Class: DAV6100

Instructor: Brandon Chiazza

Name: Natan, Randy, Wen

THE NYC REAL ESTATE DATA WAREHOUSE

Contents

[Proposal 1](#_Toc27605824)

[Background 1](#_Toc27605825)

[Scope Statement 2](#_Toc27605826)

[The Acceptable results 3](#_Toc27605827)

[Develop and Design Project 3](#_Toc27605828)

[Source Data Introduction 3](#_Toc27605829)

[Data Dictionary 4](#_Toc27605830)

[NYC Data 4](#_Toc27605831)

[Zillow Data 5](#_Toc27605832)

[City Realty data 6](#_Toc27605833)

[Data Infrastructure Conceptual Architecture 7](#_Toc27605834)

[STAGING Data Modeling 7](#_Toc27605835)

[OLAP Date Modeling 7](#_Toc27605836)

[S3 bucket 9](#_Toc27605837)

[ETL Instruction 9](#_Toc27605838)

[The Problem Encountered 10](#_Toc27605839)

[Conclusion 10](#_Toc27605840)

[Appendix 11](#_Toc27605841)

# 

# Proposal

## Background

The Real estate market in New York City has been growing year-over-year. Many good things are going on in the NYC housing market which shows that the price would keep going upward. The good geographic location, the growing population, and the boost of the economy make New York City regularly rank among the expansive real estate markets in the work. To the real estate investors, New York City is always one of the best places for long term real estate investments in the U.S. However, there are always risks come with the opportunities. As a new investor to the NYC real estate market, our client wants to earn the profit from this market while limiting the risks as low as possible. The client’s investment comes up with several ideas about how to earn profit from the market. For example, invest in short-time rentals, such as Airbnb; invest in medium and long-term rents. Also, the investing team can’t decide whether the company should focus on resident rentals or commercial rentals. Moreover, there is another way to earn profit other than invest in rentals, such us, Condominium and Cooperation sales, and brokerage.

According to the client, there are lots of opportunities, but lack of information prevents the company from maximum profit in this market. Thus, the client gets touched with us, DataCollector. The client reviewed their situation with us and would like us to build a data warehouse that could support the company’s investment to make decisions on the investment.

# Scope Statement

After a careful review of our client’s situation and intensive research of the NYC real estate market. We come up with the agreement with our client, as we mentioned in the agreement, we will do the following works for our client:

1. Collect information that could help our client make the investment decision from the public data source. The information includes building address, a building type (rent apartment, Condominium or Cooperation for sales), building rental or retail price, related tax information, Apartment types (1-bedroom, 2-bedrooms), etc.
2. Build an STAGING database using the information collected. The STAGING database should keep collecting the information from the public data source or the source related to the NYC real estate market.
3. Build an OLAP database used to analyze the information, so that the information collected could support the client in the investment decision-making. Design a schema (star schema or snowflake schema) that would enhance the data analysis and query performance.
4. Build the backup database to make sure the investment team would run queries and working on data analysis when the main database malfunctioned.
5. Analyze the collected data in the OLAP database, visualize the data based on the analysis.
6. Be prepared to use the results from analysis and visualization to answer the client’s questions.
7. The questions from the client will include but not limited to:
   * Which neighborhood has the highest sales price of (Con-do, Co-ops) which neighborhood has the lowest sales price?
   * Which neighborhood’s selling price increased most?
   * What is the average selling price of the 1-bed room in Manhattan borough?
   * What is the average rental price of the 2-bed room in Queens borough?
   * What type of apartment (studio, 1bed room, and so on) has the highest transaction amount in the past 45 days?
   * What type of apartment has the highest transaction in the dollar amount?

# The Acceptable results

At the end of the project, the client should have:

1. A STAGING database that contains the information supports the investment team in decision-making.
2. An OLAP database that would enhance the data analysis and query performance.
3. Two backup databases, one for STAGING database, and the other for OLAP database, that enable the investment to run queries when the main databases malfunctioned.
4. A dashboard that would contain the visualized analytic information of the OLAP database and be able to answer the questions related to the NYC real-estate market that clients would ask.

# Develop and Design Project

## Source Data Introduction

The source data are mainly collected from the internet. There are three different databases are used in as the source data and they are: NYC Open Data - Rolling Sales Data (NYC Open data), StreetEasy (Zillow.com or Zillow) sales data, and City Realty Market Data. The collection method is difference for the different database due to the availability. For example, the NYC Open Data provide the API key, PDF and MS excel (xlsx) files. The data collector could use all the ways to collect the data. In this project, the data collectors use the API key since the xlsx and PDF files will occupy more space in local machine when compare with API key. Zillow as a professional real estate search engine, contains structured data as Comma delimited file (csv) files and ready to use, therefore, the data collectors decide to use the csv file provide by Zillow. The data need to be collected from City Realty Market Data is also structured and stored on its website. However, City Realty neither provide the API key to allow the users to access data nor provide the csv file for the collectors to download. Thus, the data collector might need to develop a data scape program to gather the data form its website.

## Data Dictionary

### NYC Data

The NYC Open Data – Rolling Sales Data contains the Department of Finances’(DOF) Rolling Sales files lists properties that sold in the last twelve-month period as of Dec 8, 2019 in New York City for all tax classes. The files include: the neighborhood, building type, square footage, and other data. The Glossary and Excel File use information and NYC Building Class Code Description (Web link) can be found in the Appendix. The sample data dictionary is following:

**Borough:**

The Name of the borough in which the property is located,

**Neighborhood:**

DOF assessors determine the neighborhood name in course of valuing properties. The common name of the neighborhood is generally the same as the name Finance designates. However, there may slight differences in neighborhood boundary lines and some sub-neighborhoods may not be included.

**Block:**

A Tax Block is a sub-division of the borough on which real properties are located.

The Department of Finance uses a Borough-Block-Lot classification to label all real property in the City. “Whereas” addresses describe the street location of a property, the block and lot distinguish one unit of real property from another, such as the different condominiums in a single building. Also, block and lots are not subject to name changes based on which side of the parcel the building puts its entrance on.

**Lot:**

A Tax Lot is a subdivision of a Tax Block and represents the property unique location.

**Address:**

The street address of the property as listed on the Sales File. Coop sales include the apartment number in the address field.

**Zip Code**:

The property’s postal code

**Residential Units:**

The number of residential units at the listed property.

**Commercial Units:**

The number of commercial units at the listed property.

**Total Units:**

The total number of units at the listed property.

**Land Square Feet:**

The land area of the property listed in square feet.

**Gross Square Feet:**

The total area of all the floors of a building as measured from the exterior surfaces of the

outside walls of the building, including the land area and space within any building or

structure on the property.

**Sales Price:**

Price paid for the property.

**Sale Date:**

Date the property sold.

**$0 Sales Price:**

A $0 sale indicates that there was a transfer of ownership without a cash consideration. There

can be several reasons for a $0 sale including transfers of ownership from parents to children.

Zillow Data

As one of the lending organization in real-estate market, Zillow provides data on sold homes,

including median sale price for various housing types, sales counts, and foreclosures provided as a share of all sales in which the home was previously foreclosed upon. here is also current and historical for-sale listings data, ranging from median list prices and inventory counts to share of listings with a price cut, median price cut size, age of inventory, and the days a listing spent on Zillow before the sale was final. Inventory and other housing data also are available for local markets. The data dictionary is as follow:

Home types and housing stock

* All Homes: Zillow defines all homes as single-family, condominium and co-operative homes with a county record. Unless specified, all series cover this segment of the housing stock.
* Condo/Co-op: Condominium and co-operative homes.
* Multifamily 2+ units: Units in buildings with 5 or more housing units, that are not a condominiums or co-ops.

Types of Zillow home value index

All Homes (SFR, Condo, Co-op) ($): Median estimated home value for all homes of these types within a region.

Other metrics

* Median List Price ($): Median of the list price (or asking price) for homes listed on Zillow.
* Median Sale Price ($): Median of the selling price for all homes sold in a given region.
* Median List Price Per Sq Ft ($): Median of list prices divided by the square footage of a home.
* Listings with Price Cut (%): The percentage of current for-sale listings on Zillow with a price cut during the month.

Dictionary in the data used

**SizeRank:**

The index to record how many rows of data recorded.

**Region ID:**

Last five digits in the zip code

**Region Name:**

First five digits in the zip code

**Region Type:**

Zip code

**State Name:**

Abbreviation of the State’s name in the U.S.

**Date:**

Year and month

City Realty data

The data from City Realty are focus on the real-estate market in New York City. The building types include Condo, Coop, Condop, and Townhouse. City Realty contain a large amount of the data; however, the website is not data collector friend. There is not API or any type of ready-to-download data. The data dictionary is as follow:

Address:

The address of the listed sales includes street name, and apartment number.

Neighborhood:

Same as neighborhood in Rolling sales data

Beds:

Shows how many bedrooms in the apartment sold.

Size:

The size of the apartment sold in squared foot.

Price/Squared ft

The unit price of the apartment sold.

Price

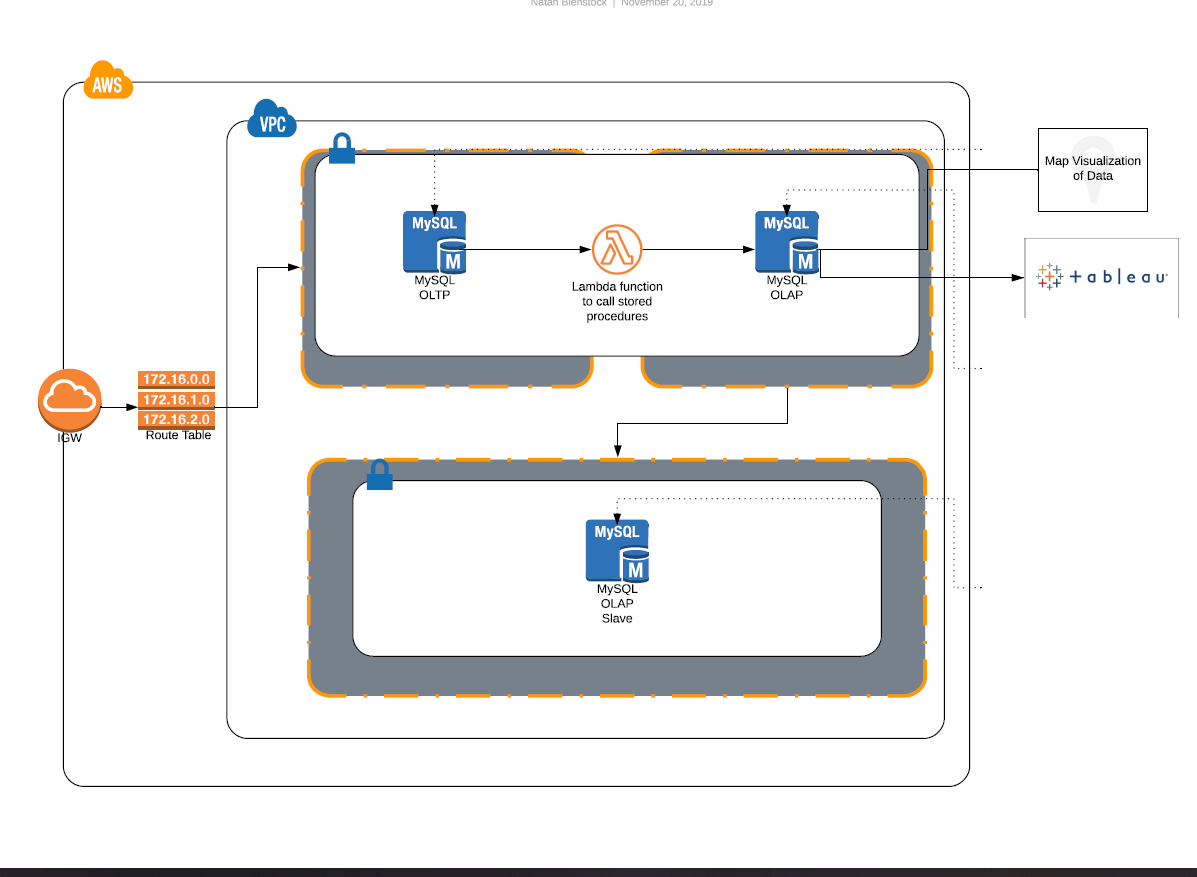
The total price of the apartment sold.

Date

The month and date that the apartment sold.

Data Infrastructure Conceptual Architecture

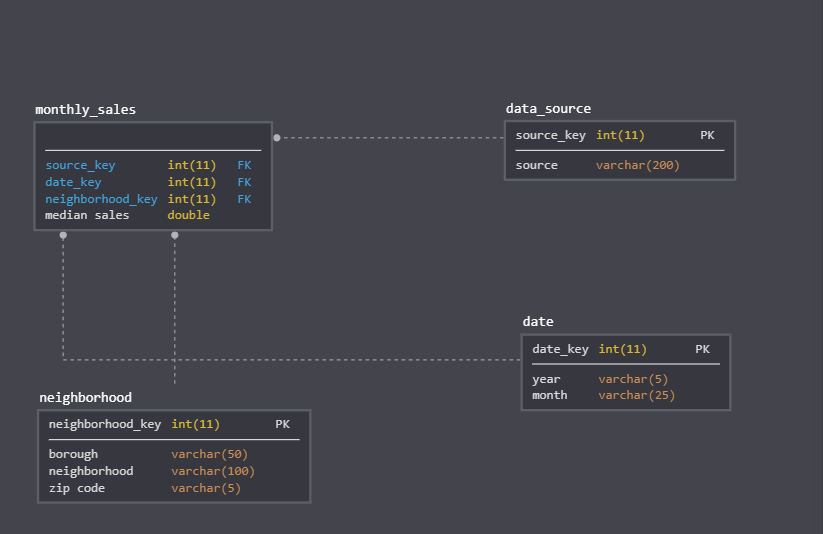
The Conceptual Architecture is showing below in the picture. Based on the study of the client’s requirement, the NYC real estate market, and budgeted cost of the project. DataCollector decides to use AWS as the foundation of the data warehouse. MySQL workbench is used to construct two databases, ETL procedure, and query the databases. The STAGING and OLAP database will be built in the same VPC and same security group. The backup database is built in the same VPC but different security groups. Thus, the client can still access the data when the Master OLAP database is unavailable. The data from the STAGING is transferred to the OLAP through several designed lambda functions and stored procedures. The OLAP database is also connected to the tableau for data visualization. Moreover, the client gets the information needed by the query OLAP database.

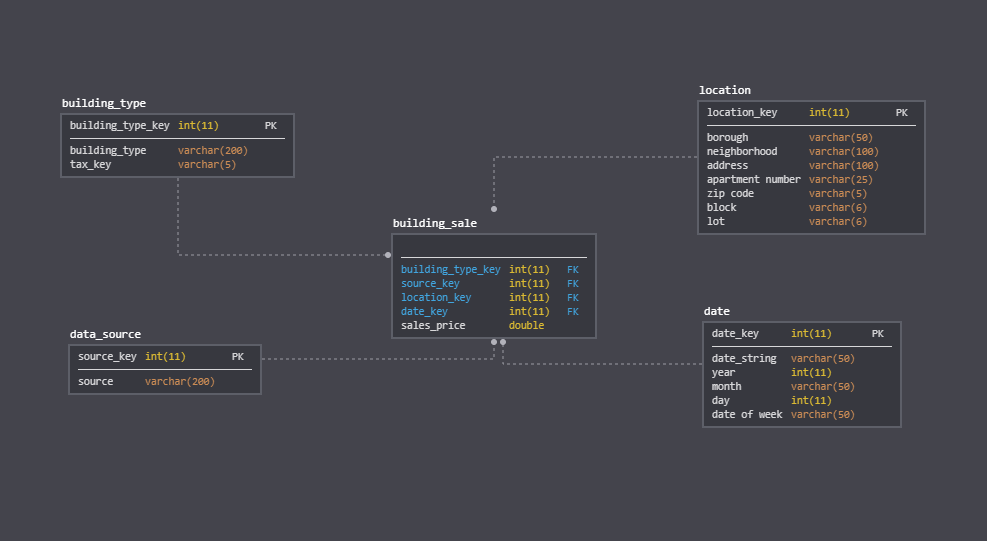


STAGING Data Modeling

Since the data are from three different sources and developers are willing to build the OLAP data modeling when transfer data from OLAP to OLAP. The developers decide to load all the data from the source and divide it into three different tables by their original source. However, not all the data from the source are loaded in the STAGING database. Developers will filler the data that are useful to the client. The data that not related to the project will not be loaded into STAGING.

OLAP Date Modeling

The developers decide to build two facts table and related dimension tables to reflect the data on a monthly and daily base. In the monthly base, the facts table is “monthly\_sales” table, it contains the “source\_key”, “date\_key”, “nieghorhood\_key”, and “median sales”. All the facts except “median sales” are foreign keys that related to the primary key in the three-dimension table. 

The daily based facts table include “building\_type\_key”, “source\_key”, “location\_key”, “date\_key”, and “sales\_price”. Excepts for “sales\_price”, other facts are foreign keys related to the primary key. The data model is shown below.

### S3 bucket

All the related documents, logs, instructions and files will be loaded into S3 bucket.

## ETL Instruction

The ETL procedures are applied at two stages. The first stage is to convert the data from the data source to the STAGING database. The data will be extracted from the data source to the local machine. The date irrelated to the project will be dropped. For example, the housing median price data related to the area other than NYC could be dropped. Also, the price per square foot data in City Realty could be dropped. After the data are processed, they will be load into the STAGING database and label by the original data source they have been extracted.

The second stage where the ETL procedures applied is between the STAGING and OLAP. In this stage, the schemas will be built first in the OLAP database. Then, the data in the STAGING will be extracted, processed, joined, and loaded into the OLAP database. The replica will be built after the OLAP construction is finished.

# The Problem Encountered

There are many problems, issues, and conflicts happened during the project. Here are the problems that are critical to the success of the project and maybe become the inspiration for the future project.

First of all, there is one problem encountered in the data collecting procedure and it is how the developer gathers the data from the City Realty database since the website doesn’t provide any tools that would be helpful for data collecting. The developer decides to develop a web scraper to gather the data from the website directly. However, the data site is large. There are about 50 rows per page and have more than 6,700 pages. Also, there are cells without any values. The web scraper cannot recognize and skip these blank. This made the scraped data very bad to process. Also, the complex HTML structure made the Python Request library hard to scape. Therefore, the developers decide to use the Python request library to scrape the data. The developers inspect the data structure of the website and write a python script to by pass the headers, automatically turn website pages and scape data.

Furthermore, the developer encounters the problem that disallows the MySQL workbench to connect the AWS database. The reason for this problem is that the in-bound rule isn’t set up correctly for the AWS VPC security group. Also, the developer is struggling about how to balance accessibility and security. The client wants to access the database from anywhere, however, due to the cost limitation, the developer can't set up the “all-traffic” at no cost. Therefore, the developer decides to set up in-bound for several secured major IPs and teach the client to add and delete in-bound and out-bound rules. So, the client can always to it by him- or her-self to access the database from anywhere.

Moreover, another problem the developer's face is to upload the documents to the S3 bucket automatically. The developers wrote the code in Python does fulfill the mission. However, due to cost limitations. The developers can only use the free tier. This made the automatic procedure fail to upload documents since the free tier will change the new credentials document every time the developer login the AWS.

# Conclusion

In the project, DataCollector, is devote on help the client, who is newly enter into the NYC real estate market, to set up the DW/BI system that would support client’s investment team in decision-making. After study the NYC real estate market, the DataCollector decide to build the data warehouse on AWS and access the database through MySQL workbench. After the design, develop, construct, and test the system, The DataCollecter finished the project successfully.

# Appendix

The GitHub repository address is <https://github.com/qxia2/DAV-6100-Final-Project>.

The “README” file in the GitHub could be ignore. The use of the MySQL ddl file, csv file, and ipynb file is documented in the Appendix of the “DAV6100 Final Project Documentation.”

Data source include:

* + Nyc\_rolling\_medians.csv
  + NYCZips.csv
  + CityRealty\_scape\_loop.ipynb(Use this to scape raw data from CityRealty)
  + Zillow Data Profile.ipynb (Use this to access Zillow Data and check the data profile)
  + City\_realty\_raw\_data is stored in following S3 bucket due to large data size (<https://s3.console.aws.amazon.com/s3/buckets/yudav6100/?region=us-east-1&tab=overview>)
  + city realty data cleaning and push to S3.ipynb (Push city realty first cleaned data to S3)

Data source access:

* + Credential.txt contain the information needed to access AWS RDS through MySQL workbench.

Data Profiling and Cleaning:

* + NYC Rolling Sales Data- Data Profile
  + NYC Rolling Median.ipynb (Updated sql scripts and cleaned files Py and csv)
  + City Realty Full Data Cleaning and Medians.ipynb (Data Cleaning and Medians calculated)
  + city\_realty\_full\_cleaned.csv (This is the version that combined with other data and load into RDS)
  + nyc\_rolling\_medians.csv (This is the version that combined with other data and load into RDS)

Data Modeling and Injecting:

* + individaul\_fact\_ddl.sql
  + individual\_fact\_load.sql
  + monthly\_fact\_ddl.sql
  + monthy\_data\_load.sql
  + updateDimensionProc.sql
  + updateFactProc.sql
  + app.zip (lambada Zip file)

Data Visualization and Project Presentation

* + Group\_1\_Executive\_Presentation\_Final.pptx (Contains the link to Tableua Dashboard)