countries which actively host TEDx events.
- The visualization depicts the countries which are actively involved in hosting more number of TEDx events throughout the year. The darker the colour the more number of TEDx events take place in that region of the world. In this display the total number of TEDx events held in United States are 238 and a graph of all the tedx events held in United States along with its online viewership is displayed.
- Hence from this analysis we can infer that in which countries respond positively towards locally organised TED event and which are the countries where the frequency of organised TEDx event is low ,hence calls for an appalling need to deploy more locally organised TEDxEvents .

**References:**

- [1] Kimball, R (1996). Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouse, New York: Wiley & Sons.
- [2] S.Rangarajan and V. &rarr;, “Data Warehouse Design – Inmon versus Kimball”, TDAN.com, 2017.[Online]. Available: <http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>. [Accessed: 8-Dec-2017].

## APPENDIX:

Code for scrapping my data from YouTube which is Youtube likes, Youtube dislikes and Youtube views.

```
>>> class YoutubeScrape(object):
...     def __init__(self, soup):
...         self.soup = soup
...         self.title = self.parse_string('.watch-title')
...         self.poster = self.parse_string('.yt-user-info')
...         self.views = self.parse_int('.watch-view-count')
...         self.published = self.parse_string('.watch-time-text')
...         self.published = re.sub(r'(Published|Uploaded) on', '', self.published).strip()
...         self.like = int(self.soup.find_all("button", class_="like-button-renderer-like-button-unclicked")[0].text.replace(',', ''))
...         self.dislike = int(self.soup.find_all("button", class_="like-button-renderer-dislike-button-unclicked")[0].text.replace(',', ''))
...     def parse_int(self, selector):
...         return int(re.sub('[^0-9]', '', self.parse_string(selector)))
...     def parse_string(self, selector):
...         return self.soup.select(selector)[0].get_text().strip()
...
>>> def scrape_html(html):
...     return YoutubeScrape(BeautifulSoup(html, "lxml"))
...
>>> def scrape_url(url):
...     html = requests.get(url).text
...     return scrape_html(html)
...
>>> urls = pd.read_csv('youtubeurl.csv', header=None)
>>>
>>> views_arr = []
>>> for i in a[0]:
...     temp = scrape_url(i)
...     print(str(temp.views) + ", " + str(temp.like) + ", " + str(temp.dislike))
...
4542265, 36608, 790
7688018, 83990, 1748
14113383, 194323, 7391
9119277, 102298, 2210
2259278, 24702, 312
9380479, 157490, 4061
4395839, 93195, 1295
7684298, 62237, 4109
2854620, 72885, 949
2111547, 29233, 709
5238872, 60499, 4673
3458330, 46879, 1852
573721, 6665, 102
951008, 26645, 230
903402, 7806, 296
4905649, 49941, 1714
```

Code for scrapping locations from TED website of various TEDX EVENTS

```
final.R x  Untitled1* x  test.R* x
Source on Save  Run  Source
1 library(htmltab)
2 library(stringr)
3 library(tidyr)
4 library(rvest)
5
6 #csv containing all the tedx url is read
7 totevents <- read.csv(file="c:\\Users\\MOLAP\\Documents\\Desktop\\tedxallfiles\\tedxurl.csv", header = TRUE, sep = ",", stringsAsFactors=FALSE)
8 #all the url from the dataset are stored here
9 eventurls<- totevents[,1]
10 #total number of urls are counted here
11 count <- length(totevents$urls)
12 #an empty matrix is created here
13 locationmat<- matrix(,count)
14 #data scraping process takes place where the scraped data is loaded in the empty matrix
15 for (i in 1:count){
16   url <- totevents$urls[i]
17   print(url)
18   Location_ted <- htmltab(doc=url, which = 1)
19   #
20   locationmat[i,1] <- Location_ted$Location[2]
21 }
22 # cleaning the the scraped data takes place here
23 locationmat <- gsub("Location","",locationmat)
24 locationmat <- gsub('([[:upper:]])', ' \\1', locationmat)
25 #locationmat
26
27 write.csv(locationmat, "DimLocations5.csv", row.names = FALSE)
28 print('the evaluation will complete in 15 seconds')
29 Sys.sleep(15)
```

29:14 (Top Level) R Scrip

```
Console  Terminal x
~/Desktop/tedxallfiles/
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxMidAtlantic"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxCaFoscariU"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxJaffa"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxBerkeley"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxCMU"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxColumbus"
[1] "https://www.ted.com/tedx/events?when=past&autocomplete_filter=TEDxBrussels"
> # Cleaning the the scraped data takes place here
> locationmat <- gsub("Location","",locationmat)
> locationmat <- gsub('([[:upper:]])', ' \\1', locationmat)
> #locationmat
>
> write.csv(locationmat, "DimLocations5.csv", row.names = FALSE)
> print('the evaluation will complete in 15 seconds')
```