Analysis of crime trends in Ontario:

A decade of statistical data (2013-2023)

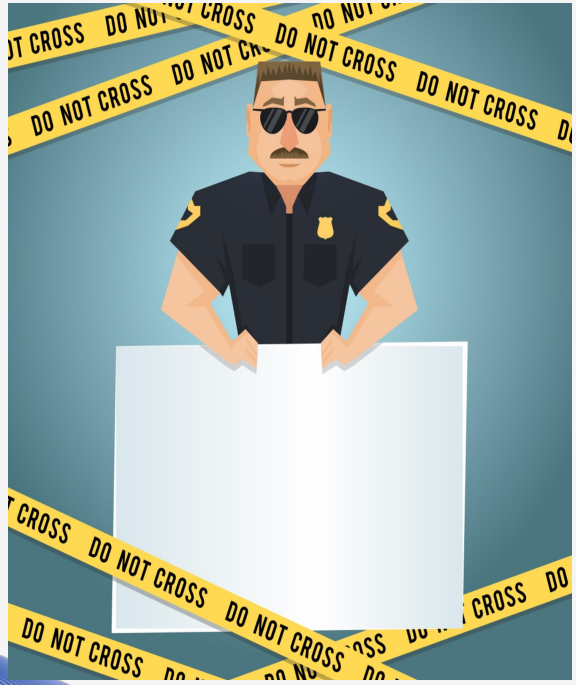
Data Engineering Track

Project 3

Group 8- Amina MAHFOUD / Gabriel OSMAN

Table of contents

01 Introduction & Project Overview



Introduction

This is a Data Engineering project that leverages on crime statistics data in the Province of Ontario, Canada to design and store relevant data into accessible databases for further analysis.

The project employs ETL workflows to process data, stores it in an SQLite database, and serves it via a Flask API. The system is designed for ease of data retrieval, filtering and analysis.

The aim of this project is to provide an accessible tool for stakeholders such as researchers, policymakers, and analysts to derive insights from historical and regional crime data.

Project overview

This is a Data Engineering project that leverages on crime statistics data in the Province of Ontario, Canada to design and store relevant data into accessible databases for further analysis.

Objectives

- Build an ETL pipeline to analyze crime statistics across different locations, time periods, and crime types in Ontario, Canada.
- Store the data in a relational SQL database (SQLite) along with an ERD with multiple tables to capture crime details, location information, details of offenders, law enforcement efforts etc.

Research Question

- How can an ETL pipeline be designed and implemented to effectively ingest, transform, and load crime data from multiple data sources using ethical practices while ensuring data integrity for various locations, violations and crimes spanning a decade?
- What is the optimal database schema and Entity-Relationship Diagram (ERD) for efficiently storing and retrieving detailed crime data, including offender profiles, location specifics, and law enforcement actions, in a PostgreSQL environment?

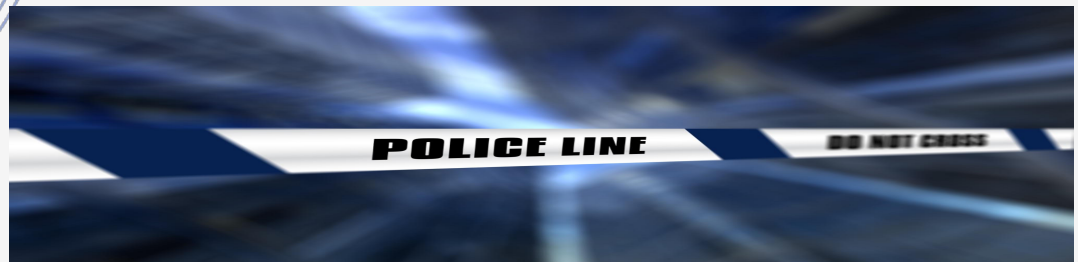
Project overview (continued)

Objectives

- The project will allow users to query and analyze crime trends, such as identifying crime hotspots, analyzing the frequency of crimes over time, and filtering crimes by severity or location.

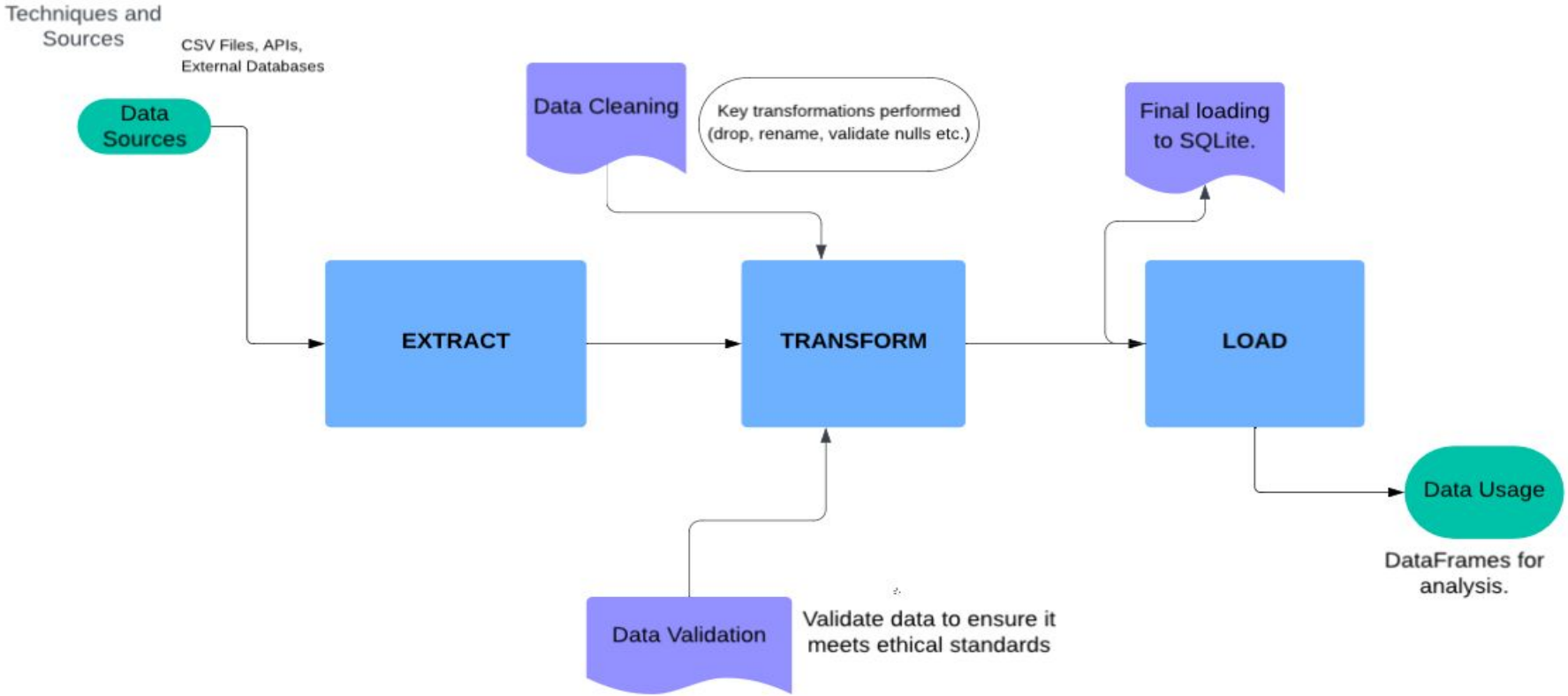
Research Question

- How can we develop user-friendly query mechanisms and analytical models within a database to enable users to identify crime hotspots, assess the temporal frequency of different crime types, and perform detailed filtering based on crime severity and geographical location?



02 ETL Workflow

ETL Pipeline Diagrams



ETL Process

The ETL process handles the data pipeline efficiently to ensure the database remains accurate and up to date. The Workflow Stages are:

A. Extract: (Techniques and Sources)
   - Use CSV files as a primary data source for crime statistics
   - Use pd.read_csv('path/to/file.csv') to read CSV files into DataFrames

B. Transform: (Processes Used)
   - Dropping Columns: Remove unnecessary columns with DataFrame.drop().
   - Renaming Columns: Use DataFrame.rename() to change column names for clarity.
   - Checking and Handling NaN Values
   - Data Cleaning and Validation: Implement data validation logic to detect and flag severe value inconsistencies.

ETL Process (continued)

C. Load:

- Load the cleaned data into the SQLite database created and managed using “DB Browser for SQLite” tool. Data is inserted into the corresponding tables (CrimeStats, HateStats, CrimeRates & location_year).
- Use SQLite3 Python library to create database connections and ensure lightweight database operations.

03 Database Design

Database Design

I. Selection of Database Type

Opted for a SQL database for the following reasons:

- Structured Data and Schema
  SQL databases use a predefined schema to structure data, ensuring that all entries adhere to a consistent format.
- Complex Querying Capabilities:
  SQL databases excel at complex querying using SQL, allowing for sophisticated data retrieval through joins, subqueries, and aggregate functions. This enables users to perform intricate data analysis and reporting.
- Support for Relational Data Models:
  SQL databases are particularly well-suited for applications with complex relationships among data entities. Their ability to handle multi-table joins and enforce relationships through foreign keys makes SQL databases advantageous for maintaining relational data integrity.

Database Design

II. Selection of SQL Database

Why SQLite Database

1. Readability

   SQLite is easily navigable, and its SQL syntax is compatible with many applications, making it user-friendly

2. Simplicity and Ease of Use

   SQLite is straightforward to integrate with applications, as it requires no intricate configurations or server management.

3. Operability:

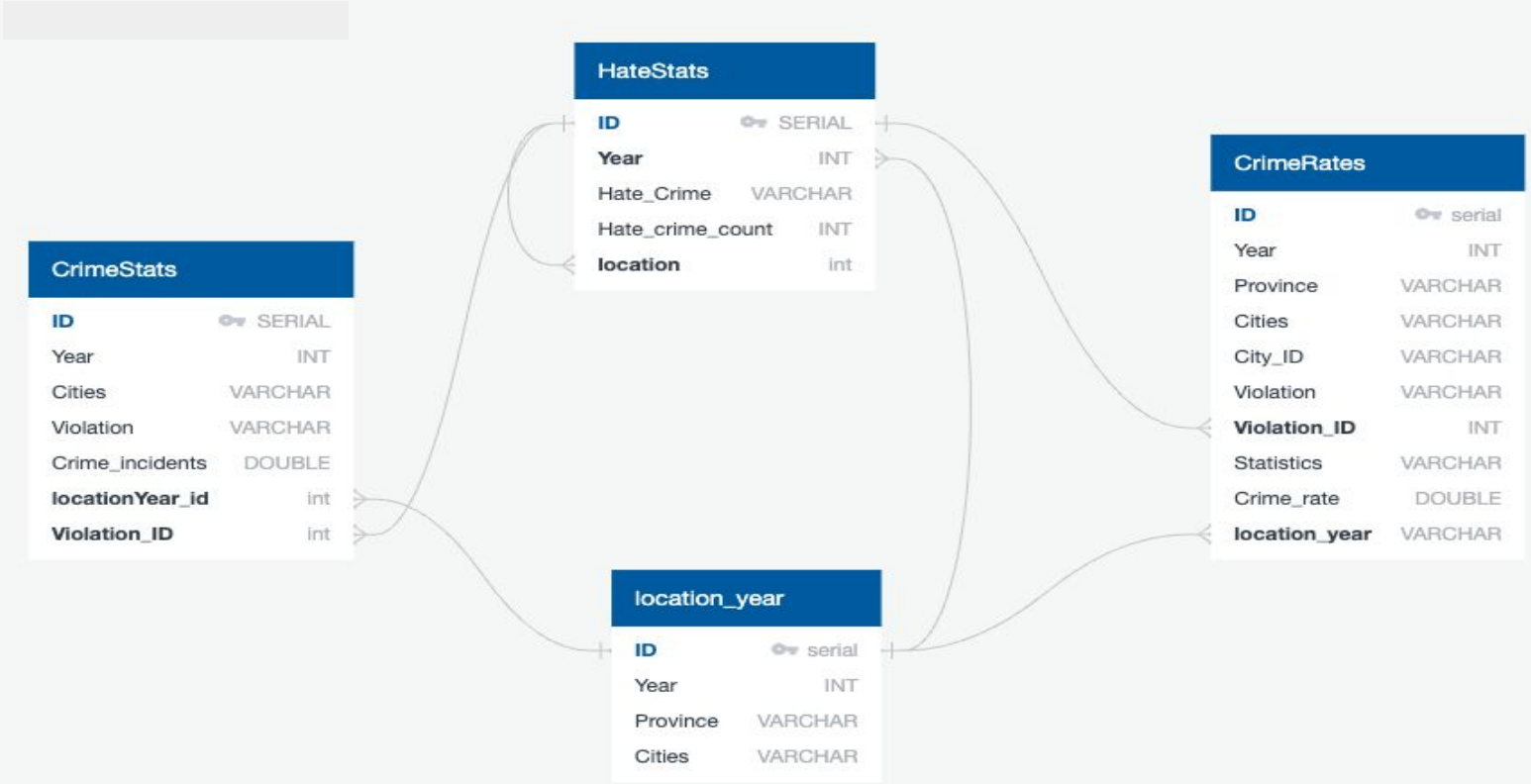
   SQLite operates seamlessly across various platforms without the need to install a separate database server. This portability allows developers to move databases between different environments easily

4. Efficiency:

   SQLite is a self-contained, serverless, and lightweight database engine. It operates directly on disk files, which minimizes overhead and leads to faster read and write operations

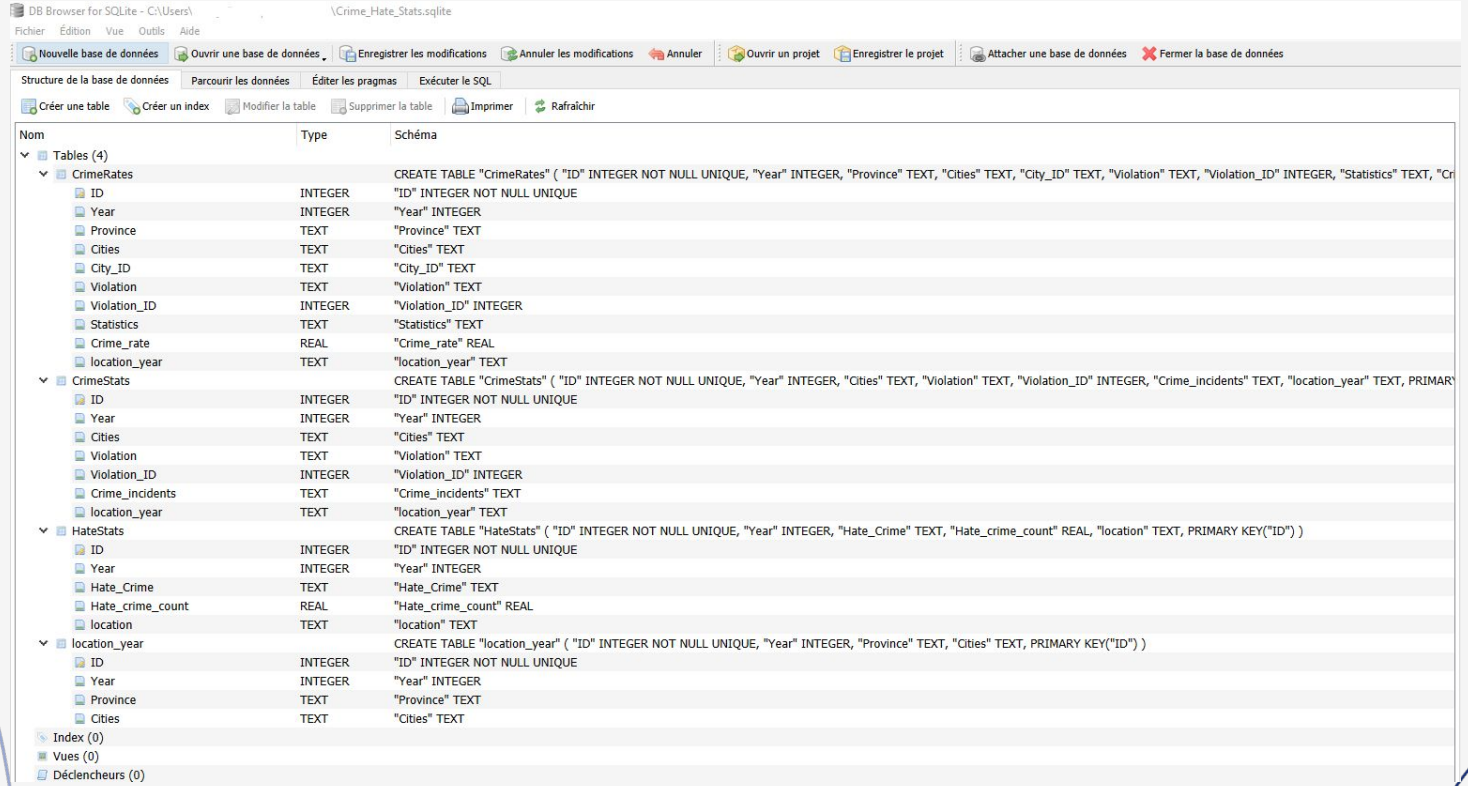
Database Design (continued)

Entity-Relationship Diagram (ERD)



Database Design (continued)

The SQLite DB



Nom	Type	Schéma
Tables (4)		
CrimeRates		CREATE TABLE "CrimeRates" ("ID" INTEGER NOT NULL UNIQUE, "Year" INTEGER, "Province" TEXT, "Cities" TEXT, "City_ID" TEXT, "Violation" TEXT, "Violation_ID" INTEGER, "Statistics" TEXT, "Cr
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
Year	INTEGER	"Year" INTEGER
Province	TEXT	"Province" TEXT
Cities	TEXT	"Cities" TEXT
City_ID	TEXT	"City_ID" TEXT
Violation	TEXT	"Violation" TEXT
Violation_ID	INTEGER	"Violation_ID" INTEGER
Statistics	TEXT	"Statistics" TEXT
Crime_rate	REAL	"Crime_rate" REAL
location_year	TEXT	"location_year" TEXT
CrimeStats		CREATE TABLE "CrimeStats" ("ID" INTEGER NOT NULL UNIQUE, "Year" INTEGER, "Cities" TEXT, "Violation" TEXT, "Violation_ID" INTEGER, "Crime_incidents" TEXT, "location_year" TEXT, PRIMAR
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
Year	INTEGER	"Year" INTEGER
Cities	TEXT	"Cities" TEXT
Violation	TEXT	"Violation" TEXT
Violation_ID	INTEGER	"Violation_ID" INTEGER
Crime_incidents	TEXT	"Crime_incidents" TEXT
location_year	TEXT	"location_year" TEXT
HateStats		CREATE TABLE "HateStats" ("ID" INTEGER NOT NULL UNIQUE, "Year" INTEGER, "Hate_Crime" TEXT, "Hate_crime_count" REAL, "location" TEXT, PRIMARY KEY("ID"))
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
Year	INTEGER	"Year" INTEGER
Hate_Crime	TEXT	"Hate_Crime" TEXT
Hate_crime_count	REAL	"Hate_crime_count" REAL
location	TEXT	"location" TEXT
location_year		CREATE TABLE "location_year" ("ID" INTEGER NOT NULL UNIQUE, "Year" INTEGER, "Province" TEXT, "Cities" TEXT, PRIMARY KEY("ID"))
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
Year	INTEGER	"Year" INTEGER
Province	TEXT	"Province" TEXT
Cities	TEXT	"Cities" TEXT
Index (0)		
Vues (0)		
Déclencheurs (0)		

04 Data Retrieval & Analysis

Combining Pandas and Flask:

Used a code that combines the data manipulation power of Pandas with Flask's web handling capabilities, allowing us to create a flexible API for querying crime and hate statistics data from an SQLite database.

Flask API with JSON output:

- **Query Parameters:** The endpoint takes query parameters, such as Hate_Crime and Cities, to filter the data.
-**Filtering Data:** Within the endpoint function, we filter the Pandas DataFrame based on the query parameters.
-**Returning JSON:** The filtered data is then converted to JSON format and returned as the API response.

Data Analysis

Opportunities for Further Exploratory Analysis:

- Trend Analysis: Users can analyze trends in hate crimes over specific years or locations, uncovering patterns and informing policy decisions.
- Comparative Studies: By allowing filtered queries, users can perform comparative analyses between different regions or time periods to identify correlations with socio-political factors.
- Dashboards and Visualizations: The retrieved data can be further processed into dashboards, facilitating real-time data visualizations and insights for stakeholders

Data Analysis (continued)

05 Technical Tool & Libraries

Languages & Libraries:

Database

Framework

Comparaison 'SQLite3' vs 'SQLAlchemy'

SQLite3

```
# Set a function to fetch data from SQLite
def fetch_data(query):
    try:
        conn = sqlite3.connect(DB_PATH)
        df = pd.read_sql_query(query, conn)
        conn.close()
        return df
```

SQLAlchemy

```
# Set a function to fetch data from SQLAlchemy
def fetch_data(query, db_url):
    try:
        engine = create_engine(db_url)
        with engine.connect() as connection:
        df = pd.read_sql_query(query,connection)
         return df
```

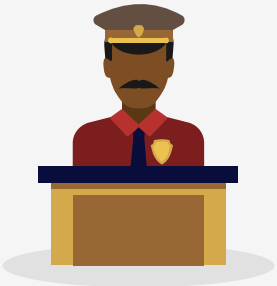
06 Ethical Considerations

Ethical Considerations

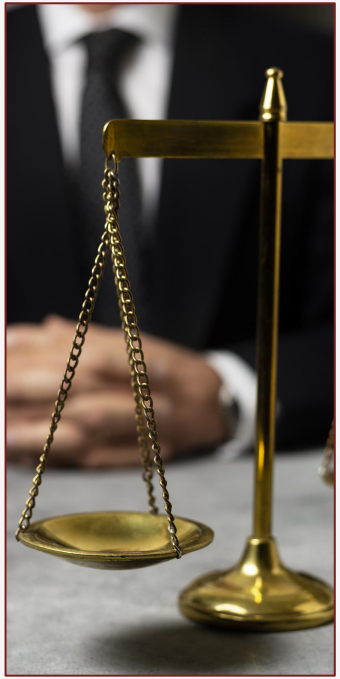
- **Minimize Bias and Misrepresentation:**
  - Ensure transformations do not mislead users or alter original meanings
  - Justify changes in data representation (e.g., aggregating data).
  - Continue to ensure integrity with data normalization

- **Ensure transparency:**
  - Maintain detailed documentation of the transformations applied.
  - Ensure users can trace back to original values if necessary.

- **Correct citation and references of source of datasets**



07 Conclusion



Conclusion

How we achieved our objectives:

- This project transforms raw crime data into actionable insights, providing user-friendly tools to analyze trends and support informed decision-making for safer Ontario communities.
- We were able to obtain raw data, modify the data using ETL, storing the data in a database (SQLite), connect to the database and retrieve using sqlite3, and instead of jsonifying data, present on the frontend using javascript for further exploration on the webpage. Typically known us **‘closing the loop’** in the Data Analytics and engineering market.

08 Documentation and references

Statistics Canada website: https://www150.statcan.gc.ca/

- Table 35-10-0066-01 Police-reported hate crime, by type of motivation, selected regions and Canada (selected police services)

- Table 35-10-0177-01 Incident-based crime statistics, by detailed violations, Canada, provinces, territories, Census Metropolitan Areas and Canadian Forces Military Police

Thank You

For Your Attention!

Any Questions