FindHome:
Your NYC Housing Violations Analyzer

Max Wong and Team
Spring 2025

NYC OpenData

Background & Problem Definition

Background

- Housing maintenance violation in the United States affects tenants, landlords, and city officials
- Many individuals find it difficult to obtain housing violation information
  - Challenging for landlords to track outstanding violations
  - City officials need an organized way to monitor and enforce housing regulations
- Goal:
  - Create a website and simplify the violation lookup process for tenants, landlords, and city officials in **NYC area**
  - Allow users to key in location details to search for potential violations

Problem Definition

- This project is significant as it helps ensure safe living conditions for everyone including tenants, landlords, and city officials
- From this project, we hope to gain insights into violation patterns across the city which helps to improve housing standards

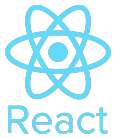
Data Source Specification

- Department of Housing Preservation & Development (HPD) Dataset (NYC opendata)
  - Size: **4.35 GB** (csv format)
  - Approximately **9.89M rows and 41 columns**
    - Unique violation ID and building ID
    - Detailed address information (house number, street, zip, apartment)
    - Dates and times of inspection or violation discovery
    - Additional descriptive information about the violation
- This dataset is useful since it provides a clear record of housing violations and offers a practical context for our topic. It has various kinds of variables (violation codes, addresses, text descriptions) that enable us to do a comprehensive analysis
- The data source is published by a reliable source, which ensures authenticity for our data management tasks

Technical Viability & Relevance

- Python will be used to create an ETL pipeline that extracts the data from the CSV file, transforms the data by handling missing values, convert data types, and filter relevant records, and finally load the transformed data into a PostgreSQL database
- Using psycopg, we define the define the table schemas and insert the data into Postgres

- Data contains structured data with rows and columns. Postgres is a RDBMS that satisfies our data requirements to maintain accurate historical records of violations
- Prioritize **consistency** and **availability** over **partition tolerance** in the **CAP Theorem**:
  - Dataset contains official records of housing code violations, which must adhere to consistency for the data to be accurate and up to date
  - NYC government and public rely on this dataset for real-time queries. For example, tenants checking violations on their buildings or landlords verifying compliance. Thus, we must prioritize availability to ensure the data remains accessible 24/7 for audits



- As a front-end Javascript library, React.js will be used to build an interaction web application that allows users to search housing violation records in different NYC neighborhoods
- Integrate React with PostgreSQL as the backend to fetch violation data

Data Cleaning

Before

```
(9936776, 20)
ViolationID            int64
BuildingID             int64
BoroID                 int64
Borough               object
HouseNumber           object
StreetName            object
StreetCode             int64
Postcode               int64
Apartment             object
Story                 object
Class                 object
InspectionDate        object
ApprovedDate          object
NOVDescription        object
CurrentStatus         object
CurrentStatusDate     object
NovType               object
ViolationStatus       object
RentImpairing         object
NTA                   object
```

After

```
(9936776, 21)
ViolationID                    Int64
BuildingID                     Int64
BoroID                         Int64
Borough              string[python]
HouseNumber          string[python]
StreetName           string[python]
StreetCode                     Int64
Postcode                       Int64
Apartment            string[python]
Story                string[python]
Class                string[python]
InspectionDate       datetime64[ns]
ApprovedDate         datetime64[ns]
NOVDescription       string[python]
CurrentStatus        string[python]
CurrentStatusDate    datetime64[ns]
NovType              string[python]
ViolationStatus      string[python]
RentImpairing                boolean
NTA                  string[python]
ClassDescription              object
```

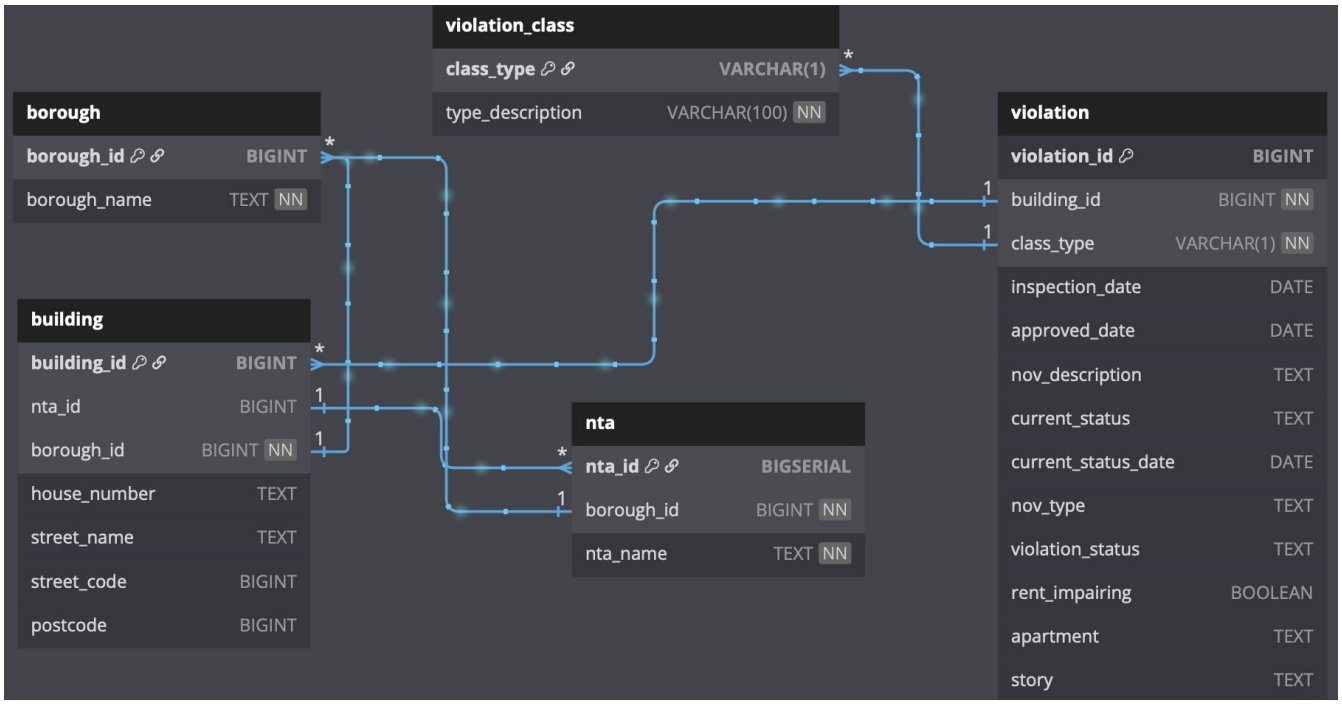
Rename Columns

```
df = df.rename(columns={
    'ViolationID'          : 'violation_id',
    'BuildingID'           : 'building_id',
    'BoroID'               : 'borough_id',
    'Class'                : 'class_type',
    'InspectionDate'       : 'inspection_date',
    'ApprovedDate'         : 'approved_date',
    'CurrentStatusDate'    : 'current_status_date',
    'NOVDescription'       : 'nov_description',
    'NovType'              : 'nov_type',
    'ViolationStatus'      : 'violation_status',
    'RentImpairing'        : 'rent_impairing',
    'NTA'                  : 'nta_name',
    'StreetCode'           : 'street_code',
    'Postcode'             : 'postcode',
    'HouseNumber'          : 'house_number',
    'StreetName'           : 'street_name',
    'Apartment'            : 'apartment',
    'Story'                : 'story',
    'CurrentStatus'        : 'current_status',
    'ClassDescription'     : 'class_description',
    'Borough'              : 'borough_name'

})
```


Database Design - Entity Relationship Diagram

- Relational database in 3rd normal form (3NF) to reduce data redundancy, improve data integrity, and maintain referential integrity



Data Table (Borough, Violation Class, NTA)

* borough_id bigint	* borough_name text
1	MANHATTAN
2	BRONX
3	BROOKLYN
4	QUEENS
5	STATEN ISLAND

* class_type varchar(1)	* type_description varchar(100)
A	Non-hazardous
B	Hazardous
C	Immediately hazardous
I	Order to repair/vacate

* nta_id bigint	* nta_name text	* borough bigint
1	Williamsbridge-Olinville	2
2	Mount Hope	2
3	Prospect Lefferts Gardens-Wingate	3
4	Belmont	2
5	Grasmere-Arrochar-South Beach-Dongan Hills	5
6	Co-op City	2
7	Queens Village	4
8	Mount Eden-Claremont (West)	2
9	Manhattanville-West Harlem	1
10	Inwood	1
11	Upper East Side-Lenox Hill-Roosevelt Island	1
12	Astoria (North)-Ditmars-Steinway	4
13	Bedford-Stuyvesant (East)	3
14	Brownsville	3
15	University Heights (South)-Morris Heights	2

Data Table (Building)

* building_id bigint	house_number text	street_name text	street_code bigint	postcode bigint	nta_id bigint	* borough bigint
1016834	66	SOUTH 6 STREET	78630	11249	96	3
1016734	2077	RYER AVENUE	26272	10457	2	2
1016638	370	54 STREET	92800	11220	204	3
1016477	373	68 STREET	99800	11220	76	3
1016342	165	MALCOLM X BOULEVARD	73190	11221	13	3
1016189	144-39	SANFORD AVENUE	61390	11355	31	4
1016184	603	4 AVENUE	54300	11215	204	3
1016156	2615	FARRAGUT ROAD	40230	11210	20	3
1016149	481	EAST 53 STREET	33713	11203	86	3
1015925	145-26	177 STREET	24940	11434	160	4
1015891	110-36	SAULTELL AVENUE	61480	11368	148	4
1015840	147-17	223 STREET	26990	11413	160	4
1015779	25-10	94 STREET	18740	11369	150	4
1015582	1124	AVENUE K	14130	11230	68	3
1015555	578	5 AVENUE	55800	11215	170	3
1015440	751	CROTONA PARK NORTH	22520	10457	46	2

Data Table (Violation)

violation_id bigint	inspection_date date	approved_date date	nov_description text	current_status text	current_status_date date	nov_type text	violation_status text	rent_impair boolean	apartment text	story text	building_id bigint	class_b varchar(1)
10081311	2013-12-30	2014-01-04	§ 27-2005 ADM CODE	VIOLATION CLOSED	2014-01-25	Original	Close	false	5	3	375411	C
10299683	2014-07-02	2014-07-07	§ 27-2013 ADM CODE	VIOLATION CLOSED	2015-08-09	Original	Close	false	1F	1	375411	A
10299685	2014-07-02	2014-07-07	§ 27-2005 ADM CODE	VIOLATION CLOSED	2015-08-09	Original	Close	false	1F	1	375411	B
10299686	2014-07-02	2014-07-07	§ 27-2005 ADM CODE	NOT COMPLIED WI	2015-08-09	Original	Open	false	1F	1	375411	B
10299690	2014-07-02	2014-07-07	§ 27-2005 ADM CODE	NOT COMPLIED WI	2015-08-09	Original	Open	false	1F	1	375411	B
10600720	2015-03-02	2015-03-02	SECTION27-2107 ADI	VIOLATION DISMIS	2015-03-16	(NULL)	Close	false	(NULL)	(NULL)	805012	I
10767483	2015-06-30	2015-07-03	§ 27-2005 ADM CODE	VIOLATION CLOSED	2016-01-06	Original	Close	false	(NULL)	1	375411	A
10767484	2015-06-30	2015-07-03	§ 27-2013 ADM CODE	VIOLATION CLOSED	2016-01-06	Original	Close	false	(NULL)	1	375411	A
10842512	2015-08-31	2015-09-01	§ 27-2005 ADM CODE	NOV SENT OUT	2015-09-02	Original	Open	false	1	1	375411	A
10013160	2013-10-21	2013-10-22	§ 27-2040 ADM CODE	VIOLATION CLOSED	2023-09-10	Original	Close	false	(NULL)	0	68995	B
10000197	2013-10-08	2013-10-12	§ 27-2005 ADM CODE	NOT COMPLIED WI	2023-11-24	Original	Open	false	(NULL)	1	74898	A
10953402	2015-10-16	2015-10-16	SECTION27-2107 ADI	VIOLATION DISMIS	2016-02-12	(NULL)	Close	false	(NULL)	(NULL)	805012	I
10001891	2013-09-30	2013-10-11	§ 27-2026 ADM CODE	VIOLATION DISMIS	2014-01-17	Original	Close	false	4C	4	291510	B
10003866	2013-10-11	2013-10-15	§ 27-2005 ADM CODE	VIOLATION CLOSED	2014-05-15	Original	Close	false	3A	1	49163	A
10003867	2013-10-11	2013-10-15	§ 27-2005 HMC:TRAC	VIOLATION CLOSED	2014-05-21	Original	Close	false	3A	1	49163	B
11130460	2016-02-24	2016-02-25	§ 27-2033 ADM CODE	VIOLATION CLOSED	2017-02-15	Original	Close	false	(NULL)	1	375411	C
11130461	2016-02-24	2016-02-25	§ 27-2104 ADM CODE	VIOLATION CLOSED	2017-02-15	Original	Close	false	(NULL)	1	375411	A
11130462	2016-02-24	2016-02-25	§ 27-2005 ADM CODE	VIOLATION CLOSED	2016-06-29	Original	Close	false	(NULL)	1	375411	C
11674017	2017-03-01	2017-03-01	§ 27-2018 ADMIN. CC	NOV SENT OUT	2017-03-02	Original	Open	false	1F	1	375411	B
11674026	2017-03-01	2017-03-01	§ 27-2005 ADM CODE	NOV CERTIFIED LAT	2017-07-01	Original	Open	false	1F	1	375411	A
10008696	2013-10-16	2013-10-17	§ 27-2005 ADMIN. CC	VIOLATION CLOSED	2014-05-22	Original	Close	false	(NULL)	0	776453	B
11674033	2017-03-01	2017-03-01	§ 27-2005 ADM CODE	NOV CERTIFIED LAT	2017-07-01	Original	Open	false	1F	1	375411	A

Data Quality

Dimension	Assessment
Accuracy	This dataset is maintained by the City of New York. It's the most accurate in terms of housing violations and actual information being reported.
Completeness	There are some null values in the data. However, the dataset contains 9+ million rows and 41 columns which provide comprehensive data
Consistency	Some columns in the dataset are stored as the wrong type, which is a clear sign of data inconsistency. This is a major issue when inserting data
Timeliness	Per the NYC HPD, the data is updated daily on NYC OpenData (latest updated on May 1, 2025)
Validity	While this dataset meets basic validity of being structured data in rows and columns, many columns' data types are improperly formatted
Uniqueness	The dataset is unique as the information comes directly from the City of New York. There are unique buildings, housing address, and street data

Technical Challenges

- Data inconsistency
  - Some data consists of special / unusual characters
- ETL process
  - Need to make sure Pandas dataframe aligns with PostgreSQL supported data types
  - Convert pandas NaN & NaT to None so PostgreSQL can read
- Database Design
  - Deciphering relationships between different tables
  - Referential integrity issues
- Inserting Data
  - Speed up runtime of inserting 9M rows of data
  - Process in-memory vs on disk

Data Governance & Licensing

Governance

- Organization
  - NYC HPD. Each violation is verified by the NYC Housing Maintenance Code or New York State Multiple Dwelling Law
- Metadata
  - Detailed information about violations and location information (41 columns)
- Data Privacy
  - NYC HPD open source data
- Business process integration
  - Business owner have to fix hazardous violation within 24 hours. Tenants have rights to view the violation history
- Master Data Integration
  - BIN (building id), detailed location (zip, borough)
- Managing the Data Lifecycle
  - HPD open source data; Inspection date (Years: 1909 to 2025)

Licensing

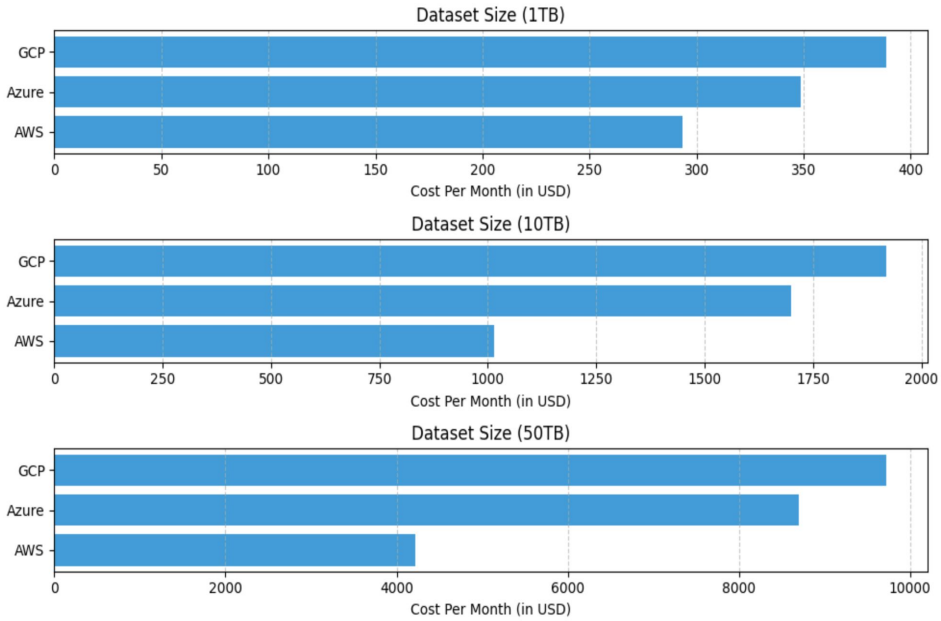
- Licensed under The Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License (CC BY-NC-ND 4.0)
- Housing violation data sourced from NYC Open Data under their terms of use

Scalability & Cost

Hardware & Software	Cost
10 core CPU (Intel i5)	~$200
32GB RAM	~$50
1TB SSD	~$75
ASUS Z790 Motherboard	~$180
300W Power Supply	~$85
Linux	Free
PostgreSQL	Free
Python	Free
Docker	Free
React.js	Free

Total: ~$590 (Hardware + Software)

Cloud Cost Estimation for 1TB, 10TB, 50TB Datasets:



- On-premise is sufficient to handle our current workload with a cost estimate of $590
- As we scale our service by entering more US markets and adding more data, we will use cloud computing services like AWS EC2 as a pay-as-you-go service to meet user demand