Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

THE CITY COLLEGE OF NEW YORK

The Ultimate Separation and Organization of Substantial Datasets:

Project 1

NYC
311

Georgios Ioannou

CSC 33600

Professor Richard Khan

25 March 2022

Project 1

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

TABLE OF CONTENTS

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

TEAM

Team Member Name: Georgios Ioannou

Team Member Email: gioanno000@citymail.cuny.edu

Team Member EmplID: 23927106

Team Name: The Unbeatable

311 Service Requests from 2010 to Present NYC Open Data Website:

https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

The Ultimate Separation and Organization of Substantial Datasets:

Project 1

INTRODUCTION

We live in a world full of data. Data is coming and going all the time and if left unorganized, then it produces no benefit to the users and becomes extremely messy. Data is traveling across the world each and every second and as a result, its size increases enormously each day. According to the German database company 'Statista' which specializes in market and consumer data, the total amount of data created, captures, copied, and consumed globally is estimated to increase to more than 18 zettabytes by 2025. This statistic indicates that there is an immediate need for people to be able to separate and organize data so that they can search and retrieve what they want faster and more efficiently. Moreover, a company's data contains the essential elements needed to manage a company's most valuable assets. Therefore, understanding and organizing this data will help a company to attain better business intelligence and as a result, play an important role in the company's success. Separating and organizing a dataset involves several techniques and steps that ensure that the data is organized correctly and eliminates duplicate data.

This project will illustrate how to separate and organize an unorganized substantial dataset from NYC Open Data. The dataset that this project focuses on is called '311 Service Requests from 2010 to Present.' This dataset is owned by the NYC OpenData and it is provided by the Department of Information Technology & Telecommunications(DoITT). This dataset was first created on October 10, 2011, and is updated daily. This is a huge dataset and until March 20, 2022, had a total of 28, 28,717 rows where each row represents a 311 Service Request. The number of rows increases every day and as a result, there is a huge need to organize this dataset as it has already become meaningless, ambiguous, and it is extremely difficult to study and retrieve data. The number of columns of this dataset is 41 and it is constant meaning that is not increased or decreased over the years.

SOFTWARE AND FILES REQUIREMENTS

To recreate and follow this project's procedure you will need the following software and files:

1. Laptop or Desktop Computer(To run MySQL Server and MySQL Workbench)
2. MySQL Server(To execute SQL queries)
3. MySQL Workbench(The official graphical user interface too for MySQL)
4. The two provided SQL files(the two SQL queries produce the unorganized and organized datasets)
5. The provided CSV file(A portion of the original dataset used to import data to the database. You can also go ahead and download the original 20GB (as of March 20, 2022) dataset.)

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

PROCEDURE

1. Download the provided SQL files and the CSV file.
2. Open MySQL Workbench and connect to your MySQL Server Connection using your password.
3. In the Query 1 file, execute the following statement:

   SHOW VARIABLES LIKE "secure_file_priv";

   This statement will display a table as seen in Image 1. Navigate to the folder specified in the **Value** column. For instance, Image 1 instructs us to navigate to the folder 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\'
4. Drag and drop the provided CSV that you downloaded in Step 1 to the folder indicated by the Value column as described in Step 3.
5. Open the SQL file named ‘GEORGIOS_IOANNOU_Project_1_Original’
6. Go to line 58 of the SQL file that was opened in Step 5. This is the LOAD DATA INFILE statement. **Change** the file path of line 58 to match the full file path of the CSV file which you changed its location in Step 4. Make sure that the file path in line 58 uses **only** forward slashes and not backslashes. For example, this is **NOT** allowed 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\GEORGIOS_IOANNOU_Project_1.csv'.
7. Execute the current SQL file.

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

8. If you get the error shown in Image 2, then you have inserted the CSV file in the wrong folder. Please go back to Step 3 and follow the instructions carefully. If the execution was successful, then a new database and a new table will be created. Moreover, the table will be displayed on your screen. This is the unorganized dataset as provided in the NYC Open Data.
9. Open the SQL file named ‘GEORGIOS_IOANNOU_Project_1_Organized’.
10. Execute the current SQL file.
11. This SQL file will create a new database and new tables as illustrated by the ER-Diagram in Image 3. In addition to this, the SQL file will display all tables with their rows on your screen. You must expect 15 tables to display on your screen.

METHOD OF SEPARATING/ORGANIZING THE DATASET INTO THEIR TABLES AND RELATIONSHIPS

As mentioned in the introduction, the ‘311 Service Requests from 2010 to Present’ is a huge dataset with more than 28 million rows and 41 columns. Each row in this dataset represents a 311 Service Request. With so much data it is obvious that there will be many duplicate values and redundancy. In fact, this is what we will try to do in this project. We will try to eliminate as many duplicate and unnecessary values as possible by creating new tables and organizing data.

The main purpose of this database is to keep historical records of all 311 Service Requests from 2010 to present. The people who are mainly interested in looking in this dataset are the New York City Council members to monitor the activities and requests in the city.

However, citizens may also want to study this dataset by curiosity. Regardless of who the reader of this dataset is, we must make it as easy as possible for them to study, search, and retrieve data.

The two main tables in this database are the **service_request** table and the **address** table. The first main table is called the **service_request** table and holds all the information needed to describe a 311 Service Request.

A 311 Service Request needs to be assigned to one agency. Therefore, an **agency** table is needed which will store all the agencies of New York City. This table will reduce repeated data such as the agency name. However, one agency can be assigned to many 311 Service Requests. Therefore, the relationship between the **agency** table and the **service_request** table is One to Many.

Moreover, a 311 Service Request is described by one complaint. Therefore, a **complaint** table must be created to store all the information needed to describe a complaint. However, one complaint can describe many 311 Service Requests because many of these requests can have the same complaint. Therefore, the relationship between the **complaint** table and the **service_request** table is One to Many.

In addition to this, a 311 Service Request is resolved by one and only one resolution. This is because each resolution has a specific date that was last updated. Therefore, a **resolution** table is needed and must have a One to One relationship with the **service_request** table.

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

A 311 Service Request may be identified as a taxi incident. Therefore, a **taxi** table is required to store all the information that describes a taxi incident. The relationship between the **taxi** table and the **service_request** table is One to Many because one **taxi** record can be involved in many 311 Service Requests.

Similarly, a 311 Service Request may be identified as a bridge/highway incident. Therefore, a **bridge_highway** table is required to group all the information that describes a bridge/highway incident. The relationship between the **bridge_highway** table and the **service_request** table is One to Many because one **bridge_highway** record can be involved in many 311 Service Requests.

Each 311 Service Request is located at a particular location described by a specific address. Therefore, an **address** table is required to store all the information needed to locate the Service Request. This is an important table because people need to be able to locate and filter a Service Request quickly. . The relationship between the **address** table and the **service_request** table is One to Many because one **address** record can be involved in many 311 Service Requests and as a result locate many Service Requests. For example, a person from one address can make many Service Requests.

The second main table is the **address** table mentioned in the previous paragraph. Any data that describes the location of a 311 Service Request will be grouped into this table. However, because this table will be extremely redundant it needs to be split up into many tables.

Georgios Ioannou

Professor Richard Khan

CSC 33600

25 March 2022

The first table that will reference the **address** table is called the **incident** table. This table stores information that will locate the address such as the zip code. The relationship between these tables is One to One because one address can have one and only one zip code.

Every address can be located by the cross streets and intersection streets. Therefore, two tables are required to store this information. The first table called the **cross_street** table will store information about the cross streets. The second table called the **intersection_street** table will store information about the intersection streets. The relationship between the **address** table and the **cross_street** table is One to One because an address is located by specific cross streets. Similarly, the relationship between the **address** table and the **intersection_street** table is One to One.

A specific scene can be used to describe the address to make it clearer for responders to locate the Service Request. Therefore, information such as Borough-Block-Lot(bbl) and landmark must be grouped because they describe the scene. Therefore, a **scene** table is created and has a One to One relationship to the **address** table. This relationship is used because one address can have one and only one bbl and can be described by only one scene. It will be impossible for responders to locate an address that was described by more than one scene.

A scene can be a park. Therefore, all information related to parks needs to be stored in a new table called **park**. The relationship between the **park** table and the **scene** table will be One to One because as with the address, one park can only describe one scene. The **park** needs to also reference the **community_distict** table to know where exactly the park is located. The relationship between the **community_district** table and the **park** table is One to Many because one community district can have many parks.

To locate a 311 Service Request fast the community district is also needed. Therefore, a table named **community_district** must be created to store information such as the borough name and the community board. One community district has many addresses. Therefore, the relationship between the **community_district** table and the **address** table is One to Many.

Every address can be located by specific state plane coordinates and latitude/longitude. Therefore, two tables are required to store this information. The first table called the **coordinate_state_plane** table will store information about the state plane coordinates. The second table called the **coordinate_location** table will store information about the latitude and longitude. The relationship between the **address** table and the **coordinate_state_plane** table is One to One because an address is located by specific state plane coordinates. Similarly, the relationship between the **address** table and **coordinate_location** table is One to One.

CONCLUSION

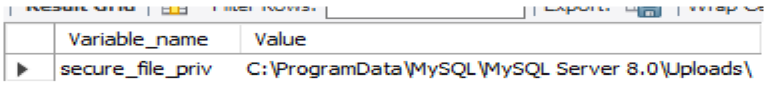
Overall, data organization is crucial and it is extremely important for success. Without separating, designing, and organizing data we will not be able to accomplish most of our everyday tasks such as scheduling an appointment with a doctor. As technology evolves, the amount of data increases exponentially and the requirement to organize data becomes more and more crucial.

Georgios Ioannou

Professor Richard Khan

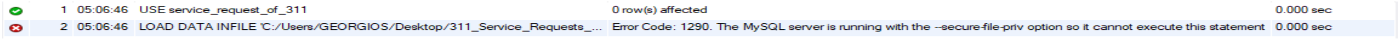
CSC 33600

25 March 2022



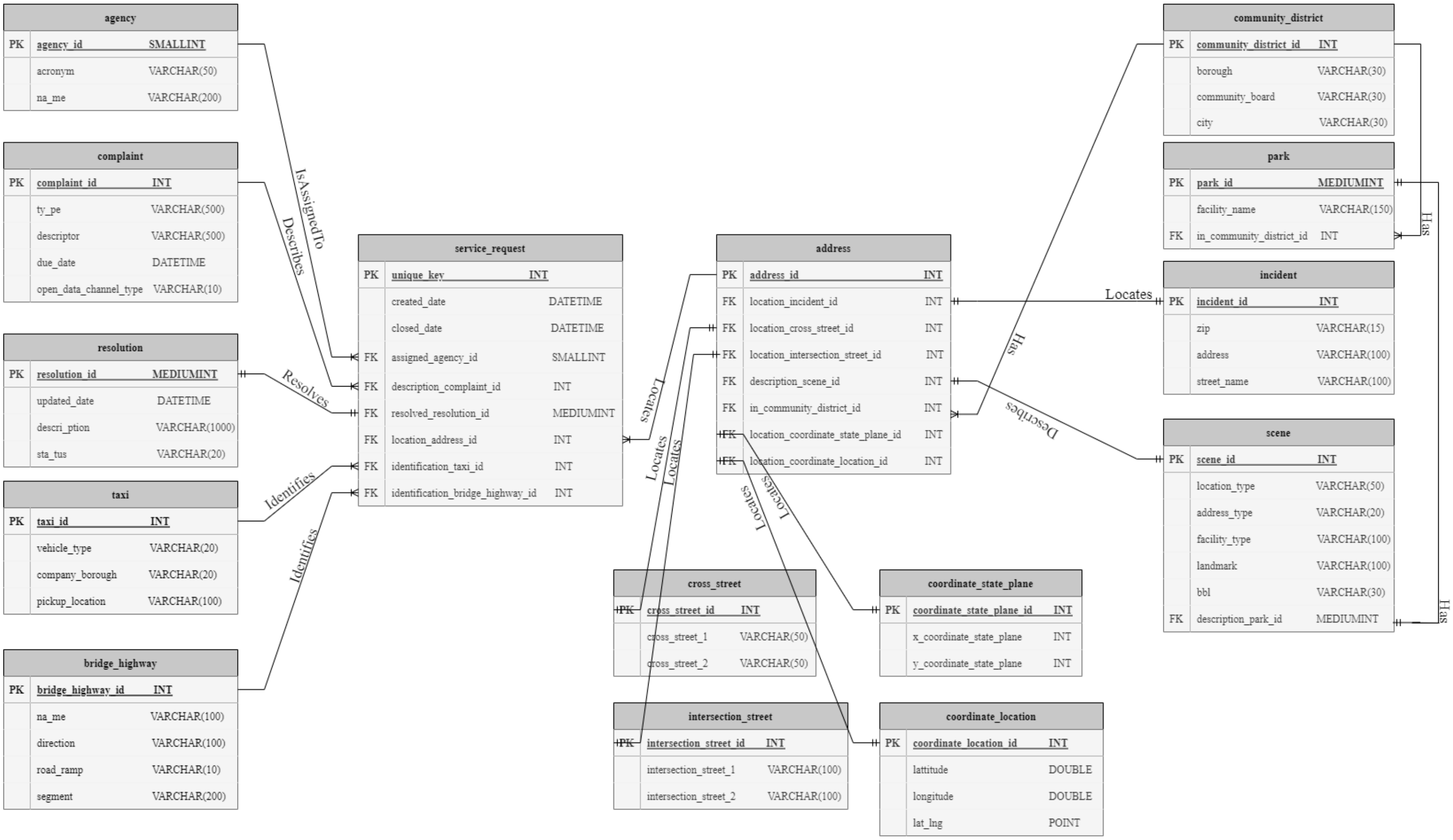
	Variable_name	Value
▶	secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

Image 1. Execution Result of: SHOW VARIABLES LIKE "secure_file_priv";



```
1  05:06:46  USE service_request_of_311                                0 row(s) affected                                                                                0.000 sec
2  05:06:46  LOAD DATA INFILE 'C:/Users/GEORGIOS/Desktop/311_Service_Requests_...  Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement  0.000 sec
```

Image 2. Error When The CSV File Is In The Wrong Folder



	agency	
PK	agency_id	SMALLINT
	acronym	VARCHAR(50)
	na_me	VARCHAR(200)

	complaint	
PK	complaint_id	INT
	ty_pe	VARCHAR(500)
	descriptor	VARCHAR(500)
	due_date	DATETIME
	open_data_channel_type	VARCHAR(10)

	resolution	
PK	resolution_id	MEDIUMINT
	updated_date	DATETIME
	descri_ption	VARCHAR(1000)
	sta_tus	VARCHAR(20)

	taxi	
PK	taxi_id	INT
	vehicle_type	VARCHAR(20)
	company_borough	VARCHAR(20)
	pickup_location	VARCHAR(100)

	bridge_highway	
PK	bridge_highway_id	INT
	na_me	VARCHAR(100)
	direction	VARCHAR(100)
	road_ramp	VARCHAR(10)
	segment	VARCHAR(200)

	service_request	
PK	unique_key	INT
	created_date	DATETIME
	closed_date	DATETIME
FK	assigned_agency_id	SMALLINT
FK	description_complaint_id	INT
FK	resolved_resolution_id	MEDIUMINT
FK	location_address_id	INT
FK	identification_taxi_id	INT
FK	identification_bridge_highway_id	INT

	address	
PK	address_id	INT
FK	location_incident_id	INT
FK	location_cross_street_id	INT
FK	location_intersection_street_id	INT
FK	description_scene_id	INT
FK	in_community_district_id	INT
FK	location_coordinate_state_plane_id	INT
FK	location_coordinate_location_id	INT

	cross_street	
PK	cross_street_id	INT
	cross_street_1	VARCHAR(50)
	cross_street_2	VARCHAR(50)

	intersection_street	
PK	intersection_street_id	INT
	intersection_street_1	VARCHAR(100)
	intersection_street_2	VARCHAR(100)

	coordinate_state_plane	
PK	coordinate_state_plane_id	INT
	x_coordinate_state_plane	INT
	y_coordinate_state_plane	INT

	coordinate_location	
PK	coordinate_location_id	INT
	lattitude	DOUBLE
	longitude	DOUBLE
	lat_lng	POINT

	community_district	
PK	community_district_id	INT
	borough	VARCHAR(30)
	community_board	VARCHAR(30)
	city	VARCHAR(30)

	park	
PK	park_id	MEDIUMINT
	facility_name	VARCHAR(150)
FK	in_community_district_id	INT

	incident	
PK	incident_id	INT
	zip	VARCHAR(15)
	address	VARCHAR(100)
	street_name	VARCHAR(100)

	scene	
PK	scene_id	INT
	location_type	VARCHAR(50)
	address_type	VARCHAR(20)
	facility_type	VARCHAR(100)
	landmark	VARCHAR(100)
	bbl	VARCHAR(30)
FK	description_park_id	MEDIUMINT

Image 3. ER-Diagram