PROJECT REPORT ON

“DOB Job Application Filings Database”

Submitted by
Kexin Lian
Dongquan Qiu
Wenqian Liu
Junyao Xie
Chunye Xie

Prepared for
Apan 5310
SQL & Relational Databases
Instructor Nickolaos Machairas

Master’s in Applied Analytics
Department of School of professional Studies
Columbia University

August 10, 2020

Problem Statement

For this project, we assume that we're a data consulting team hired by the NYC Department of Buildings (DOB). Every year, there are numerous building construction going on in New York City, the number of residential and business buildings raised DOB receives a large amount of construction job applications, but only a small amount of them can be approved and put into action. The data comes from different sources and platforms, and is manually input into the system, which results in many null values, errors and typos in the information stored. We are hired by the client to build a database to make the data storage more accurate and efficient, and to generate some insights of building construction progress in NYC.

We spent several days exploring datasets. Some of them are pre-defined and others can not result in 15 tables at 3NF. This dataset is appropriate for this project because it is not predefined and there are enough attributes that can result in 15 tables at 3NF.

Our data resources include:

1. DOB job application filings: https://data.cityofnewyork.us/Housing-Development/DOB-Job-Application-Filings/ic3t-wcy2?fr%20om=groupmessage&isappinstalled=0
2. DOB Now: build-job application filings: https://data.cityofnewyork.us/Housing-Development/DOB-NOW-Build-Job-Application-Filings/w9ak-ipjd
3. DOB permit insurance: https://data.cityofnewyork.us/Housing-Development/DOB-Permit-Issuance/ipu4-2q9a
4. Labor Statistics for the New York City region: https://www.labor.ny.gov/stats/nyc/

Proposal

Our client has a lot of historical building construction job application data on hand, as well as new data that would come in which need to be stored on a timely basis. In order to help the client better manage the data and conduct analysis, we will develop a relational schema and load old data for our client. To fit the client's situation, we will develop a comprehensive schema that can allow them to store the data, extract the data, and update the database on an ad-hoc basis when new data comes in. The database will contain validation and constraints to ensure correct data types and data values are input into the system. At the end, we will build an interactive dashboard for visualization purposes. After we finish our analysis, they can also see the trends and gain valuable insights in the future.

After building the database and visualization dashboard, we will show the client how to use the database to generate insights that will benefit the department as well as the city. The building construction application database can provide valuable insights of city development and building safety. First, we can understand the types, locations, dates, scales and facilities of the buildings that were on the planning. By investigating the data, we can see the trend of how New York City evolved over years and depict a picture of how the city will look like in the near future. For the department, it's necessary to monitor and manage buildings in the city to develop better plans for city construction. This information can also be used as a tool to attract investors, residents, employers and promote the city economy.

Team structure and Timeline

Team structure

Name	Email	roles
Junyao Xie	jx2409@columbia.edu	General member
Dongquan Qiu	dq2148@columbia.edu	General member
Chunye Xie	cx2245@columbia.edu	General member
Wenqian Liu	wl2749@columbia.edu	General member
Kexin Lian	kl3191@columbia.edu	General member

Team contract

PROJECT VISION

Our group's grade expectation on this project is A and our sprint goal is A+.

ELEMENTS OF EFFECTIVE TEAMWORK

Communication

Due to the geographic diversity of our group members and for effective teamwork to occur, an environment of speaking free, open and expressing appropriate ideas and feelings is needed.

Each member is supposed to actively listen to other members, respond to other members in time, and provide effective non-judgmental feedback. We will use WeChat and Slack as tools for communication.

Participation

The success of our team relies on team members' participation and contributions. Our group will try the best to assign equal assignments and tasks to each member, respect and help each other when anyone needs. Team members need to contribute fully to the best of their ability. Members need to take initiative in participating in the group tasks, especially in areas where they may have strengths. Those with greater ability may also need to help those who may be struggling by guiding, coaching or critiquing. To make teams work well, without informing the group in advance, all members need to attend our weekly meetings, express their ideas and thoughts, actively communicate with others on platforms such as Slack, complete their assigned tasks, and revise their work after receiving feedback.

Progress And Assessment

Members of an effective team will contribute to an attitude of action and momentum. Often, progress is a good indicator of how well the team is working together. Regular assessment is necessary for a team to ensure it is continuing to work well together. An effective team is not afraid to make changes in how it is organized or in it's procedures so that improvement in achieving the goal/objective occurs.

CONTRACT AGREEMENT

This is an official contract. Once you have signed it you are accountable.

Name: Junyao Xie	Signature: Junyao Xie	Date: July 7, 2020
Name: Chunye Xie	Signature: Chunye Xie	Date: July 7, 2020
Name: Kexin Lian	Signature: Kexin Lian	Date: July 7, 2020
Name: Dongquan Qiu	Signature: Dongquan Qiu	Date: July 7, 2020
Name: Wenqian Liu	Signature: Wenqian Liu	Date: July 7, 2020

Project Timeline

Progress Stage	Start Date	Tasks/Requirements	Who is responsible for	Due Date
1	Jul 6, 2020	Finalize the project scenario. Detail the reasoning behind our choice, our motivation, the research we have performed and our initial plan of action.	All members	Jul 13, 2020
2	Jul 13, 2020	**Find the dataset**	All members	Jul 20, 2020
		Create the database schema.	Junyao Xie, Chunye Xie	
		Draw the ER diagram.	Dongquan Qiu, Wenqian Liu, Kexin Lian	
3	Jul 20, 2020	Revise the database schema and ER diagram according to feedback.	Junyao Xie	Jul 27, 2020
		Clean and combine Datasets into one.	Dongquan Qiu	
		Transform and enter the data to the database system using Python.	Chunye Xie, Wenqian Liu, kexin Lian	
4	Jul 27, 2020	Revise Python code for transforming and entering the data to the database system.	Kexin Lian	Aug 3, 2020
		Plan for how the customers will interact with the database system.	Chunye Xie, Junyao Xie	
		plan for redundancy and performance.	Dongquan Qiu, Wenqian Liu	
5	Aug 3, 2020	Metabase Dashboard	All members	Aug 10, 2020

The Most Challenging task

As highlighted on the table above, the most challenging task we met in this project is finding the dataset. The dataset in this project is unusual since it requires us to normalize it into at least 15 tables. Such a dataset might be common in any data-related organization, but it was difficult for students to find in an open resource. Everyone in our team had spent several days exploring datasets on kaggle, kdnuggets, Quora, etc., but all of them are not appropriate for our project. Some of them are pre-defined, and others can not result in 15 tables at 3NF. Finally, we found our dataset on a government web site.

Database Schema

Normalization plan and execution

There are 50 variables in our dataset combined and organized by four related datasets. We normalized the relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Our SQL code for creating tables is provided in Appendix.

Initial data

doc	fee_status	job_id	job_status	latest_action_date	professional_cert	prefiling_date
01	STANDARD	110083524	X	4/8/09 0:00	Y	2/4/08
01	STANDARD	103039766	X	12/19/02 0:00	Y	12/10/01
01	EXEMPT	102616885	R	5/3/00 0:00	NaN	5/2/00
01	EXEMPT	102616894	P	8/29/01 0:00	NaN	5/2/00
01	EXEMPT	402461180	P	9/25/06 0:00	NaN	9/20/06

paid_date	fully_paid_date	borough_name	house_number	building_bin	job_street_name	owner_company1_name	owner_company2_name
2/4/08	2/4/08	MANHATTAN	152	1084455	WEST 57 STREET	Carnegie Hall Towers	LYCEUM THEATER CORP
12/10/01	12/10/01	MANHATTAN	280	1035441	PARK AVENUE	Boston Properties	
5/2/00	5/2/00	MANHATTAN	600	1019483	THIRD AVENUE	Gale & Wen	AMERICAN MUSEUM OF NATURAI
5/2/00	5/2/00	BROOKLYN	22	4003540	30 PLACE	30th Place Holdings LLC	
9/20/06	9/20/06	STATEN ISLAND	601	1012268	WEST 26 STREET	RXR SL OWNER LLC	DK 1191 OCEAN AVENUE LLC

owner_first_name	owner_last_name	street_name	city	state	zip	building_type	landmarked	permit_type	unemployment_rate
Michael	Taub	152 West 57th Street	New York	NY	10019	OTHERS	N	EW	4.4.
Michael	Taub	152 West 57th Street	New York	NY	10019	OTHERS	N	EW	4.4
Robert	Riggs	30-30 Thomson Avenue	New York	NY	10021	OTHERS	NaN	EW	4.4
Mark	Karasick	455 MADISON AVENUE	NY	NY	10022	OTHERS	N	EQ	6.6
MITCHELL	GRANT	38 WEST 21ST STREET	NYC	NY	10010	OTHERS	NaN	NaN	6

labor_force	owner_cell_phone	owner_home_phone	permittee_first_name	permittee_last_name	permittee_phone	permit_status	permit_subtype	permit_sequence
848.9	2126031610	2128793672	STEPHEN	EISNER	2126974422	ISSUED	OT	1
848.9	2123086661	2128881234	CHI	CHAN	6463019922	ISSUED	OT	2
848.9	2129867787	2120372937	ROBERT	SCHUBERT	5162421188	ISSUED	FP	1
1171.5	2129243880	21283943653	MICHAEL	PALADINO	6318421700	ISSUED	FN	1
211	2128947000	2129009879	SANG	KIM	7184295000	NaN	NaN	NaN

filing_rep_first_name	filing_rep_last_name	manager_first_name	manager_last_name	applicant_first_name	applicant_last_name	applicant_company_name	applicant_professional_title	applicant_license_number
WILLIAM	VITACCO	Alissa	Morrow	Bruce	Lilker	Build it brother	PE	60859
BALDO	SACCHERI	Hibba	Swanson	MATT	MARKOWITZ	inspiration design build	RA	22409
NaN	NaN	Giorgia	Rubio	Frank	Eilam	blue ladder construction	RA	24701
WILLIAM	VITACCO	Tayler	Stott	Ravi	Shenoy	builder gorilla	PE	55232
TYANNA	HARRIS	Ariya	Holman	RODNEY	GIBBLE	design 4 you	PE	63244

gis_latitude	gis_longitude	manager_company_name	owner_company1_type	owner_company2_type	work_type1	work_type2
40.68725	-73.974168	Victoria Restoration Corp	CORPORATION	INDIVIDUAL	OT	
40.747521	-73.985239	50 SUTTON PLACE SO. OWNER	PARTNERSHIP	CORPORATION	OT	
40.751281	-73.981593	C/O BLDG Management Co., Inc	INDIVIDUAL	CORPORATION	FP	PL
40.792464	-73.964113	The Estate of Vincent Terranva	CORPORATION	PARTNERSHIP	EQ	BL
40.840096	-73.915569	ASTERDAM PEDIATRIC	CORPORATION	PARTNERSHIP	OT	SP

Satisfying 1NF

To satisfy 1NF, the values in each column of a table must be atomic. Entries in a column are the same type. All our initial tables satisfy these requirements except for the onwer_phone1, onwer_phone2(cell phone and home phone), work_type1, work_type2, owner_company1, and owner_company2. Therefore, we separated information out of the initial table and made a table called phone_number, another table called work_type, and a third table called owner_copmany as following:

*Green indicates that this column is the primary key.

Phone_number

phone_id	owner_id	owner_phone	owner_phone_type
1	1	2126031610	cell
2	1	2123086661	home
3	2	2129867787	cell
4	2	2129243880	home
5	3	2128947000	cell

*We assign each owner an id. The owner information table is shown in 2NF.

Work_type

job_id	work_type
110083524	OT
110083524	FP
102616885	FP
102616894	EQ
402461180	OT

Owner_company

owner_company_id	owner_id
1	1
2	1
3	3
4	4
5	4

*We assign each owner's company an id. The owner's company information table is shown in 2NF.

Satisfying 2NF

To conform to 2NF and remove duplicities, every non key attribute must depend on the key attribute. If we regard doc and job id as a composite key, not all non-key attributes depend on it. For example, unemployment_rate, labor_force, and borough depend only on job id rather than the composite key. In 2NF, every non candidate-key attribute must depend on the whole candidate key, not just part of it. Therefore, we divided them into the job info table in which job id and doc are the composite key and into the job location table in which only job id is the primary key. We repeated this process to other attributes. There are 8 tables after the second normalization.

Job_info

job_id	doc	fee_status	job_status	latest_action_date	house_number	building_bin	job_street_name
110083524	01	STANDARD	X	4/8/09 0:00	152	1084455	WEST 57 STREET
103039766	01	STANDARD	X	12/19/02 0:00	280	1035441	PARK AVENUE
102616885	01	EXEMPT	R	5/3/00 0:00	600	1019483	THIRD AVENUE
102616894	01	EXEMPT	P	8/29/01 0:00	22	4003540	30 PLACE
402461180	01	EXEMPT	P	9/25/06 0:00	601	1012268	WEST 26 STREET

owner_first_name	owner_last_name	street_name	city	state	zip	building_type	landmarked	permit_type	unemployment_rate
Michael	Taub	152 West 57th Street	New York	NY	10019	OTHERS	N	EW	4.4.
Michael	Taub	152 West 57th Street	New York	NY	10019	OTHERS	N	EW	4.4
Robert	Riggs	30-30 Thomson Avenue	New York	NY	10021	OTHERS	NaN	EW	4.4
Mark	Karasick	455 MADISON AVENUE	NY	NY	10022	OTHERS	N	EQ	6.6
MITCHELL	GRANT	38 WEST 21ST STREET	NYC	NY	10010	OTHERS	NaN	NaN	6

Phone number

phone_id	owner_id	owner_phone	owner_phone_type
1	1	2126031610	cell
2	1	2123086661	home
3	2	2129867787	cell
4	2	2129243880	home
5	3	2128947000	cell

Owner

owner_id	owner_first_name	owner_last_name	street_name	city	state	zip
1	Michael	Taub	52 West 57th Stre	New York	NY	10019
2	BRUCE	SIMON	70 Lexington Aven	New York	NY	10021
3	Robert	Riggs	30-30 Thomson Aver	New York	NY	10023
4	Mark	Karasick	MADISON AVEN	NY	NY	10022
5	MITCHELL	GRANT	WEST 21ST STRE	NYC	NY	10010

Owner_company

owner_company_id	owner_id
1	1
2	1
3	3
4	4
5	4

Work_type

job_id	work_type
110083524	OT
110083524	FP
102616885	FP
102616894	EQ
402461180	OT

Owner_company_info

owner_company_id	owner_company_name	owner_company_type
1	Carnegie Hall Towers	CORPORATION
2	Boston Properties	PARTNERSHIP
3	Gale & Wen	INDIVIDUAL
4	30th Place Holdings LLC	CORPORATION
5	RXR SL OWNER LLC	CORPORATION

Job location

job_id	house_number	building_bin	job_street_name	owner_id	borough_name	unemployment_rate
110083524	152	1084455	WEST 57 STREET	1	MANHATTAN	4.4
103039766	280	1035441	PARK AVENUE	2	BRONX	6.6
102616885	600	1019483	THIRD AVENUE	3	QUEENS	4.3
102616894	21	4003540	30 PLACE	4	BROOKLYN	5
402461180	601	1012268	WEST 26 STREET	5	STATEN ISLAND	4.8

labor_force	owner_first_name	owner_last_name	street_name	city
848.9	Michael	Taub	152 West 57th Street	New York
607.5	Michael	Taub	770 Lexington Avenue	New York
1139	Robert	Riggs	30-30 Thomson Avenue	New York
1171.5	Mark	Karasick	455 MADISON AVENUE	NY
211	MITCHELL	GRANT	38 WEST 21ST STREET	NYC

Building

building_bin	building_type	landmarked	permit_type	permit_status	permittee_first_name	permittee_last_name	permittee_phone
1084455	OTHERS	N	EW	ISSUED	STEPHEN	EISNER	2126974422
1035441	OTHERS	N	EW	ISSUED	CHI	CHAN	6463019922
1019483	OTHERS	NaN	EW	ISSUED	ROBERT	SCHUBERT	5162421188
4003540	OTHERS	N	EQ	ISSUED	MICHAEL	PALADINO	6318421700
1012268	OTHERS	NaN	NaN	NaN	SANG	KIM	7184295000

manager_first_name	manager_last_name	manager_company name
Alissa	Morrow	Victoria Restoration Corp
Hibba	Swanson	) SUTTON PLACE SO. OWNER'S INC
Giorgia	Rubio	C/O BLDG Management Co., Inc.
Tayler	Stott	The Estate of Vincent Terranva
Ariya	Holman	ASTERDAM PEDIATRIC

Satisfying 3NF

A table in third normal form is a table in 2NF that has no transitive dependencies. This means that all fields can be determined and only by the key in the table and no other column.

In our 2NF tables, some attributes do not only depend on the primary key, but also depend on the non-key attribute. For example, in the building table, permittee phone depends on permittees and permittees depend on building bin. Therefore, we removed such a transitive dependency by separating them into two tables as follows. The same issue exists in tables containing information about manager, borough, applicant, and filing_representative. We repeated the process and separated them from their 2NF tables. There are 15 tables in total after the third normalization. Our all 3NF tables are shown as follows:

Job_info

job_id	doc	fee_status	job_status	latest_action_date	professional_cert	prefiling_date
110083524	01	STANDARD	X	4/8/09 0:00	Y	2/4/08
103039766	01	STANDARD	X	12/19/02 0:00	Y	12/10/01
102616885	01	EXEMPT	R	5/3/00 0:00	NaN	5/2/00
102616894	01	EXEMPT	P	8/29/01 0:00	NaN	5/2/00
402461180	01	EXEMPT	P	9/25/06 0:00	NaN	9/20/06

paid_date	fully_paid_date	filing_representative_id	applicant_id
2/4/08	2/4/08	2	1
12/10/01	12/10/01	23	2
5/2/00	5/2/00	2	3
5/2/00	5/2/00	34	4
9/20/06	9/20/06	7	5

Job-location

job_id	borough_id	house_number	building_bin	job_street_name	owner_id
110083524	1	152	1084455	WEST 57 STREET	1
103039766	1	280	1035441	PARK AVENUE	2
102616885	1	600	1019483	THIRD AVENUE	3
102616894	2	21	4003540	30 PLACE	4
402461180	3	601	1012268	WEST 26 STREET	5

Borough_info

borough_id	borough_name	unemployment_rate	labor_force
1	MANHATTAN	4.4	848.9
2	BRONX	6.6	607.5
3	QUEENS	4.3	1139
4	BROOKLYN	5	1171.5
5	STATEN ISLAND	4.8	211

Owner

owner_id	owner_first_name	owner_last_name	street_name	city	state	zip
1	Michael	Taub	52 West 57th Stree	New York	NY	10019
2	BRUCE	SIMON	70 Lexington Aven	New York	NY	10021
3	Robert	Riggs	30 Thomson Aver	New York	NY	10023
4	Mark	Karasick	MADISON AVEN	NY	NY	10022
5	MITCHELL	GRANT	WEST 21ST STRE	NYC	NY	10010

Owner_company

owner_company_id	owner_id
1	1
2	1
3	3
4	4
5	4

Owner_company_type

owner_company_id	owner_company_name	owner_company_type
1	Carnegie Hall Towers	CORPORATION
2	Boston Properties	PARTNERSHIP
3	Gale & Wen	INDIVIDUAL
4	30th Place Holdings LLC	CORPORATION
5	RXR SL OWNER LLC	CORPORATION

Building

building_bin	building_type	landmarked	permit_type	permit_status	permit_subtype	permit_sequence
1084455	OTHERS	N	EW	ISSUED	OT	1
1035441	OTHERS	N	EW	ISSUED	OT	2
1019483	OTHERS	NaN	EW	ISSUED	FP	1
4003540	OTHERS	N	EQ	ISSUED	FN	1
1012268	OTHERS	NaN	NaN	NaN	NaN	NaN

manager_id	GIS_LATITUDE	GIS_LONGITUDE	permittee_id
1	40.765185	-73.979279	1
2	40.754694	-73.964242	2
3	40.782566	-73.948313	3
4	40.753612	-73.992417	4
5	40.756651	-73.972264	5

Phone_number

phone_id	owner_id	owner_phone	owner_phone_type
1	1	2126031610	cell
2	1	2123086661	home
3	2	2129867787	cell
4	2	2129243880	home
5	3	2128947000	cell

Permittee

permittee_id	permittee_first_name	permittee_last_name	permittee_phone
1	STEPHEN	EISNER	2126974422
2	CHI	CHAN	6463019922
3	ROBERT	SCHUBERT	5162421188
4	MICHAEL	PALADINO	6318421700
5	SANG	KIM	7184295000

Filing_representatives

filling_representative_id	filing_rep_first_name	filing_rep_last_name
1	WILLIAM	VITACCO
2	BALDO	SACCHERI
3	NaN	NaN
4	LEONARD	HERCZEG
5	TYANNA	HARRIS

Manager

manager_id	manager_first_name	manager_last_name	manager_company_id
1	Alissa	Morrow	34
2	Hibba	Swanson	56
3	Giorgia	Rubio	23
4	Tayler	Stott	4
5	Ariya	Holman	5

Applicant

applicant_id	applicant_first_name	applicant_last_name	applicant_professional_title	pplicant_license_numbe	applicant_company_id
1	Bruce	Lilker	PE	60859	1
2	MATT	MARKOWITZ	RA	22409	2
3	Frank	Eilam	RA	24701	3
4	Ravi	Shenoy	PE	55232	4
5	RODNEY	GIBBLE	PE	63244	5

Applicant_company_info

applicant_company_id	applicant_company_name
1	Build it brother
2	inspiration design build
3	blue ladder construction
4	builder gorilla
5	design 4 you

Work_type

job_id	work_type
110083524	OT
110083524	FP
102616885	FP
102616894	EQ
402461180	OT

Manager_company_info

manager_company_id	manager_company name
1	Victoria Restoration Corp
2	50 SUTTON PLACE SO. OWNER'S INC
3	C/O BLDG Management Co., Inc.
4	The Estate of Vincent Terranva
5	ASTERDAM PEDIATRIC

ER diagrams

We created an ER diagram on LucidChart. See Appendix “ER diagram”. Here is the link to LucidChart:
https://app.lucidchart.com/documents/edit/6b149624-2460-4009-9d59-d029f81fbc21/0_0

ETL plan and execution

We use Python to perform the ETL process and we used the Jupyter Notebook as our coding tool because it shows the results line by line and provides great visualization after each process. It can also be used to show the process to others.

The first step is to load necessary packages that will allow us to perform the ETL process. We mainly used 'Pandas' for dataframe manipulations and 'Numpy' for mathematical computations. We also used sqlalchemy to build connections between Python and PostgreSql. We wrote the SQL code based on our normalization plan in advance and tested the code for each table in PostgreSql to ensure correctness. We then use Python to connect to the database and pass the statements to create the tables in our database.

With the database and all tables created, we started to extract, transform and load the dataset into the database. We loaded the dataset from the local file path. Our dataset is saved as the Excel format, therefore we use the read_Excel function from Pandas to load the data. Before making the modifications to the dataset, we performed some dataset exploration process to ensure that our normalization plan properly corresponds to the information stored in the dataset. Next, we started our ETL process and created small data frames for each table we have in the database. We extracted values based on the attributes for each table and added unique identifiers as the primary keys. We first load the data into those tables which can exist independently, and then load the data into the tales with foreign key constraints.

In the process of ETL, we discovered some inherent errors in the original dataset. For example, there is a "zip code" column in the dataset. We designed the relevant table and assigned integer data type to this attribute since we assume that all zip code should be stored in a 5 digit integer format such as "12345" after initial examination of the original dataset. However, there are values such as '12345X' or '11001\', which violates our data type constraint. Therefore, for such values, we made some manual adjustments to the wrong values in order for the dataset to be loaded into the database. For similar errors in other tables, we use different ways to make small adjustments.

After making proper adjustments and creating dataframes, we repeated the ETL process for 15 tables and successfully loaded all the data into our database. Due to the fact that the Python code is too long, we won't be attaching it to the appendix. You can find the full code and results of our ETL process by clicking the following link:
https://github.com/cinnabar723/5310-project/blob/master/Project_ETL_v3.ipynb

Interaction

1. **How your customers will interact with the database system you designed.**
   NYC Department of buildings(DOB) requires an application submitted for review before any building construction project begins in New York City. Applicants need to download

the form from DOB website and mail it to DOB. The database our group designs will comply with DOB's process and help DOB perform tasks. A complete process of this interaction includes collecting data, cleaning data, organizing data, and transferring data to a cloud data warehouse for future analysis.

Considering the volume and complexity of DOB's data, we decide not to use the Hadoop system. Although at the present, our customers are using the paper to collect data, they are moving towards an electronic filing system where applicants and companies can submit forms online. Based on DOB's current situation, firstly, we will build a html website for them to collect the data. Second, DOB uses a relational database management system such as MySQL and PostgreSQL to create and store their source data. For this job application filling data, we created 15 tables, so DOB's employees need to create 15 tables in a relational database system as we created in the first place. Then, application information will be automatically updated into MySQL every one hour by Fervvo's database connector. Frevvo's RESTful Database Connector uses Extensible Markup Language (XML) and JavaScript Object Notation (JSON) to connect the HTML forms and our database via a secure HTTPS connection. Third, since job application filling is just a small segment of DOB's service, DOB needs an ETL tool such as Talend, Informatica, and Kettle to extract, transform, and load data. For example, ETL tools can save time and be functional when DOB needs to combine job application data with other data like building data, or when DOB wants to migrate data from their database to other departments' databases for comprehensive decision-making. At last, structured data will be transferred and stored in a cloud database.

2. **What will you implement for analysts (direct querying) and for "C" level officers (reports)? What tools are you using?**

It is more convenient to perform database actions with a programming language. The analysts can use some programming tools like Jupyter Notebook(Python), R studios(R programming), etc. to connect to our database, directly query the database, extract the data they need and analyze the data. Besides, they can also use some non-technical tools like looker and tableau to extract and analyze the data.

As for "C" level officers, we will create a real time dashboard to visualize the data and show the results. It is time-consuming and can't update timely when we manually make and update a dashboard. Therefore, we will create a SQL dashboard by a tool called Metabase that can pull in data from our database, build custom data visualizations and dashboard, and update the dashboard automatically and timely when there are some

updates in the database. It has data visualization and dashboard updates in near-real time(refreshing) up to every 60 seconds.

3. **Did you plan for redundancy and performance?**
   In addition, we planned for the redundancy and performance. First, the data replication can help prevent data redundancy by storing the same data in multiple locations. We can also have a central and master field to update all of the places where the data is redundant through one central access point. It can ensure the consistency and receive the information needed. Second, certain cloud platforms and databases, including AWS and Talend Data Fabric will help improve the performance and data quality. It will both ensure the data quality and visualization report for different internal staffs as well as obtaining real-time data updates.

Analytics Applications

According to the analytical procedures, we are able to apply SQL and Python codes to generate valuable insights through the number and visualization as well. Certain questions are listed below as examples.

1. How many jobs are professionally certified? (#51046)
2. How many jobs available are there in each borough in NYC?
3. How many jobs are available in the last 5 years?
4. What is the most common work type?
5. How many residential buildings (family) are in job applications?
6. What are the most popular streets for building construction?
7. Top 5 manager companies that engaged the most construction?
8. How does the top 1 company distribute its construction on Map?
9. Top 5 owner companies that own the most constructions here?
10. What is the trending of job markets and which area can be the most potential one?

At the beginning of the analytics, we decided to explore the insights in the application and job datasets, in order to find out the characteristics in the job market over the past few years. First, there are 51046 jobs that are professionally certified, which require applicants with certain professional knowledge for the work. The most 5 work types are OT, PL, EQ, MH, and SP. Moreover, the number of available jobs is gradually increasing, from 779 in 2015 to 20036 in 2019, with ~225% compounded growth rate according to the past 5 years. Meanwhile, among boroughs in the New York region, Queens offers the most available jobs in the market and Manhattan offers the second most jobs. In addition, Manhattan obtains the lowest unemployment rate in 2019, indicating that the opportunities and demands are increasing in this specific

borough. We believe that Queens and Manhattan maintain the most market shares of jobs with our research and analytics.

On the other hand, we explored the data from the building's perspective, in order to understand what companies most engaged in the work and what type are they. First, 31009 out of 100009 buildings are categorized as 1-2-3 FAMILY, while the rest are categorized as OTHERS. 1-2-3 FAMILY type building counts as ~44.9% of all. Among the building constructions, the most popular borough is in Queens while the most popular street name is BROADWAY street. It matches our conclusion that Queens maintain the most available jobs above as well. Second, the top 3 companies that own the most construction buildings are NY School Construction Authority, NYC HPD, and NYCHA. These are the local institutions in the area. In addition, BEECH ASSO is the top 1 manager company that engaged the most construction buildings in this case. The majority of its distribution are located in Manhattan and Queens. Overall, these two boroughs are the most popular areas with construction buildings work and maintain huge opportunities to offer work among all boroughs.

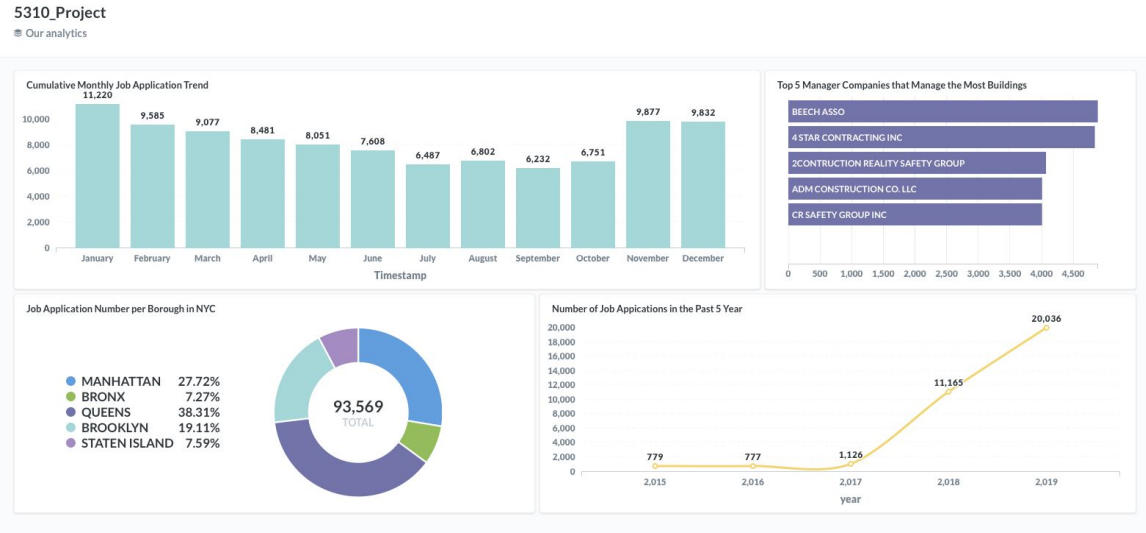
Metabase

We put four insights into our Metabase dashboard. The first one was automatically generated by Metabase, the other three were generated by Sql query executions in Metabase and further visualization.

First, we showed the annual and monthly trend of the number of job applications in each month for all years. We can see that there is a clear declining trend from January to October, then the number of job applications increased from November to December, which means people plan more construction work during winters, and plan less during summer. Second, we listed the top 5 companies that manage the most buildings. Those companies are normally large companies and monitor most of the city's constructions. It is always a good idea to maintain good relationships with these companies and keep an eye on them. Third, we listed the number of job applications for each borough in NYC. From the data shown, we can see that most building construction planning took place in Queens, and least in Bronx. Lastly, we drew a curve and showed the job construction planning growth situation for the past 5 years. We can see a dramatic increase in the number of job applications since 2017, which could be an indicator of growth in favorable policies, number of residents and boost of the economy.

The dashboard provides a basic visualization and demonstration of the situation of the building construction progress in NYC from various aspects. Audiences will have an understanding of the

amount, location, time and trend of the constructions based on the information provided on the dashboard. The dashboard can be used as a great presentation to higher level management for a brief introduction to the current construction situation. The information including years and measurement in the dashboard can be modified to satisfy different needs.



The link for the interactive dashboard can be found below:
http://localhost:13195/public/dashboard/3a35b13d-7a1f-4e0b-bfdc-817823ea9362

Conclusion

We aimed to build an accurate and efficient database for our client. Firstly, we executed the normalization and drew an ER diagram to figure out the database schema. Then we prepared the sql code for each table. After that, we ETL the dataset into our database through Python. Besides, we made a dashboard for the C-level officers to timely check the change.

Appendix

Schema SQL Code

```
create table applicant_company_info(
    applicant_company_id integer,
    applicant_company_name varchar(200),
     PRIMARY KEY (applicant_company_id)
);

create table applicant(
    applicant_id integer,
```

```
    applicant_first_name varchar(50),
    applicant_last_name varchar(50),
    applicant_professional_title varchar(50),
    applicant_license_number varchar(10),
    applicant_company_id integer,
    primary key (applicant_id),
    FOREIGN KEY (applicant_company_id) REFERENCES applicant_company_info
(applicant_company_id)
);

create table permittee(
    permittee_id integer,
    permittee_first_name varchar(50),
    permittee_last_name varchar(50),
    permittee_phone varchar(20),
    primary key (permittee_id)
);

create table filing_representative(
    filing_representative_id integer,
    filing_rep_first_name varchar(50),
    filing_rep_last_name varchar(50),
    primary key (filing_representative_id)
);

create table borough_info(
    borough_id integer,
    borough_name varchar(50),
    unemployment_rate varchar(10),
    labor_force integer,
    primary key (borough_id),
    check (borough_name in('BRONX','BROOKLYN','QUEENS','STATEN
ISLAND','MANHATTAN'))
);

create table manager_company_info (
    manager_company_id integer,
```

```
    manager_company_name varchar(200),
    PRIMARY KEY (manager_company_id)
);

create table manager(
    manager_id integer,
    manager_first_name varchar(50),
    manager_last_name varchar(50),
    manager_company_id integer,
    primary key (manager_id),
    FOREIGN KEY (manager_company_id) REFERENCES
manager_company_info(manager_company_id)
);

create table owner(
    owner_id integer,
    owner_first_name varchar,
    owner_last_name varchar,
    street_name varchar(200),
    city varchar(50),
    state char(2),
    zip integer,
    primary key (owner_id)
);

create table owner_company_type(
    owner_company_id integer,
    owner_company_name varchar(100),
    owner_company_type varchar(100),
    PRIMARY KEY (owner_company_id)
);

create table owner_company(
    owner_company_id integer,
    owner_id integer,
    primary key (owner_company_id, owner_id),
```

```
    FOREIGN KEY (owner_company_id) REFERENCES owner_company_type
(owner_company_id),
    FOREIGN KEY (owner_id) REFERENCES owner (owner_id)
);

create table phone(
    phone_id integer,
    owner_id integer,
    owner_phone varchar(20),
    owner_phone_type varchar(10),
    primary key(phone_id),
    foreign key(owner_id) references owner(owner_id)
);

create table building(
    building_bin integer,
    building_type varchar(50),
    landmarked char(1),
    permit_type varchar(2),
    permit_status varchar(15),
    permit_subtype varchar(2),
    permit_sequence varchar(5),
    manager_id integer,
    permittee_id integer,
    gis_latitude numeric(10,6),
    gis_longitude numeric(10,6),
    primary key(building_bin),
    Foreign key (manager_id) references manager (manager_id),
    Foreign key (permittee_id) references permittee (permittee_id),
    check (landmarked in('Y','N','L','C')),
    check (building_type in('1-2-3 FAMILY','OTHERS')),
    check (permit_status in ('IN PROCESS','ISSUED','REVOKE','RE-ISSUED')),
    check (permit_type in ('AL','DM','EQ','EW','FO','NB','PL','SG'))
);

create table job_location(
    job_id integer,
```

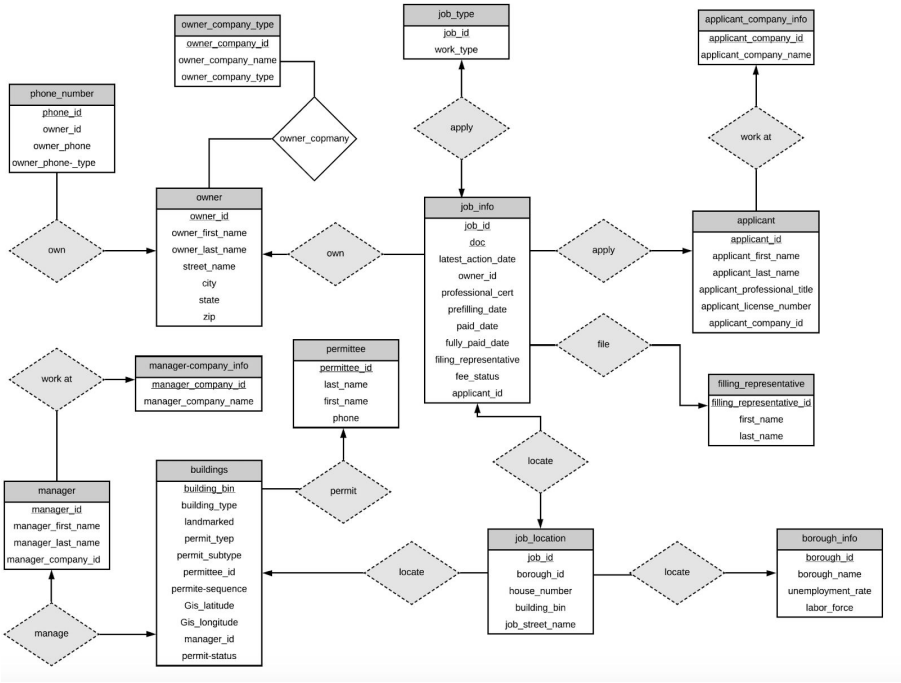
```
    borough_id integer,
    house_number varchar(20),
    building_bin integer,
    job_street_name varchar(100),
    owner_id integer,
    primary key(job_id),
    Foreign key (borough_id) references borough_info(borough_id),
    Foreign key (building_bin) references building(building_bin),
    FOREIGN key (owner_id) REFERENCES owner (owner_id)
);

create table job(
    job_id integer,
    doc_id integer,
    fee_status varchar(20),
    job_status varchar(1),
    latest_action_date date,
    professional_cert varchar(1),
    prefiling_date date,
    paid_date date,
    fully_paid_date date,
    filing_representative_id integer,
    applicant_id integer,
    primary key(job_id, doc_id),
    FOREIGN KEY (job_id) REFERENCES job_location (job_id),
    FOREIGN KEY (filing_representative_id) REFERENCES filing_representative
(filing_representative_id),
    FOREIGN KEY (applicant_id) REFERENCES applicant (applicant_id),
    check (professional_cert in('J','N','Y')),
    check (job_status in('3','A','B','C','D','E','F','H','J','K','P','Q','R','U','X'))
);

create table job_type(
    job_id integer,
    work_type varchar(2),
    primary key (job_id, work_type),
    FOREIGN KEY (job_id) REFERENCES job_location (job_id),
```

check(work_type in ('BL','CC','EQ','FA','FB','FP','FS','MH','OT','SC','SD','SF','SH','SP','PL'))
);

ER Diagram



Analytics Procedures Code (both SQL & Python)

SQL Part

#Question - How many jobs are professionally certified?

#Answer - 51046

```
SELECT COUNT(*)
FROM job
WHERE professional_cert = ‘Y’;
```

#Question - How many jobs available are there in each borough in NYC?

```
SELECT b.borough_id, b.borough_name, COUNT(job_id) AS ct
FROM borough_info b
INNER JOIN job_location j
ON b.borough_id = j.borough_id
GROUP BY b.borough_id
ORDER BY b.borough_id ASC;
```

Data Output

	borough_id [PK] integer	borough_name character varying (50)	ct bigint
1	1	MANHATTAN	25941
2	2	BRONX	6802
3	3	QUEENS	35845
4	4	BROOKLYN	17883
5	5	STATEN ISLAND	7098

#Question - top 5 company that has the most building

```
SELECT manager_company_name, COUNT(building_bin) AS ct
FROM manager_company_info c
INNER JOIN manager m
ON c.manager_company_id = m.manager_company_id
INNER JOIN building b
ON b.manager_id = m.manager_id
GROUP BY c.manager_company_id
ORDER BY ct DESC
LIMIT 5;
```

	manager_company_name character varying (200)	ct bigint
1	BEECH ASSO	4887
2	4 STAR CONTRACTING INC	4840
3	2CONTRUCTION REALITY SA...	4069
4	ADM CONSTRUCTION CO. LLC	4007
5	CR SAFETY GROUP INC	4006

#Question - How many jobs are available in the last 5 years?

```
SELECT EXTRACT(year FROM latest_action_date) AS year, COUNT(job_id) AS ct
FROM job
GROUP BY 1
ORDER BY year DESC
OFFSET 1
LIMIT 5;
```

	year double precision	ct bigint
1	2019	20036
2	2018	11165
3	2017	1126
4	2016	777
5	2015	779

#Question - What is the most common work type (most jobs are this type)

```
SELECT work_type, COUNT(*)
FROM job_type
GROUP BY work_type
ORDER BY COUNT(*) DESC;
```

Data Output

	work_type character varying (2)	count bigint
1	OT	28550
2	PL	25285
3	EQ	22554
4	MH	16033
5	SP	13277
6	FP	12195
7	BL	12069

PYTHON Part

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
%cd /Users/dq/Desktop
df = pd.read_excel('APAN5310_team2_data_v8.xlsx')
```

#Question - how many residential buildings (family) in different type in job applications, and ratio

#Answer - 31009 out of 100009 are 1-2-3 family building types, with 44.94% of all.

```
df1 = df.building_type.value_counts().rename_axis('type').reset_index(name='counts')
#find total number of buildings with building_type
df[['building_type']].count()
#calculate the ratio
ratio = df1.loc[1,'counts']/df1.loc[0,'counts']
Ratio
```

	type	counts
0	OTHERS	69000
1	1-2-3 FAMILY	31009

```
#calculate the ratio
ratio = df1.loc[1,'counts']/df1.loc[0,'counts']
ratio
```

```
0.4494057971014493
```

 # Question - what are the most popular streets and boroughs for building construction?

#Answer - Most popular: BROADWAY street and QUEENS borough

```
df2 = df.drop_duplicates(subset='building_bin')
df2.job_street_name.value_counts()
df2.borough_name.value_counts()
```

```
QUEENS            25437
BROOKLYN          13176
MANHATTAN         10481
STATEN ISLAND      5795
BRONX              5127
Name: borough_name, dtype: int64
```

```
BROADWAY              667
JAMAICA AVENUE        319
NORTHERN BOULEVARD    291
ROOSEVELT AVENUE      202
LIBERTY AVENUE        193
                     ...
E. 99TH ST.             1
S 3 ST                  1
W 134 STREET            1
18 AVE.                 1
WEST 16 ST              1
Name: job_street_name, Length: 8660, dtype: int64
```

#Question - top5 manager companies that engaged the most constructions?

#Answer - top5 manager companies are: BEECH ASSO, 4 STAR CONTRACTING INC, 2CONTRUCTION REALTY SAFETY GROUP, ADM CONSTRUCTION CO. LLC, CR SAFETY GROUP INC.

```
df3 =
df[['building_bin','job_street_name','borough_name','manager_company_name','gis_latitude','gis_longitude']]
df3.drop_duplicates(subset='building_bin', inplace=True)
```

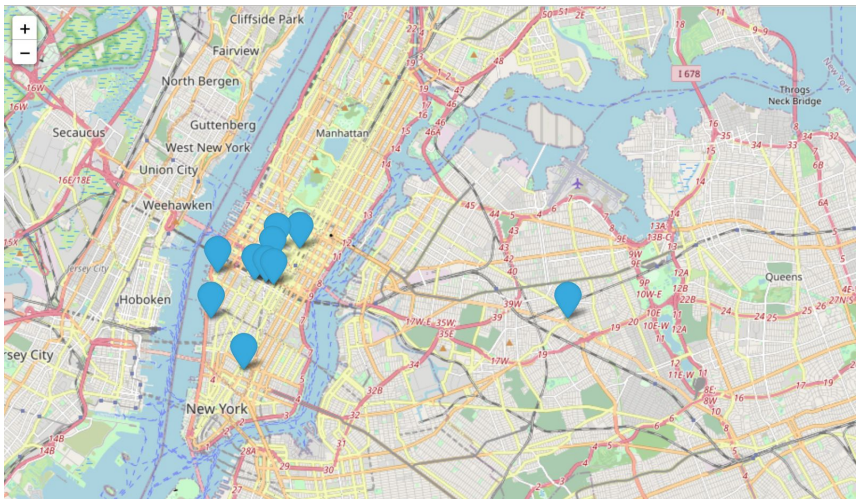
```
df3.manager_company_name.value_counts()[:5]

df4 = df3.loc[df['manager_company_name']=='BEECH
ASSO',{'building_bin','gis_latitude','gis_longitude'}]
df4[['manager_company']]=df3[['manager_company_name']]
df4.head(10)
```

```
BEECH ASSO                          4887
4 STAR CONTRACTING INC              4840
2CONTRUCTION REALITY SAFETY GROUP    4069
ADM CONSTRUCTION CO. LLC            4007
CR SAFETY GROUP INC                 4006
Name: manager_company_name, dtype: int64
```

```
#Question - Map the top1 company's construction
#Answer - graph below
import folium
m = folium.Map(
   location=[df3['gis_latitude'][0],df3['gis_longitude'][0]],
   zoom_start=12)
folium.Marker([40.723843,-73.996356],
          popup='1007929',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.738151,-74.008462],
          popup='1085734',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.757369,-73.984096],
          popup='1022633',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.748790,-73.992244],
          popup='1014343',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.747521,-73.985239],
          popup='1015853',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.748400,-73.988144],
          popup='1083630',
          icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.750934,-74.005955],
```

```
            popup='1012268',
            icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.757582,-73.975805],
            popup='1035454',
            icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.738215,-73.877274],
            popup='4045260',
            icon=folium.Icon(icon='blue')).add_to(m)
folium.Marker([40.753818,-73.985696],
            popup='1080614',
            icon=folium.Icon(icon='blue')).add_to(m)
m
```



#Question - which owner have the most constructions here (owner_company1_name)
#Answer - NY SCHOOL CONSTRUCTION AUTHORITY has the most constructions.

```
df5 =
df[['building_bin','owner_company1_name','owner_company1_type']].drop_duplicates(subset='building_bin')

df6 = df5[df5.owner_company1_name != 'NONE']
df6 = df6[df6.owner_company1_name != 'OWNER']
df6 = df6[df6.owner_company1_name != 'owner']
df6 = df6[df6.owner_company1_name != 'Owner']

df6.owner_company1_name.value_counts()
```

```
NY SCHOOL CONSTRUCTION AUTHORITY    284
NYC HPD                             283
NYCHA                               177
NYC SCA                             177
NYC HOUSING AUTHORITY               161
                                   ...
Prospect Seeley Housing Corp.         1
PLAZA                                 1
Treo Builders, LLC                    1
420-428 AMSTERDAM LLC                 1
1                                     1
Name: owner_company1_name, Length: 26941, dtype: int64
```