# **Chicago Crime Data Dashboard**

- In this project a dashboard is created showing the 10 highest crime blocks in Chicago. 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 transferred 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

## function to get data

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

```
In [3]: Note that the second content is a second content in the second content is a second content in the second content is a second content in the s
```

## get data

```
In [4]: N
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

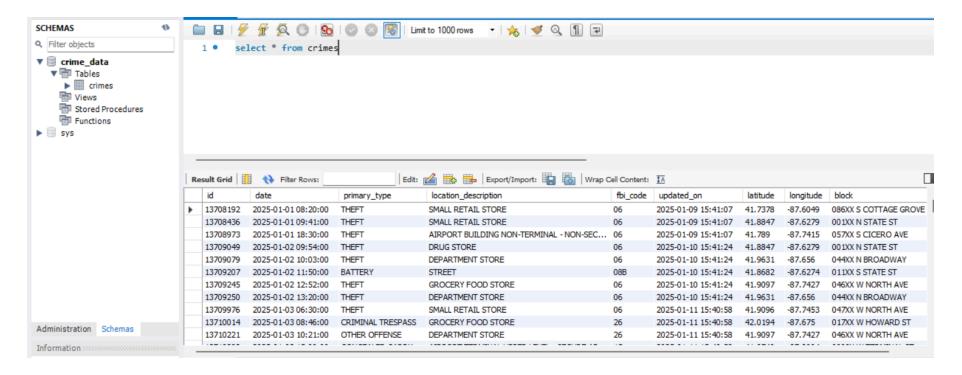
### prepare data

## backfill into mysql database avoiding duplicates

```
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)
             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



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

