

# **IDS 521: Big Data Solution for Enterprise-scale Social-Media Analytics**

Microsoft Azure SQL Data Warehouse

Vijitha Vasantha Kumar

Nithyadharshni Sampathkumar

## **Introduction:**

Azure has more than 50 cloud services. The integrated tools, pre-built templates and managed services make it easier to build and manage enterprise, mobile, Web and Internet of Things (IoT) apps faster, using skills and technologies we already know. Microsoft is a Leader for Cloud Infrastructure as a Service, Application Platform as a Service, and Cloud Storage Services for the second consecutive year.

## **Data Sources:**

Azure SQL Data warehouse processes enormous amount of relational and non-relational data. It supports different data sources-

### **SQL database**

Relational databases.

### **Azure tables**

They have a flexible schema, compared to the RDBMS tables. The data in these tables are stored in key-value pair formats and hence can be accessed easily. The Azure storage table can also store telemetry data in aggregated format.

### **Azure Data Lake**

Stores unlimited data and provides simplified data management and governance. It comprises of the azure data factory, stream analytics and event hubs.

We started by creating a logical SQL server by logging into the Azure Portal.

On the trial version they provide a list of Regions in which they advise us to create the logical SQL server. Once we create the server we share the Azure Subscription ID, Region and the Logical SQL Server Name after which they send a confirmation email about the validation period.

Our initial approach was to migrate our existing Twitter database used in assignments into the data warehouse – since it can sit directly atop a relational database. There were 2 proposed solutions to go about this -

**Data Platform Studio** offered by red-gate which establishes a gateway connection between the local database and Azure SQL

**Azure Data Factory** a data movement service which is used to ingest data from source through a data management gateway

Since they both involved installing separate applications we then tried to load the flat files in the Azure Blob storage and migrate them into the data warehouse. We found that the Blob storage best supports .bacpac files (Database backup files).

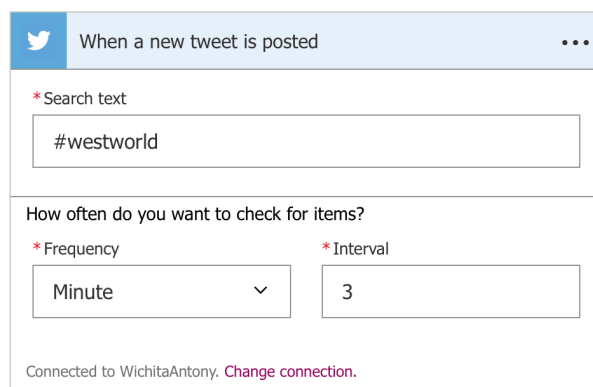
We then started thinking about whether Azure supports a twitter API and found the Logic App has partnered with multiple social media applications including Twitter. We developed a small logic app to see its function-

Collection of data from Social Media –

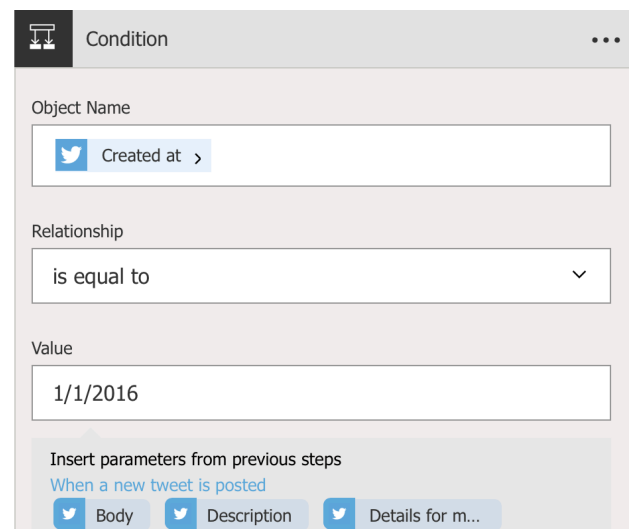
### Logic App + Partner APIs

In the assignment problem, we had data from multiple sources, Twitter and Earnings Release data. For data from Twitter which should run real-time and scrape tweets from Twitter based on certain conditions we felt the following Logic App can help

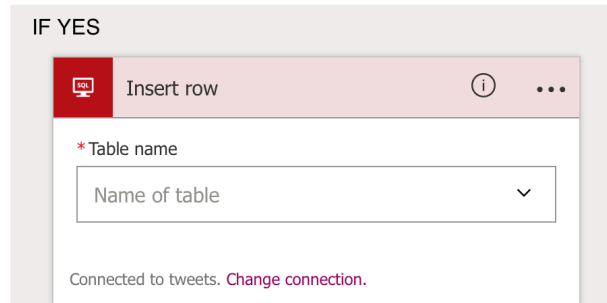
- Set up Trigger for multiple Tweet scenarios and additionally specify conditions it must satisfy
- To test we retrieved data from Twitter every time there was a new tweet with the hashtag #westworld
- Follow the trigger with an action or condition. The condition could be a rule to filter the Tweets.



The screenshot shows the 'When a new tweet is posted' trigger configuration in the Logic App designer. It includes a search text field with the value '#westworld', a frequency dropdown set to 'Minute', and an interval field set to '3'. At the bottom, it indicates the connection is 'Connected to WichitaAntony' with a 'Change connection' link.



The screenshot shows the 'Condition' configuration panel. It has three main sections: 'Object Name' with a dropdown set to 'Created at', 'Relationship' with a dropdown set to 'is equal to', and 'Value' with a text field containing '1/1/2016'. At the bottom, there is a section titled 'Insert parameters from previous steps' with three buttons: 'Body', 'Description', and 'Details for m...'. The 'Body' button is currently selected.



**Figure 1-** Azure Logic app with Twitter API

In our case, tweets with the symbol tags #AAPL or #CSCO can be extracted using the logic app. We can then perform dynamic SQL queries on the extracted data to obtain summary statistics such as the twitter peaks.

Usage of logic app is billed on a volume-based tiered model. So, we are charged depending on the number of actions we perform in the logic app.



**Figure 2-** Usage fee on the Logic app

Here is a small snippet where we retrieved tweets whenever there was a new tweet with a hash tag #westworld.

```
{
  "body": {
    "TweetText": "Ohhhhhh that is so sweet!!!!!!!!!!!!!! #Westwo",
    "TweetId": "804694700489932800",
    "CreatedAt": "Fri Dec 02 14:32:21 +0000 2016",
    "RetweetCount": 0,
    "TweetedBy": "pangiannak",
    "MediaUrls": [],
    "TweetLanguageCode": "en",
    "TweetInReplyToUserId": "",
    "Favorited": false,
    "UserMentions": [],
    "OriginalTweet": null,
    "UserDetails": {
      "FullName": "Panagiotis Giann",
      "Location": "Krypton",
      "Id": 2569279207,
      "UserName": "pangiannak",
      "FollowersCount": 511,
      "Description": "Oh sweetheart... You don't break into my ho",
      "StatusesCount": 14177,
      "FriendsCount": 515,
      "FavouritesCount": 137,
      "ProfileImageUrl": "https://pbs.twimg.com/profile_images/74"
    }
  },
}
```

**Figure 3-** Output of a single collected tweet

## Connecting to the SQL Data Warehouse and Querying the application:

Since migrating the existing database from mysql to the Azure data warehouse was tedious, we created a SQL data warehouse with the sample data warehouse provided by Azure – AdventureWorks DW.



*Figure 4- Quick access options in the SQL data warehouse*

Establishing a connection with the Azure SQL data warehouse in the SQL Server can be achieved through various means. Some among them are:

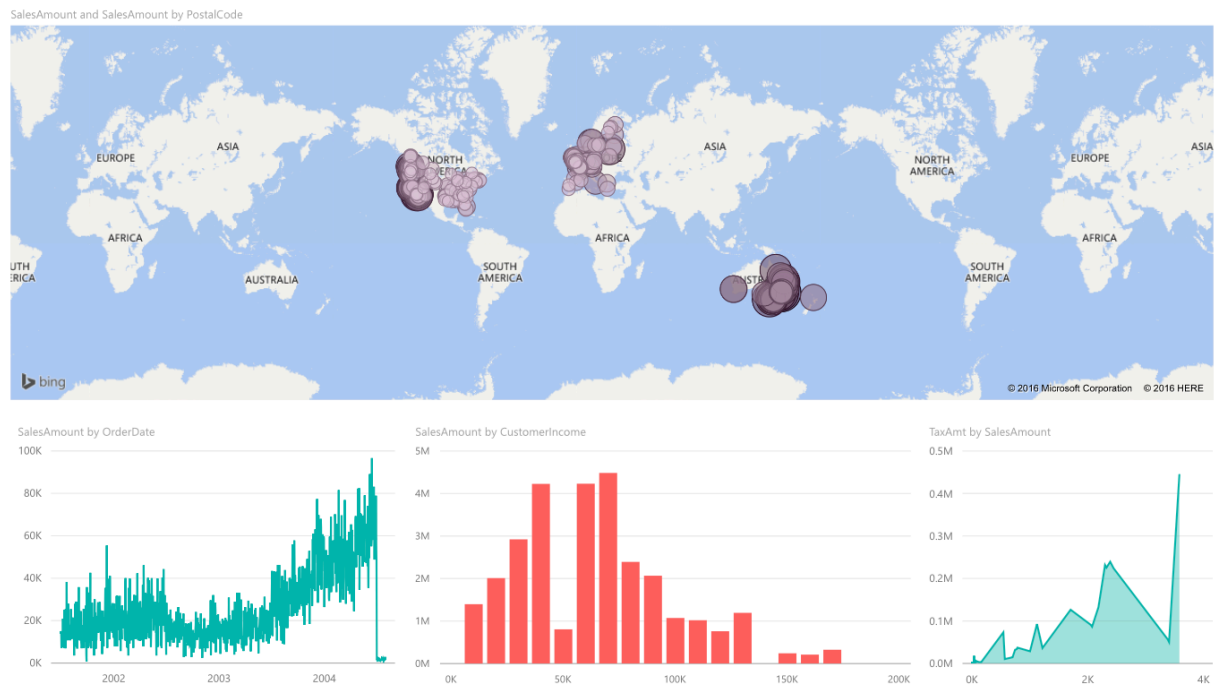
- The **sqlcmd** in command prompt followed by the server credentials authentication
- **Microsoft SQL Server Management Studio**
- **Microsoft Visual Studio** which we used in our project to connect our local database to the Azure data-warehouse. Primarily, we had to authorize a set of IPs that can access the data warehouse. Adding the logical SQL server through the SQL Server Object Explorer of Visual Studio prompts us for the server credentials. On successful authentication, we will be able to write dynamic SQL commands to build the application.

## Visualization / Business Intelligence

The primary tools recommended for BI are

- Power BI
- Microsoft Excel using
  - PowerMap
  - PowerQuery

Below are some of the analytics performed on the AdventureWorks DW in PowerBI. They were simple drag and drop on the aggregated table columns of AdventureWorks DW from simple online tutorials. The BI part of it was perhaps the easiest to learn and use.

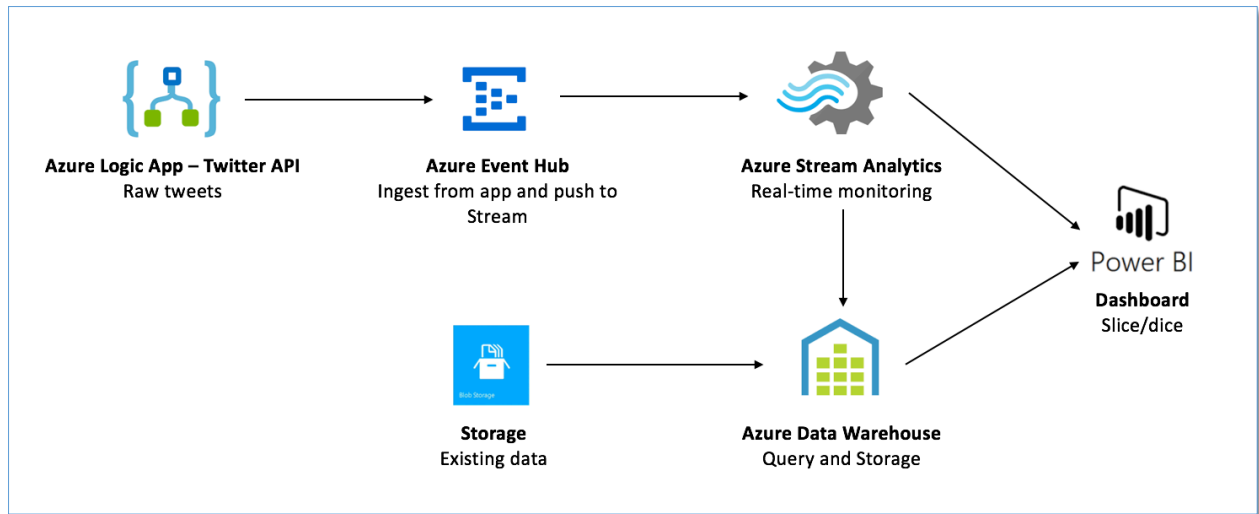


**Figure 5-** Power BI report on the aggregated table of AdventureWorks DW

Also, tools from popular BI partners including Looker Data Sciences, Tableau Software, and Qlik Technologies are recommended and supposedly easy to use with SQL Data Warehouse.

## Design Specification

We found that Azure has services that we can use to implement the enterprise scale application end to end. Below is a design specification using few of the features of Azure we learnt over the course of this project. Azure Stream Analytics studies information flow from Twitter and feedback from it can be used in scaling up/down decisions and security.



**Figure 6-** Architecture using Azure services and products for end to end application

The Storage portion can be used for information that does not come in from the Social Media Partner API. This can be the release earnings part of our assignment. Flat files can be stored on the Blob containers and read in using d-sql in Visual Studio to integrate into our data warehouse.

## Advantages

While working with Azure we found the following features alluring when compared to RDBMS and other warehousing applications in the market:

- **True cloud elasticity for freedom with control**

It allows experimentation with different data warehousing units to explore the best fit for our application. We can start with the lowest DWUs (Data Warehousing Units) while building the application. We can also play with scaling up/down during busy/lull periods. During weekends, assuming there won't be much chatter about companies, we can bring down the DWU units.



**Figure 7-** Scalability slider

### Decoupled storage and compute model

Since the storage and compute is decoupled we pay only for the compute we use and the storage we use. Unlike other applications we are not forced to over-pay or over-provision.

- **Geo-Replication**

Potential to configure up to four readable secondary databases in the same or different regions. Whenever the primary database is not available, the secondary database serves as the source for querying and data usage. This was a default option even on our trial version. Since we were advised to restrict our usage within a few regions, we created our replication server in the same region as our SQL server. It however gives us the option to create a replication server anywhere in the world.



*Figure 8- Usage fee on geo-replication*

- **Innovative Security**

Has a built in audit and threat protection by employing machine learning to warn ahead of irregular patterns in usage. Provides SSO capability with Azure Active Directory letting end users with the right privilege to be able to run analytics against the data warehouse.

- **High reliability and uptime**

Faces no downtime issues even when scaling is performed

- **Support and Service level agreements for every service offered**

- **Ease of Use**

- **Extensive support documentation**

## **Disadvantages**

- Supports software best in the Microsoft Suite and partner products. Indirect and cumbersome methods to interface with other applications. While there is a way to interface with other vendor applications and open source products, it involves purchasing another solution and multiple data transformations. Using the Microsoft suite for end to end implementation is at first trial a smoother experience.
- DSQL does not support BLOB varchar data types and the fix for this is to use exec to run the code in chunks



- While we were extremely guarded in our use of the free credit of \$200, we quickly ran out of the free credits while experimenting with multiple features and missed to actively disable or pause the applications.

While there is a steep learning curve, once you get the hold of using Azure and the keywords, the extensive support documentation and blogs out there enable implementation of the application in the market easy and user-friendly.

### **Bibliography and Citations**

- Azure SQL Data Warehouse Documentation - <https://docs.microsoft.com/en-us/azure/sql-data-warehouse/>
- <http://www.jenunderwood.com/2015/10/12/real-time-analytics-with-azure-and-power-bi/>
- <http://www.10thmagnitude.com/tech-blog/the-seven-coolest-features-of-azure-sql-database-and-sql-data-warehouse/>
- <https://www.simple-talk.com/cloud/cloud-data/using-ssis-to-load-data-into-azure-sql-data-warehouse/>