Chicago Crime Data Dashboard

- In this project a dashboard is created showing the 10 most dangerous blocks in Chicago according to crimes commited. An API call is made to cityofchicago.org for the data. The data is then filtered to only include the current 10 most dangerous blocks in Chicago. Finally, some data preparation is done: date columns are changed to datetime, lat and lon are changed to numeric values, and nan values are changed to type None.

- The data is then transfered to a local mysql database - avoiding duplicates according to the primary key (id). The actual script to do this is a .py file which is scheduled to run and update the mysql database daily at 5pm using Microsoft Task Scheduler.

- The Dashboard is connected to the database, and it refreshes daily at 6pm using Power BI's built in 'schedule refresh' feature.

libraries

In [1]:
```
import requests
import pandas as pd
from datetime import datetime
import time
import mysql.connector
import numpy as np
```

function to get data

In [2]:
```
today = datetime.now().strftime('%Y-%m-%dT00:00:00.000')
```

In [3]:
```
base_url = "https://data.cityofchicago.org/resource/ijzp-q8t2.json"
start_date = "2025-01-01T00:00:00.000"
end_date = today
limit = 1000

def fetch_crime_data(offset):
    url = (f"{base_url}?"
           f"$limit={limit}&"
           f"$offset={offset}&"
           f"$where=date >= '{start_date}' AND date <= '{end_date}'")

    response = requests.get(url)
    if response.status_code == 429:
        time.sleep(1800)
    elif response.status_code == 200:
        return response.json()
```

get data

In [4]:
```
all_data = []
offset = 0

while True:
    data = fetch_crime_data(offset)
    if not data:
        break
    all_data.extend(data)
    offset += limit
```

filter columns and only get 10 blocks with most amount of crimes

In [5]:
```
df = pd.DataFrame(all_data)
df = df[['id', 'date', 'primary_type',
         'location_description', 'block',
         'fbi_code', 'updated_on', 'latitude', 'longitude']]
```

In [6]:
```
top_blocks = df['block'].value_counts().nlargest(10).index
df = df[df['block'].isin(top_blocks)]
```

prepare data

In [9]:
```
# convert date columns to datetime
df['date'] = pd.to_datetime(df['date'], errors='coerce')
df['updated_on'] = pd.to_datetime(df['updated_on'], errors='coerce')

# convert latitude and longitude to float
df['latitude'] = pd.to_numeric(df['latitude'], errors='coerce')
df['longitude'] = pd.to_numeric(df['longitude'], errors='coerce')

# convert nan values to none so mysql stores them as null
df = df.replace(np.nan, None)
```

backfill into mysql database avoiding duplicates

In [10]:
```
with open('sql_password.txt', 'r') as file:
    sql_password = file.read()
```

In [11]:
```
db_config = {
    "host": "localhost",
    "user": "root",
    "password": sql_password,
    "database": "crime_data"
}
```

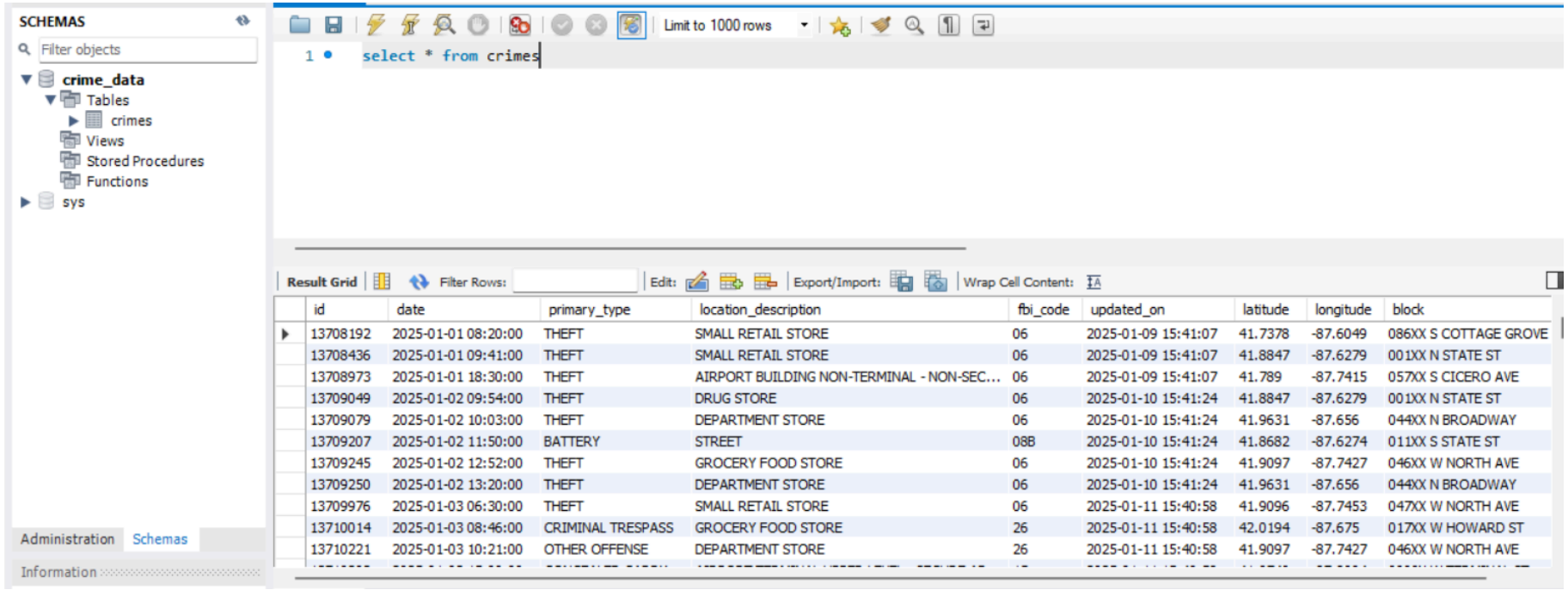
In [12]:
```
conn = mysql.connector.connect(**db_config)
cursor = conn.cursor()
```

In [13]:

```
insert_query = """
INSERT INTO crimes (id, date, primary_type, location_description, block,
                    fbi_code, updated_on, latitude, longitude)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
ON DUPLICATE KEY UPDATE
    date = VALUES(date),
    primary_type = VALUES(primary_type),
    location_description = VALUES(location_description),
    block = VALUES(block),
    fbi_code = VALUES(fbi_code),
    updated_on = VALUES(updated_on),
    latitude = VALUES(latitude),
    longitude = VALUES(longitude);
"""

for _, row in df.iterrows():
    cursor.execute(insert_query, tuple(row))

conn.commit()
cursor.close()
conn.close()

print("Data successfully inserted into MySQL")
```

```
Data successfully inserted into MySQL
```

mysql database



id	date	primary_type	location_description	fbi_code	updated_on	latitude	longitude	block
13708192	2025-01-01 08:20:00	THEFT	SMALL RETAIL STORE	06	2025-01-09 15:41:07	41.7378	-87.6049	086XX S COTTAGE GROVE
13708436	2025-01-01 09:41:00	THEFT	SMALL RETAIL STORE	06	2025-01-09 15:41:07	41.8847	-87.6279	001XX N STATE ST
13708973	2025-01-01 18:30:00	THEFT	AIRPORT BUILDING NON-TERMINAL - NON-SEC...	06	2025-01-09 15:41:07	41.789	-87.7415	057XX S CICERO AVE
13709049	2025-01-02 09:54:00	THEFT	DRUG STORE	06	2025-01-10 15:41:24	41.8847	-87.6279	001XX N STATE ST
13709079	2025-01-02 10:03:00	THEFT	DEPARTMENT STORE	06	2025-01-10 15:41:24	41.9631	-87.656	044XX N BROADWAY
13709207	2025-01-02 11:50:00	BATTERY	STREET	08B	2025-01-10 15:41:24	41.8682	-87.6274	011XX S STATE ST
13709245	2025-01-02 12:52:00	THEFT	GROCERY FOOD STORE	06	2025-01-10 15:41:24	41.9097	-87.7427	046XX W NORTH AVE
13709250	2025-01-02 13:20:00	THEFT	DEPARTMENT STORE	06	2025-01-10 15:41:24	41.9631	-87.656	044XX N BROADWAY
13709976	2025-01-03 06:30:00	THEFT	SMALL RETAIL STORE	06	2025-01-11 15:40:58	41.9096	-87.7453	047XX W NORTH AVE
13710014	2025-01-03 08:46:00	CRIMINAL TRESPASS	GROCERY FOOD STORE	26	2025-01-11 15:40:58	42.0194	-87.675	017XX W HOWARD ST
13710221	2025-01-03 10:21:00	OTHER OFFENSE	DEPARTMENT STORE	26	2025-01-11 15:40:58	41.9097	-87.7427	046XX W NORTH AVE

automate call with task scheduler (every day at 5pm)