School of Business Administration Sciences

Departments of Business Administration, Economics, Accounting and Finance

Student's Name:
Stergios Separdanis

Father's Name:
Georgios

Supervisor's Name:
Evangelos Kalampokis

Interdepartmental Postgraduate Program in Information Systems

Thesis Title:
Crime in Los Angeles: Data Analysis and Trends (2020–2025)

October 2025

Acknowledgements

I would like to express my deep appreciation to my beloved parents, who have supported me emotionally, psychologically, and financially throughout my studies and my life. I truly doubt I could have made it this far without their unwavering presence by my side.

Abstract

The purpose of the thesis was to investigate the “Crime Data from 2020 to Present” dataset, obtained from the Los Angeles Open Data Portal by applying the full data lifecycle: collection → exploration → cleaning → storage → modeling → visualization → interpretation. The methodology involved using Excel for initial exploration and Power Query for cleaning the CSV, SQL Server 2022 for storing the cleaned data, SSMS for creating tables, relationships, the database schema, and SQL queries used for data import and testing. Power BI was used to construct calculated columns and measures that enabled more complex analysis with DAX, build visualizations and a dashboard, and interpret the results. The analysis revealed that the most common crime type was stolen vehicle, with serious crimes accounting for the majority of incidents. Negligent manslaughter had the highest percentage of closed cases, while pickpocketing had the lowest. The overall percentage was 20.09%, with Mission and Pacific representing the highest and lowest area-level values. Crime against child and Devonshire were the crime type and the area with the highest average reporting delay. Men were victimized more often than women, and the most affected ethnic group was Hispanic; the mean victim age was 39.4 years. Central reported the highest crime levels, with clustering observed in downtown Los Angeles. Crime peaked during afternoons, Fridays, Januaries, and the year 2022. Armed incidents represented 32.56% of all cases, mostly street-based. The findings contributed to understanding urban crime patterns and may support decision-making by authorities.

Table of Contents

1. Introduction……..…1
2. Literature Review…..……..…..…3
3. Methodology…….…..6
3.1. Dataset Search Process……...…..8
3.2. Data exploration in Excel……...….9
3.3. Data preprocessing in Power Query…….....13
3.4. Installation of SQL Server 2022……18
3.5. Installation of SSMS……..…20
3.6. Connection establishment between SQL Server and SSMS…….21
3.7. Database creation……22
3.8. Import table creation……23
3.9. Data importation into the crime_import table……...23
3.10. Creation of fact and dimension tables…….25
3.11. Data importation into fact and dimension tables……31
3.12. Creation of the ERD……...39
3.13. SQL Query Construction and Evaluation……39
3.13.1. Core Queries for Star Schema……...……..40
3.13.2. Exploratory queries for validation and testing……....43
3.14. Installation of Power BI Desktop……..53
3.15. Connecting Database to Power BI via SQL Queries……55
3.16. Creation of calculated columns and measures with DAX……..……..57
4. Data Analysis and Interpretation of Findings
4.1. Visualization of forensic data……..64
4.2. Dashboard and Interactive Analysis……..92
5. Conclusions and Suggestions
5.1. Conclusions……...…..…..95
5.2. Suggestions……...……....98
6. References……..…....101
7. Appendix…….....103

List of figures

Screenshot 1.1: “CSV on a table in Excel”…….9
Screenshot 1.2: “CSV on a table in Excel”……...……10
Screenshot 1.3: “CSV on a table in Excel”……...….10
Screenshot 1.4: “CSV on a table in Excel”……...11
Screenshot 2.1: “Applied steps in Power Query”……16
Screenshot 2.2: “Applied steps in Power Query”……17
Screenshot 3: “Database-ready version of CSV”……17
Screenshot 4: “Database-ready version of CSV in Notepad”……18
Screenshot 5: “Downloading SQL Server”……...…19
Screenshot 6: “SQL Server installed”……...19
Screenshot 7: “Downloading SSMS”…….……20
Screenshot 8: “SSMS installed”…….21
Screenshot 9: “Connection with server ready”……22
Screenshot 10: “Database created”……..22
Screenshot 11: “Table and Columns constructed”……..23
Screenshot 12: “Data imported into crime_import”……..……...24
Screenshot 13: “All data presented”……..25
Screenshot 14: “High-level ERD overview”……..27
Screenshot 15: “dimArea created”…….……...27
Screenshot 16: “dimCrime created”…….…….28
Screenshot 17: “dimPremise created”……..……28
Screenshot 18: “dimVictim created”…….……..…29
Screenshot 19: “dimWeapon created”……..……..……...…30
Screenshot 20: “factCrime created”……..……..…31
Screenshot 21: “Data imported into dimArea”……..……...32
Screenshot 22: “All area related data presented”……...32
Screenshot 23: “Data imported into dimCrime”……..……..……33
Screenshot 24: “All crime related data presented”…….33
Screenshot 25: “Data imported into dimPremise”……...……..…34
Screenshot 26: “All premise related data presented”…….34
Screenshot 27: “Data imported into dimVictim”……..……..…35
Screenshot 28: “All victim related data presented”…….……35

Screenshot 29: “Data imported into dimWeapon”……..……..…36
Screenshot 30: “All weapon related data presented”…….36
Screenshot 31: “Data imported into factCrime”……..…….38
Screenshot 32: “All factCrime’s data presented”…….38
Screenshot 33: “ERD designed and saved”…….39
Screenshot 34: “dimArea”: key fields returned”……...…40
Screenshot 35: “dimCrime”: key fields returned”……..……41
Screenshot 36: “dimPremise”: key fields returned”…….41
Screenshot 37: “dimVictim”: key fields returned”…….42
Screenshot 38: “dimWeapon”: key fields returned”……...….42
Screenshot 39: “factCrime”: key fields returned”…….43
Screenshot 40: “Incident count by crime type”……44
Screenshot 41: “Incident count by area”……..45
Screenshot 42: “Victim count by descent”……45
Screenshot 43: “Incident count by part 1 and 2”……46
Screenshot 44: “Victim count by sex and crime type”……47
Screenshot 45: “Weapon incident count by premise”…….48
Screenshot 46: “Average victim age by crime type”……...48
Screenshot 47: “Incident count by occurrence time”……..49
Screenshot 48: “Incident count by year-month”……..……..……49
Screenshot 49: “Percentage of armed incidents”……50
Screenshot 50: “Incident count per lat–lon combination”……...……51
Screenshot 51: “Average delay by crime type”……51
Screenshot 52: “Average delay by area”……..52
Screenshot 53: “Number of incidents by date of occurrence”……...……52
Screenshot 54: “Downloading Power BI Desktop”……..53
Screenshot 55: “Power BI Desktop installed”……..54
Screenshot 56: “Power BI running”……...54
Screenshot 57: “Server’s database and Power BI connected — first query’s data imported”……...……55
Screenshot 58: “Second query’s data imported”……56
Screenshot 59: “Data from all tables imported”……...56
Screenshot 60: “Model view of the star schema”……57
Screenshot 61: “Period of day column created”……..58
Screenshot 62: “Day of week column created”……59

Screenshot 63: “Hour of day column created”……....59
Screenshot 64: “Average delay measure created”…….60
Screenshot 65: “Weapon percent measure created”…….61
Screenshot 66: “Average victim age measure created”…….62
Screenshot 67: “Closed percent measure created and formatted”……..63
Screenshot 68: “Incidents by sex”……...……..65
Screenshot 69.1: “Incidents by crime type for males: Top 5”……...….66
Screenshot 69.2: “Incidents by crime type for females: Top 5”…….67
Screenshot 70: “Part 1 vs Part 2 crimes”…….68
Screenshot 71: “Incidents by crime type: Top 5”…….69
Screenshot 72: “Incidents by day period”……..…….....70
Screenshot 73: “Armed incidents by premise: Top 5”…….71
Screenshot 74: “Incidents by month”……72
Screenshot 75: “Incidents by year”……73
Screenshot 76: “Average victim age”……73
Screenshot 77: “Percentage of armed incidents”……74
Screenshot 78: “Incidents by descent: Top 5”……75
Screenshot 79: “Incidents by area: Top 5”……...76
Screenshot 80.1: “Average victim age by crime type: Top 5”……77
Screenshot 80.2: “Average victim age by crime type: Bottom 5”…….....78
Screenshot 81.1: “Number of crimes per year–month”…….……...…..79
Screenshot 81.2: “Number of crimes per year–month”…….……..…79
Screenshot 82.1: “Number of crimes per occurrence time”……..80
Screenshot 82.2: “Number of crimes per occurrence time”……..80
Screenshot 83.1: “All crime locations”……..81
Screenshot 83.2: “Crime locations in zoomed view”…….……82
Screenshot 83.3: “Crime locations with > 1000 incidents”…….……82
Screenshot 84.1: “Average reporting delay by crime type: Top 5”……83
Screenshot 84.2: “Average reporting delay by crime type: Bottom 5”…….84
Screenshot 85.1: “Average reporting delay by area: Top 5”……..85
Screenshot 85.2: “Average reporting delay by area: Bottom 5”……86
Screenshot 86: “Average reporting delay”……86
Screenshot 87: “Incidents by day of the week”……87
Screenshot 88.1: “Closed cases by area: Top 5”……...….88
Screenshot 88.2: “Closed cases by area: Bottom 5”……..89

Screenshot 89.1: “Closed cases by crime type: Top 5”……..90
Screenshot 89.2: “Closed cases by crime type: Bottom 5”……91
Screenshot 90: “Closed cases”……..91
Screenshot 91.1: “Single selection filters applied”……..92
Screenshot 91.2: “Multiple selections per slicer”…….93
Screenshot 91.3: “Cross-filtering between visuals”……....94
Screenshot 92: “SQL Server connection settings and query input in Power BI”……..…123
Screenshot 93: “Table renamed”……124
Screenshot 94: “Sex filter applied”……..……127
Screenshot 95: “Female color altered”……...128
Screenshot 96: “Chart’s values formatted”……128
Screenshot 97: “Chart’s title and page name changed”……...…...129
Screenshot 98: “Top 5 filter applied”……...……130
Screenshot 99: “Y-axis title removed”…….....130
Screenshot 100: “X-axis title changed”……...…131
Screenshot 101: “Data labels activated and formatted”……...…131
Screenshot 102: “Victim sex filter added”……..132
Screenshot 103: “Advanced filtering applied”……134
Screenshot 104: “Data sorted by the incident count”……...135
Screenshot 105: “Year’s type changed to categorical”……...…136
Screenshot 106: “Chart’s value formatted”……....137
Screenshot 107: “Category label deactivated”……..137
Screenshot 108: “Title added”…….……138
Screenshot 109: “Data format from decimal to percentage”……...…139
Screenshot 110: “Each year with its own color”……141
Screenshot 111: “Markers for months enabled”……142
Screenshot 112.1: “Gray average line added”……..143
Screenshot 112.2: “Average line’s title and value added in gray”……..143
Screenshot 113.1: “Lat & lon filter added”……...145
Screenshot 113.2: “> 1000 incidents filter added”……...……145
Screenshot 114: “Page moved to the front”……...……149
Screenshot 115: “Dashboard title added”……..150
Screenshot 116: “Description added”……150
Screenshot 117: “Visuals placed with height-width adjusted”……151

Screenshot 118: “Slicers added”……..……..152
Screenshot 119: “Filter removed”……152
Screenshot 120: “Horizontal lines added”……..……153
Screenshot 121: “Vertical lines added”……...…153
Screenshot 122: “Slicers’ categories selected”……...…..154
Screenshot 123: “Year category selected”……...…..154

1. Introduction

Data analysis plays a crucial role in both academic research and professional environments, as it allows transforming raw datasets into actionable insights. In this context, the “Crime Data from 2020 to Present” dataset was chosen for analysis because it contains over a million recent crime incidents — with attributes such as crime type, time, location, and victim demographics — recorded in Los Angeles, one of the largest and most well-known U.S. cities, and the second most populous in the country. LA is characterized by its cultural diversity, sprawling metropolitan area, and long-standing crime challenges, which render it an excellent case study for forensic data analysis. Unlike most studies that rely on business-related datasets (such as sales or financial data), this project explores a real-world public safety dataset that enables the investigation of spatio-temporal crime patterns and victim demographics, providing both academic insight and practical relevance for law enforcement and public policy.

The objective of this project is to investigate the aforementioned real-world dataset, obtained from the official source of the Los Angeles Open Data Portal by applying the full data lifecycle: collection → exploration → cleaning → storage → modeling → visualization → interpretation. In the Methodology chapter, the technique chosen for this project is presented, along with its tools, advantages, and limitations, and it is compared with alternative approaches.

Starting from the “Dataset Search Process” (collection) section, in which we identify and select a suitable dataset through Google Dataset Search, we end up with the crime dataset eventually chosen for this thesis, in CSV format. The “Data Exploration in Excel” (exploration) section follows, where the various columns of the CSV file are examined. Next, the “Data preprocessing in Power Query” (cleaning) section is presented, where the data types and formats of several columns are adjusted to ensure compatibility with SQL Server.

After installing SQL Server 2022 and SQL Server Management Studio (SSMS),

and establishing their connection, the sections “Database Creation”, “Import table creation”, “Data importation into the `crime_import` table”, “Creation of fact and dimension tables”, and “Data importation into fact and dimension tables” are carried out, all of which correspond to the “Storage” step of the data lifecycle. Here, a database schema is designed, tables with their respective columns are created, and relationships are defined, ensuring the CSV data are properly structured for storage.

Following this, we present the “SQL Query Construction and Evaluation” (modeling) section, which consists of two subsections: “Core Queries for Star Schema” and “Exploratory queries for validation and testing”. The first subsection focuses on queries used for importing data into Power BI, while the second includes queries exclusively utilized for validation and testing during the development process.

Once the construction and testing of queries comes to an end, the following sections are elaborated: “Installation of Power BI Desktop”, “Connecting Database to Power BI via SQL Queries”, and the “Creation of calculated columns with DAX” (modeling). In the aforementioned sections, we proceed with the installation of the business intelligence platform, connect the database by importing the core queries for the star schema, and create calculated columns and measures essential for analysis using the Data Analysis Expressions (DAX) language.

Finally, the “Visualization of forensic data” and the “Dashboard Creation and Interactive Analysis” sections take place, both of which belong to the “Visualization” and “Interpretation” steps of the data lifecycle. The first demonstrates the production of various visualizations (e.g., clustered column and line charts) based on the imported queries, and the second includes the creation of an interactive dashboard, along with the interpretation of their findings. The thesis concludes with a discussion of the findings, their limitation, and potential directions for future research.

2. Literature Review

Criminal data analysis plays a crucial role for law enforcement, policy-making, and sociological research, since it enables the identification of crime trends, hotspots and victimization patterns. There are quite a few approaches used in criminal data analysis, such as statistical methods, business intelligence (BI) tools, and machine learning.

The first study relevant to the current thesis was presented by a UCLA-led team of scholars and law enforcement officials, as reported by Wolpert (2015, October 7). The researchers developed a mathematical model to guide where the LAPD should deploy its units, with the purpose of decreasing the number of occurring crimes. The model was designed using 10 years of crime data and six years of mathematical research, and it is able to predict the time and place of serious crimes based on historical patterns in a given area. According to the report, the model managed to predict and prevent twice as many crime incidents as crime analysts. However, it was also criticized by some social scientists, who argued that human and criminal behavior is too complex to be fully captured by such a mathematical model (Wolpert, 2015, October 7).

In contrast to the UCLA study, which focused on mathematical modeling and predictive policing, Jenga et al. (2023) offered a broader literature-based perspective through a systematic review. It evaluated 68 different crime prediction techniques developed between January 2020 and August 2022. It provided a better understanding of machine learning algorithms used in crime analysis and prediction, highlighted their limitations, and outlined possible directions for future research in the field. Its purpose was to gather and synthesize the existing knowledge on machine learning-based crime prediction through the well-established methodology of a Systematic Literature Review, and to assist scientists and law enforcement agencies in mitigating and preventing future crime occurrences. This study — which was the most up-to-date review on the use of machine learning in crime prediction within the aforementioned period — found that crime is influenced by numerous internal and external factors, and that crime prediction can play a crucial role in

improving communities and economies worldwide by supporting governments, police, and law enforcement agencies. However, a large number of articles were discarded during the exclusion phase of the review, which could introduce bias in the definition of irrelevance. Moreover, the study was restricted to journal articles and reviews, leaving out conference papers, technical reports, datasets, and practical projects. Another limitation was that it did not include recent scientific work published after 2022 (Jenga et al., 2023).

While Jenga et al. (2023) concentrated on a narrower set of recent studies, Mandalapu et al. (2023) offered a more extensive review, analyzing more than 150 papers on machine and deep learning approaches for crime prediction. Their study provided access to the datasets used in prior research and examined notable techniques for predicting crime, offering insights into different factors and trends associated with criminal activities. It also highlighted potential limitations and proposes future directions that could enhance the accuracy of crime prediction. One of the key strengths of this study was its comprehensiveness, which made it valuable for both researchers and law enforcement authorities. Another important contribution was the collection and archiving of publicly available datasets for future scholars. However, despite the advancements in the field, there remained a lack of consolidated knowledge on how these technologies can be effectively utilized for solving the problem of crime prediction. In addition, crime data were often difficult to collect, incomplete, or unreliable, while privacy and ethical concerns were also raised (Mandalapu et al., 2023).

In contrast to the study carried out by Mandalapu et al. (2023), which analyzed numerous crime prediction machine and deep learning techniques, Fisk et al. (2025) focused on improving predictive crime models by extending the mathematical framework of Artificial Neural Networks (ANN) tailored to general spatiotemporal problems and appropriately applying them. The authors formulated a semi-analytical approach to solving Geographically and Temporally Weighted Regression (GTWR) and applied it to London crime data. They extended the Geographically and Temporally Weighted Neural Network (GTWNN) framework with three mathematical enhancements, which are subsequently applied to both the London and Detroit datasets. The approach

achieved high predictive accuracy, validating the underlying assumptions and approximations. The GTWNN framework introduced mathematical advances, enabling more flexible and accurate modeling of spatiotemporal crime data. However, the study noted lack of continuity along spatial and temporal axes and limited incorporation of historical contextual information (Fisk et al., 2025).

Overall, the reviewed literature highlighted the wide range of approaches that had been applied to crime prediction, ranging from mathematical modeling and predictive policing to systematic literature reviews of machine and deep learning algorithms, and more advanced frameworks such as deep learning applied to spatiotemporal data. While these studies demonstrated the potential of predictive models in supporting law enforcement and shaping policy decisions, they also revealed meaningful constraints, including methodological complexity, data availability, lack of focus on recent or highly localized datasets, and ethical concerns. In this context, the present thesis aims to fill part of this gap by implementing a complete data lifecycle approach — from collection, exploration, cleaning, and storage to modeling, visualization, and interpretation — on the official Los Angeles crime dataset (2020–present). Unlike much of the existing literature, which is often limited to specific case studies, highly theoretical, or restricted to specific methodological comparisons, this work emphasizes a practical, transparent, and reproducible workflow using accessible business intelligence tools, with the goal of extracting actionable insights for crime analysis.

3. Methodology

The subject of the project was investigated with the use of a number of tools.

- **Excel:** for the initial exploration of the dataset and the cleaning of the CSV with Power Query
- **SQL Server 2022:** for the storage of the cleaned CSV.
- **SSMS:** for the creation of tables, relationships between them, the database schema, and execution of SQL queries — both core and exploratory — that organize and filter data.
- **Power BI:** for the construction of calculated columns and measures that allow for more complex analysis with DAX, for building visualizations and an interactive dashboard, and interpreting the results (this will be analyzed more thoroughly in the chapter of Data Analysis and Interpretation of Findings).

The advantages of the technique we selected for this project are the following:

- Covers all data cycle: collection → exploration → cleaning → storage → modeling → visualization → interpretation.
- Well-known tools in professional and academic environments, each serving a distinct role in the data lifecycle.
- Preserves the star schema, ensuring clarity, maintainability, and modularity in Power BI exactly as in SQL Server.
- Allows management of large datasets via SQL Server.
- Offers advanced analytic capabilities with DAX.
- Enables thematic analysis through modular SQL queries.
- Supports reproducible workflows through structured query design and transparent data transformations.

However, this technique also has some disadvantages:

- Requires knowledge of multiple tools.
- Not easily automated; requires manual execution of steps across different platforms.
- Limited scalability in real-time scenarios without automation/scripting.

The approach that was eventually selected is not the only one available. There are numerous techniques commonly used by researchers. The first approach we will discuss involves the use of the Python computer programming language and its libraries, such as pandas, matplotlib, seaborn, and plotly. Its advantages are:

- Vast flexibility and automation capabilities.
- Ideal for large datasets and advanced analytics.
- Large community and many libraries available for data science.

Nevertheless, this approach also has some disadvantages:

- Requires programming skills.
- Longer learning curve compared to GUI-based tools.
- Not an “all-in-one” solution like Power BI for visualizations and dashboards without an additional setup.

The second approach we will discuss involves the use of the R programming language and its packages, such as tidyverse, ggplot2, dplyr. Its advantages are:

- Strong libraries for data cleaning, transformation, and statistical analysis.
- Good for reproducible research with RMarkdown.
- Widely used in academia and statistical research.
- High-quality visualizations with ggplot2.

Nevertheless, this approach comes with certain limitations:

- Requires programming knowledge
- Steep learning curve compared to GUI-based tools.
- Not as strong for interactive dashboards as Power BI or Tableau.
- Smaller community compared to Python in data science.

The third approach we will discuss involves the use of Tableau with SQL or Excel for data preprocessing. Its advantages are:

- Fast creation of reports with drag-and-drop interface.
- Excellent for interactive dashboards and professional visualizations.
- Can connect easily to multiple data sources (SQL, Excel, CSV, etc.).
- Popular in business environments; strong industry adoption.

However, this approach also has certain drawbacks, namely:

- License costs (not free, unlike Power BI Desktop).
- Less flexibility for advanced analytics compared to DAX or programming languages.
- Limited ETL (cleaning and transformation) capabilities compared to Power Query or Python.

The advantages and disadvantages of the chosen technique explain why it was ultimately selected, despite its limitations. The methodology enabled thematic exploration through fourteen targeted SQL queries, which were used exclusively for validation and testing during development. For the Power BI integration, six dedicated queries were generated — one for the fact table and five for the dimension tables — ensuring that the star schema and the existing relationships were preserved intact within the BI model. This design choice allowed for cohesion and reproducibility across the workflow, while directly supporting the creation of interactive dashboards that reflect the original database schema. Moreover, this approach was selected because it demonstrates proficiency in multiple tools.

3.1. Dataset Search Process:

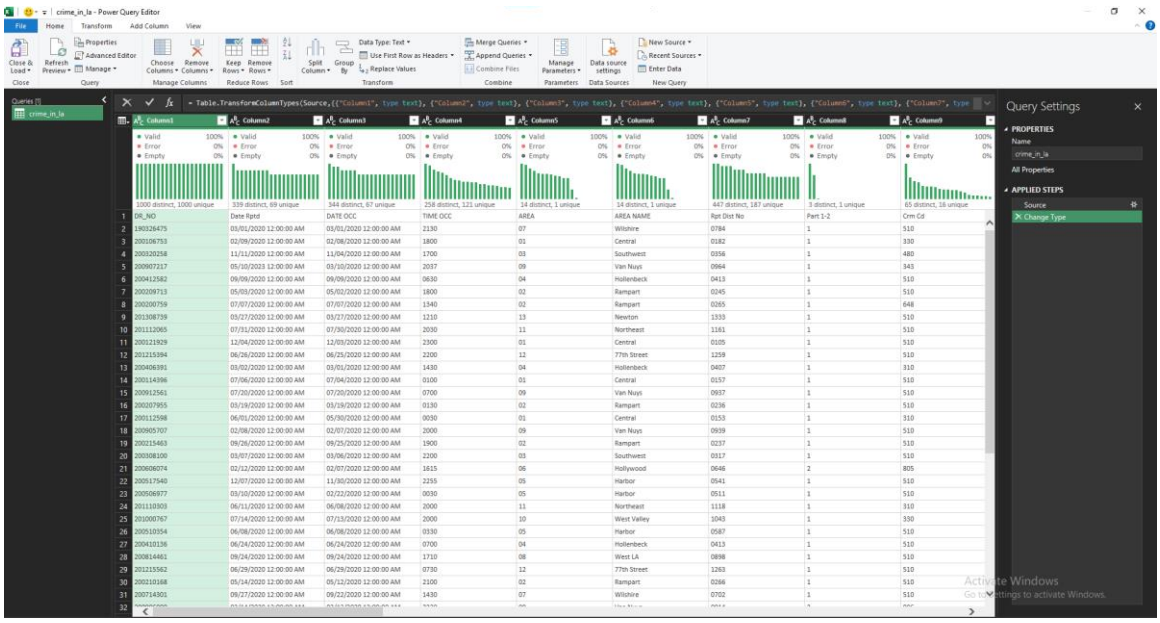
Initially, I searched for crime datasets on Kaggle and immediately noticed that there were plenty of crime dataset categories to choose from, including General Crime Incidents, Global Terrorism, Gun Violence, Homicide Reports, Mass Shootings, Police Violence, Hate Crimes, Crimes Against Women etc that were relevant to different countries and cities across the world, such as Switzerland, France, India, Brazil, South Africa, South Australia, Chicago, Denver, Cambridge, Philadelphia, Portland, Montreal, Boston, London, New York City, and San Francisco.

After days of research, I located a preprocessed version of the raw data that were used for this project, but I ended up discarding it, even though it was classified as Gold with a usability of 10 out of 10 points and had numerous votes on Kaggle. The reason behind this decision is that the raw dataset, which I located at the Los Angeles Open Data website with the help of Google Dataset

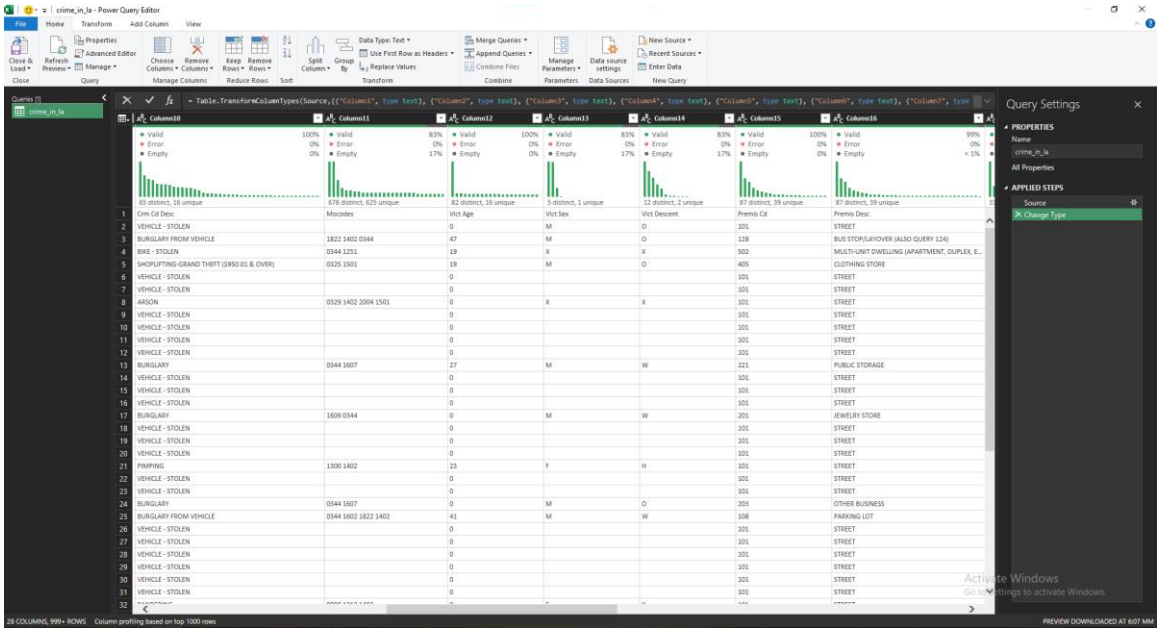
Search, had recently been updated, containing approximately 42.000 additional incidents. Having more rows was a vast advantage compared to the preprocessed dataset. However, the raw dataset had its disadvantages too. Some of its columns needed cleaning, others needed formatting in order to yield meaningful inferences.

3.2. Data exploration in Excel:

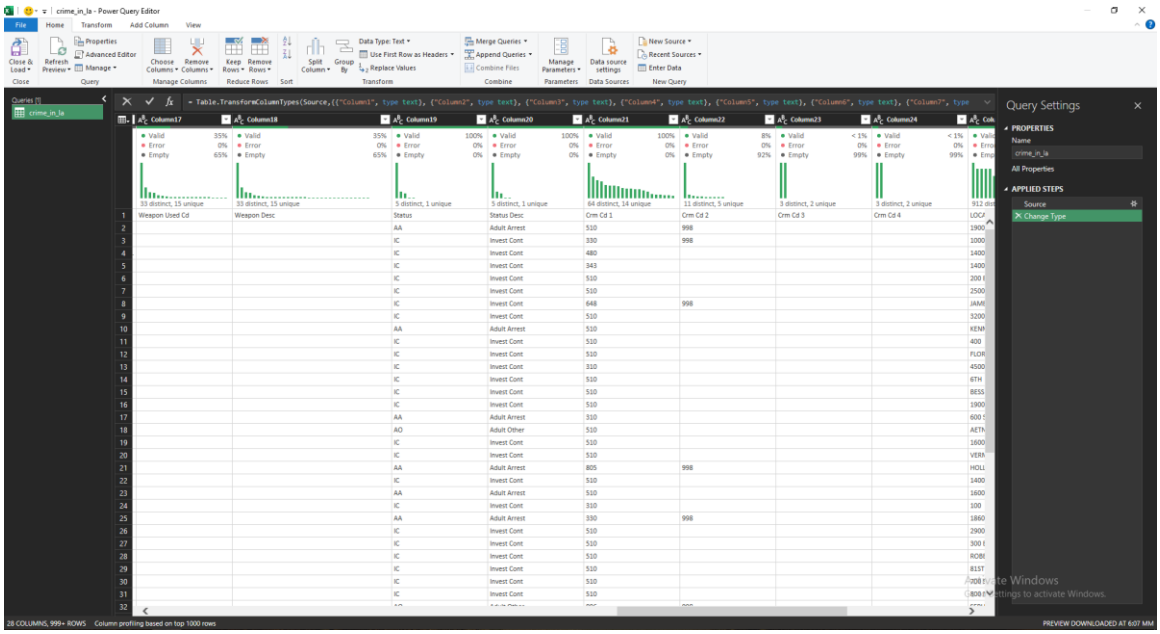
Right after we downloaded the dataset in simple CSV format and assigned it a name of our liking (crime_in_la_raw in our case), we proceeded to explore it. By carrying out the steps shown in Appendix 1, we ended up in Power Query with a table that included all columns and rows of the CSV file.



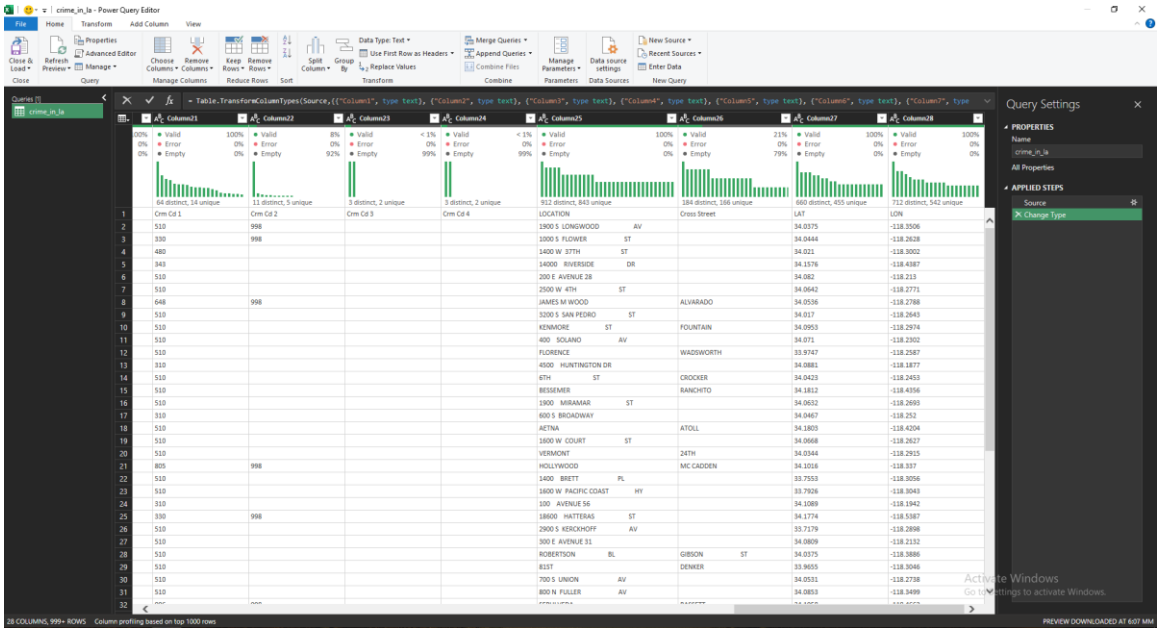
Screenshot 1.1: “CSV on a table in Excel”



Screenshot 1.2: “CSV on a table in Excel”



Screenshot 1.3: “CSV on a table in Excel”



Screenshot 1.4: “CSV on a table in Excel”

As we can see, we have columns that were critical to our project and ready to be used:

- `AREA NAME` (name of the LAPD division).
- `Crm Cd Desc` (crime description).
- `Premis Desc` (type of premises description).
- `Weapon Desc` (weapon used description),
- `Status Desc` (case status description).

We also have columns that were important but needed data cleaning/formatting to generate analytical value:

- `Date Rptd` (date the crime was reported) had the wrong data type, and the time component needed to be removed.
- `DATE OCC` (date the crime occurred) had the wrong data type, and the time component needed to be removed.
- `TIME OCC` (time of occurrence) had the wrong data type and required formatting.
- `Part 1-2` (part 1 stands for serious crimes like rape, robbery, homicide while part 2 encompasses less serious crimes like vandalism, and trespassing) had the wrong data type.
- `Vict Age` (victim’s age) had the wrong data type and needed cleaning.
- `Vict Sex` (victim’s sex) needed cleaning.

- `Vict Descent` (victim's race) needed cleaning.
- `LAT` (geographical latitude of the incident) had the wrong data type.
- `LON` (geographical longitude of the incident) had the wrong data type.

However, we also observed that several columns were not directly useful for our analysis and often contained incorrect data types. Nevertheless, they were preserved to maintain normalization, enable easier combinations through JOINs and subqueries, reduce repetitive data, and allow future expansion (e.g., adding new weapons). These are:

- `DR_NO` (unique crime report number) had the wrong data type but serves as a unique identifier essential for data integrity and relationships.
- `AREA` (numeric code identifying the LAPD geographical division) had the wrong data type — `AREA NAME` conveys the same information more clearly.
- `Rpt Dist No` (reporting district number) already exists in text format in `Premis Des` (premise description) column and had the wrong data type.
- `Crm Cd` (numeric code for the committed crime) is considered superfluous, as Crm Cd Desc provides richer and more interpretable information. It also had the wrong data type.
- `Mocodes` (codes that describe how the crime was committed) contains specialized codes which are hard to interpret without documentation.
- `Premis Cd` (premise code) is needless — `Premis Desc` contains all relevant info. It also had the wrong data type.
- `Crm Cd 1, 2, 3 & 4` (sub-crime codes with 1 being the most serious crime and 4 the least serious crime) are frequently vacant, we already have `Crm Cd Desc` in our possession. It also had the wrong data type.
- `Weapon Used Cd` (weapon used code) had the wrong data type and can be interpreted easier with `Weapon Desc`.
- `Status` (code of case status) can be interpreted easier with Status Desc. It also had the wrong data type.
- `LOCATION` (nearest block where the crime occurred) is mostly void and redundant due to the presence of `LAT` and `LON`.
- `Cross Street` is mostly vacant and less useful than `LAT` and `LON`.

3.3. Data preprocessing in Power Query:

Changing the formatting of `Date Rptd`, `DATE OCC`, `TIME OCC` was the first step performed with the assistance of Power Query. Starting with the transformation of `Date Rptd`, the steps located at Appendix 2 were followed, in which:

- 03/01/2020 12:00:00 AM is transformed into 03/01/2020 and we do not need to see time since all its values are 12 AM.
- `Date.ToText` converts the `Date Rptd` column from a date type to text string formatted in ISO standard (yyyy-MM-dd). Using this format ensures consistency between the date values in the CSV file and the SQL Server database, avoiding potential misinterpretation due to different locale or date format settings.
- the resulting column becomes text, so that when we export the CSV and import it into the database, no errors related to date types will occur.

While in the Power Query Editor, steps three to ten — provided in Appendix 2 — were followed to remove the time component and adjust the format of `DATE OCC`, using the code presented in Appendix 3.

The next stage involved updating the `TIME OCC` as well, by carrying out the detailed steps provided in Appendix 4, in which:

- both first and second `Number.FromText` take `TIME OCC`'s text (the column has text type as default) and convert it into number, for example, "2130" becomes 2130.
- `Number.IntegerDivide` divides second's `Number.FromText` values by 100 and returns the integer portion of the result, for example, 1430 / 100 = 14.
- first `Text.From` creates a text value from `Number.IntegerDivide`'s result, for example, 21(integer) becomes "21" (string).
- first `Text.PadStart` returns text with a length of two characters by padding the start of `Text.From`'s value, for example, "6" turns into "06" while "12" remains as is.
- & combines the first `Text.PadStart`'s result (hours e.g., 15) with the separator ":" and the second `Text.PadStart`'s result (minutes e.g., 38).

- `Number.Mod` divides second's `Number.FromText` values by 100 and returns the remainder, for example, 0345 / 100 = 45.
- second `Text.From` creates a text value from `Number.Mod`'s result, for example, 50 (integer) becomes "50" (string).
- second `Text.PadStart` returns text with a length of two characters by padding the start of `Text.From`'s value, for example, "2" turns into "02" while "22" remains as is.
- the end product is time in the form of "21:30", which allows us to import `TIME OCC`'s values into the database without any issues.
- the resulting column becomes text, so that when we export the CSV and import it into the database, no errors related to time types will occur.

Right after the formatting of `Date Rptd`, `DATE OCC` and `TIME OCC` was completed, we proceeded into cleaning and formatting the three victim related columns, `Vict Age`, `Vict Sex`, and `Vict Descent`. Starting with `Vict Age`, the steps provided in Appendix 5 were followed, in which we kept the age range of 0–90 as valid, since life expectancy at birth in the US is 78.4 years and rarely exceeds 90 (Centers for Disease Control and Prevention, 2025). Ages outside this range (with negative values, 99, 120) were dealt with as unknown, taking the value of 0, so that we could maintain a more consistent and realistic analysis.

Next, it was time to clean the second victim related column, named `Vict Sex`, with the steps shown in Appendix 6, in which all values that were neither F (female) nor M (male) were converted to X (unknown sex). This was done because the dataset contained sex values like "H", "–", blanks, and converting them to X ensured greater consistency in our project.

The last victim related column that we cleaned was `Vict Descent`. To do that, the steps depicted in Appendix 7 were carried out, in which the values of victim's descent that were either equal to "–" or blank were eliminated and an X was put in their place. This means that we kept only the real values, such as:

- A (Other Asian),
- B (Black),

- C (Chinese),
- D (Cambodian),
- F (Filipino),
- G (Guamanian),
- H (Hispanic/Latin/Mexican),
- I (American Indian/Alaskan Native),
- J (Japanese),
- K (Korean),
- L (Laotian),
- O (Other),
- P (Pacific Islander),
- S (Samoan),
- U (Hawaiian),
- V (Vietnamese),
- W (White),
- Z (Asian Indian),

and the rest of values were converted to X (unknown).

Our next task was to change the data type of the columns that had the wrong one by following the step shown in Appendix 8. The columns were the following:

- `DR_NO`,
- `AREA`,
- `Rpt Dist No`,
- `Part 1-2`,
- `Crm Cd`,
- `Premis Cd`,
- `Weapon Used Cd`,
- `Crm Cd 1, 2, 3 & 4`.

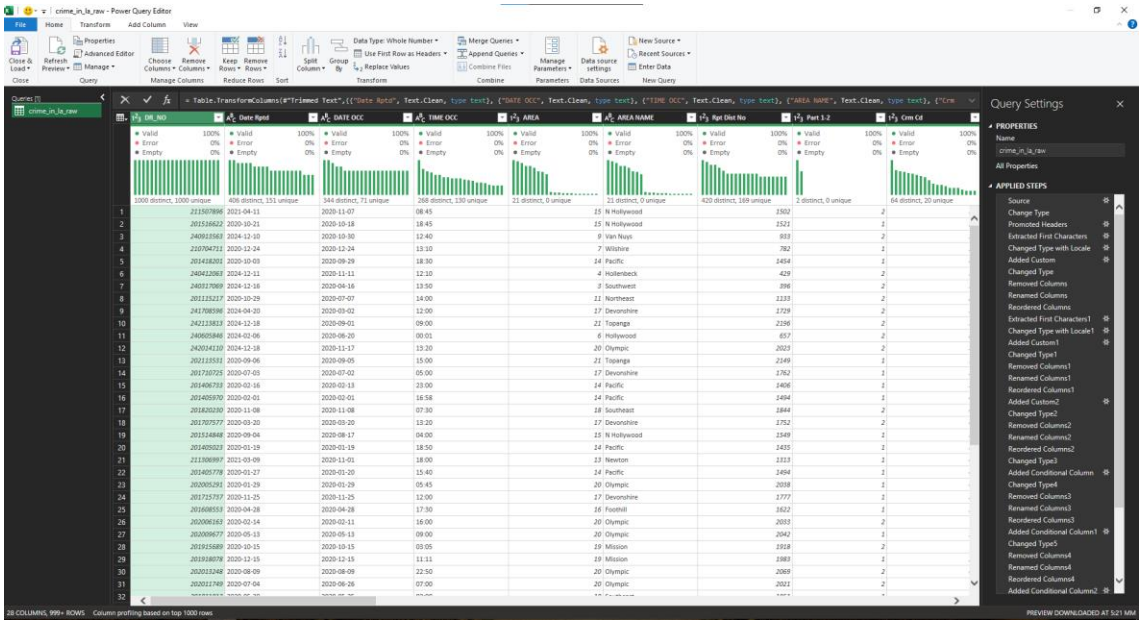
We mentioned a while ago that `Date Rptd`, `DATE OCC`, `TIME OCC`, `LAT`, and `LON` had wrong data types but if we endeavored to change them to “Date”, “Date”, “Time”, “Decimal Number”, and “Decimal Number” types, respectively, columns values could have resulted in errors and we wanted to prevent that from happening. For example, if we changed the data type of `Date Rptd` from “Text” to “Date”, its values would have generated an error due to the fact that “2020-

03-01” couldn’t have been converted into 2020/03/01. After all, it was not essential either, since they already had the proper format and thus, were ready to be imported into the database table without any problems.

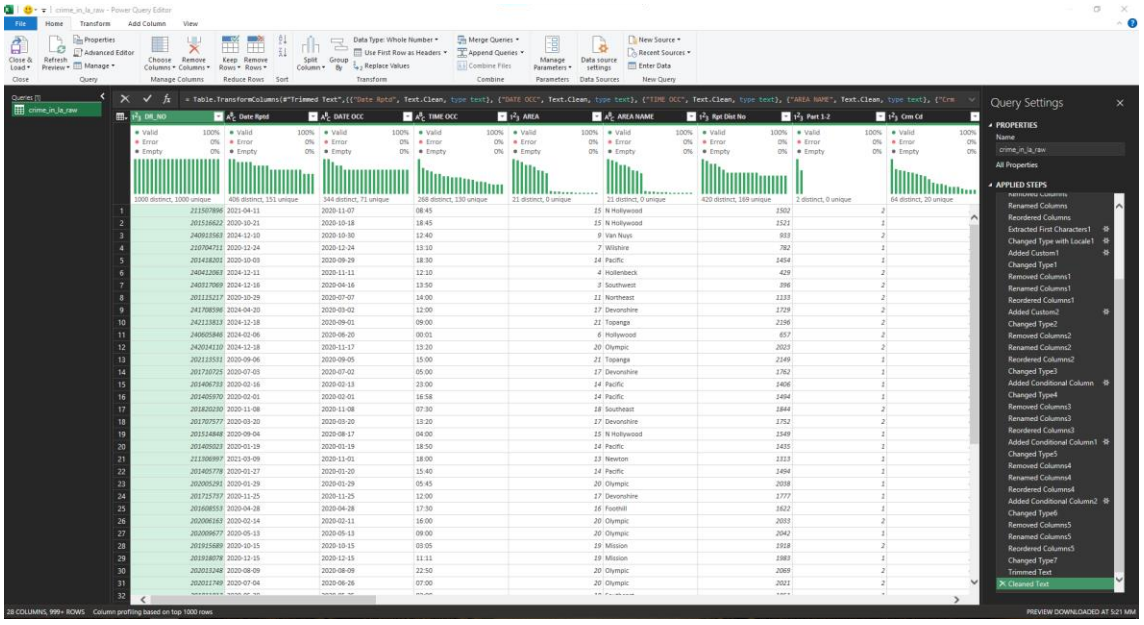
It was time for the execution of the last few essential steps that gave our CSV its final, preprocessed form. This form was intended to be imported into the database that we would create soon. The steps are located at Appendix 9, in which:

- Needless spaces before and after text that could cause problems in SQL groupings or lookups are eliminated e.g., “ Wilshire ” ≠ “Wilshire”.
- Non-printable characters are erased e.g., from copy-paste Excel, hidden tabs.
- Excel is set to write dot and not comma in decimals just for the current session. This can be helpful if your locale is different than “english (United States)” e.g., “greek (Greece)”.

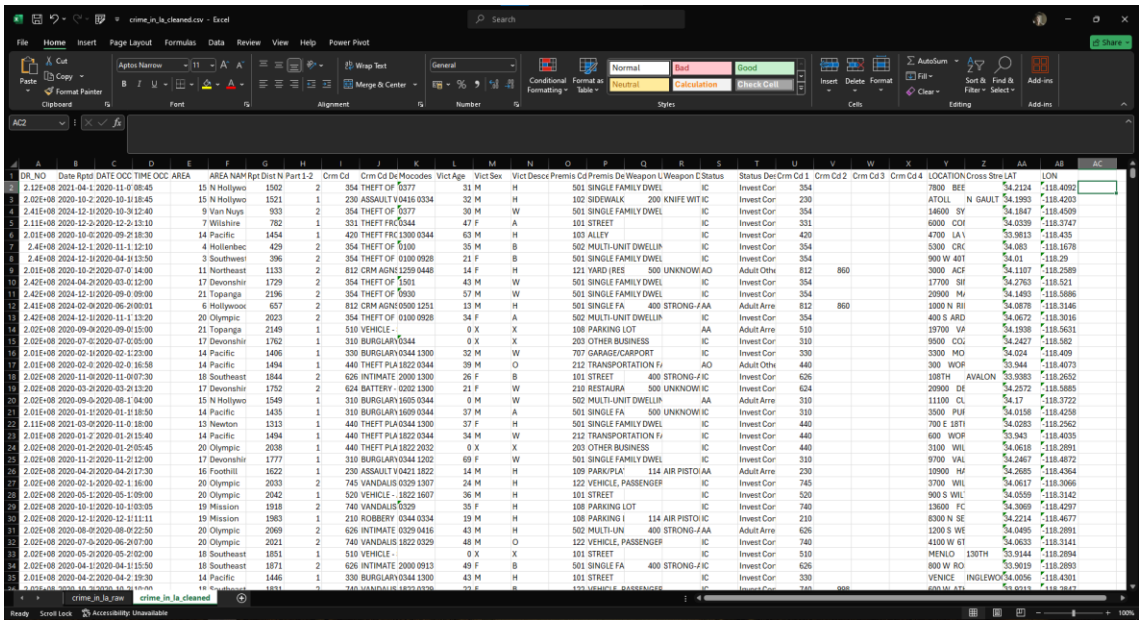
Note that steps one and two took place typically and proactively, since Power Query’s overview revealed no issues. We could have added steps for removing errors/duplicates but we eventually did not because column profiling based on top 1000 rows showed no errors.



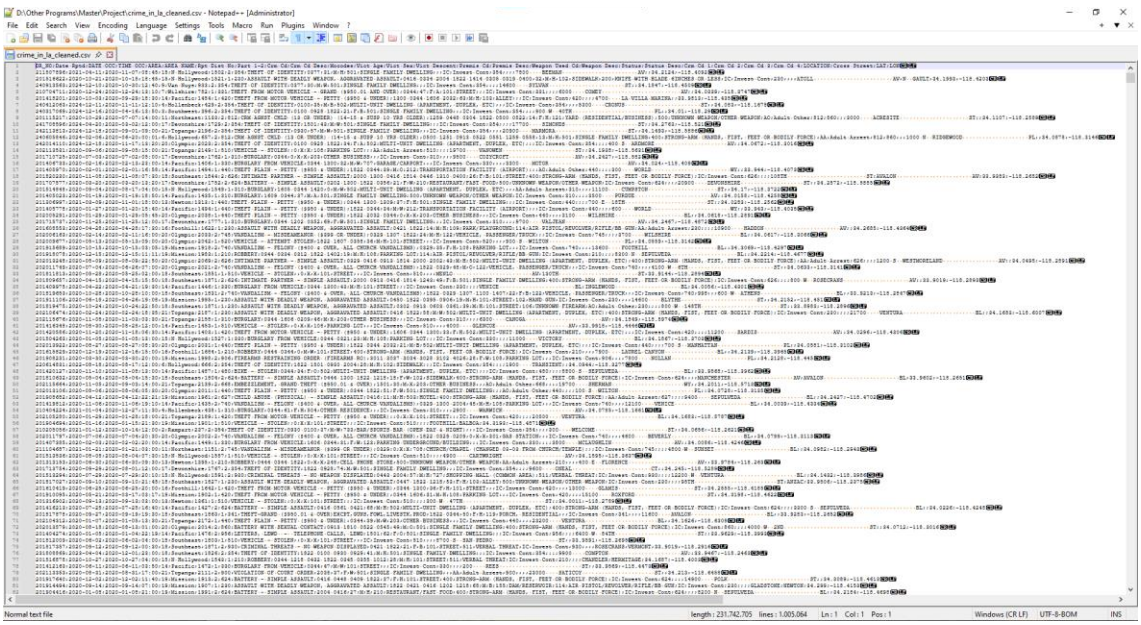
Screenshot 2.1: “Applied steps in Power Query”



Screenshot 2.2: “Applied steps in Power Query”



Screenshot 3: “Database-ready version of CSV”

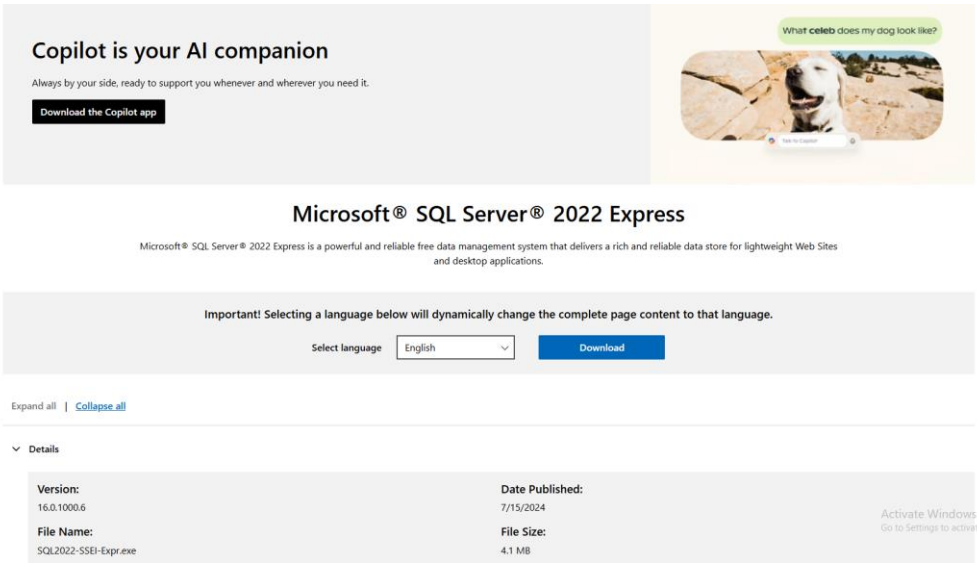


Screenshot 4: "Database-ready version of CSV in Notepad"

3.4. Installation of SQL Server 2022:

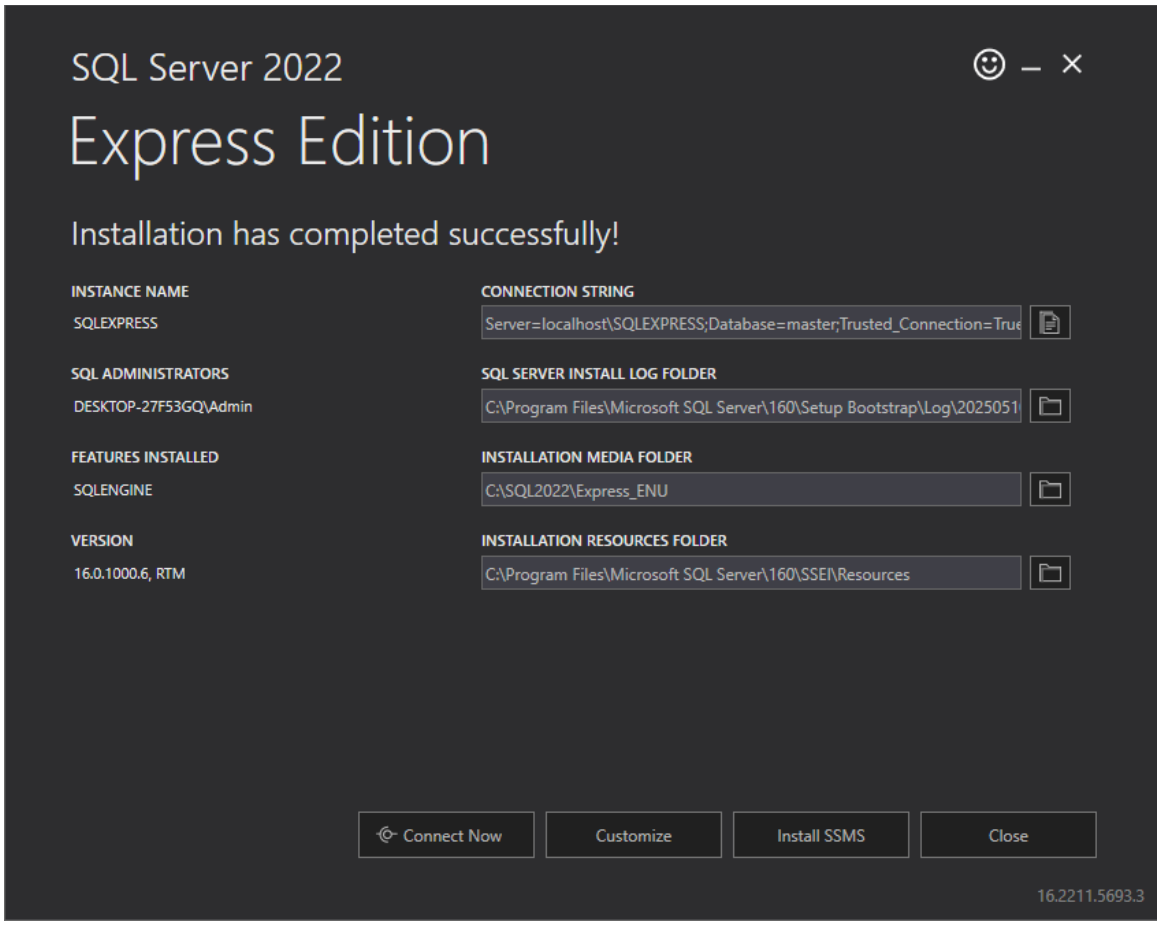
Since data cleaning and formatting of the CSV had come to an end, as it was mentioned before, we proceeded with the creation of a database and its tables in order to be able to store, organize and analyze our data in the most efficient manner. We leveraged SQL (also known as Structured Query Language) after setting up the database server.

First of all, we downloaded and installed Microsoft® SQL Server® 2022 Express, the system that stored, managed, and processed our data.



Screenshot 5: "Downloading SQL Server"

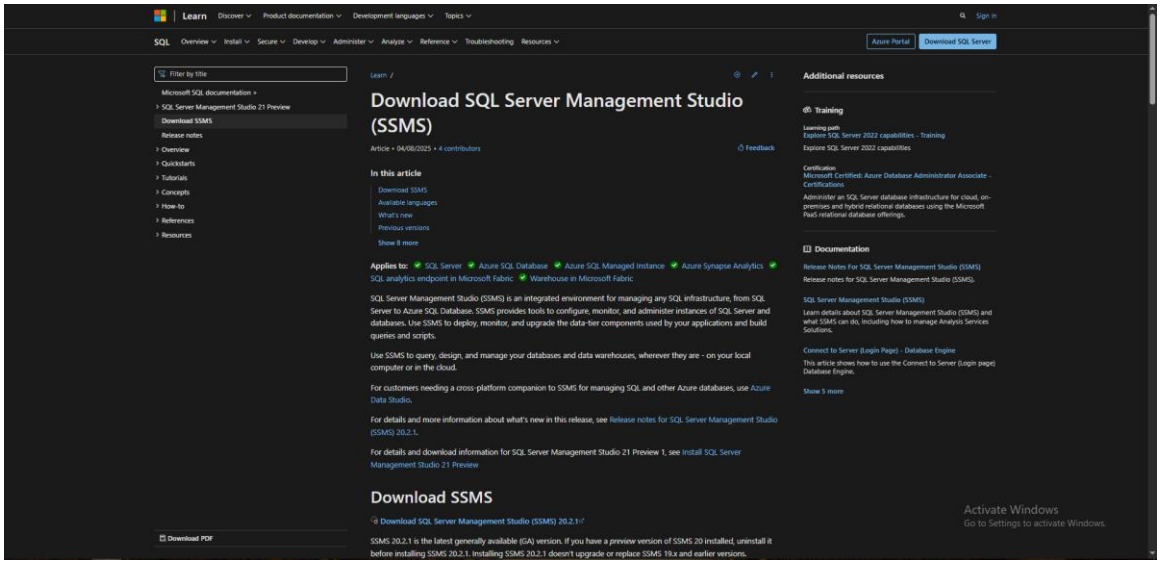
After downloading, we installed by following a couple of plain steps, which are shown in Appendix 10.



Screenshot 6: "SQL Server installed"

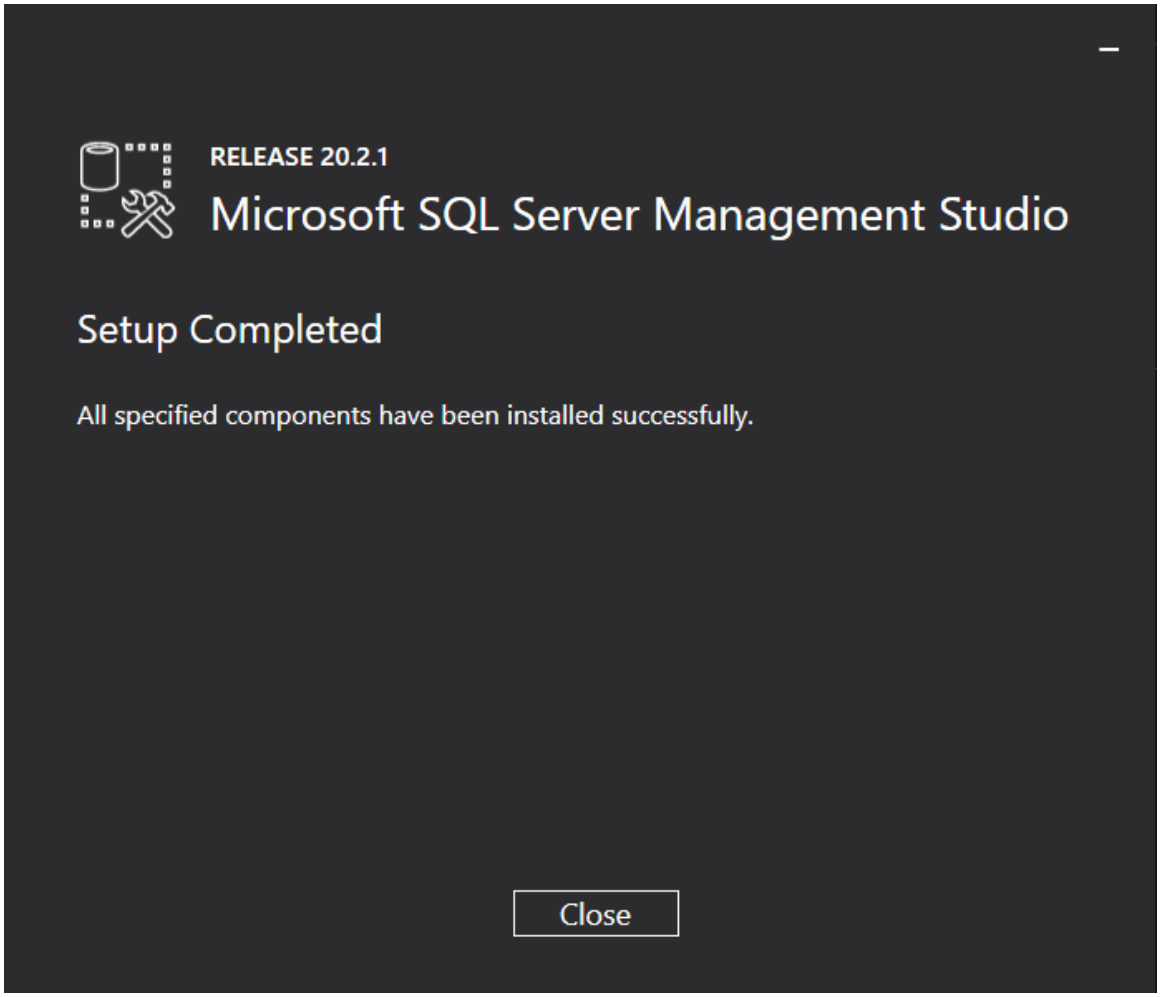
3.5. Installation of SSMS:

Secondly, we downloaded and installed another known Microsoft application which is built to facilitate compiling, executing and debugging SQL queries, creating tables, relationships, views, backups and other functions, the SQL Server Management Studio (SSMS). SSMS is a management tool (GUI) which allows us to connect to our server, either locally or remotely.



Screenshot 7: "Downloading SSMS"

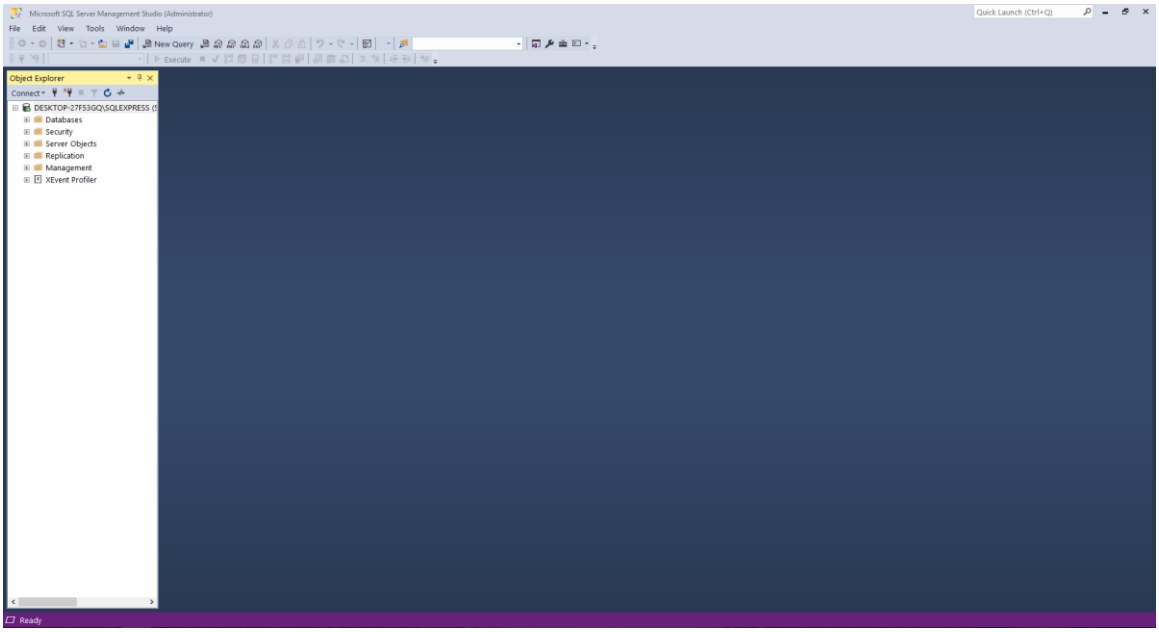
As soon as the file was downloaded, we proceeded into installing. The two installation steps are included in Appendix 11.



Screenshot 8: "SSMS installed"

3.6. Connection establishment between SQL Server and SSMS:

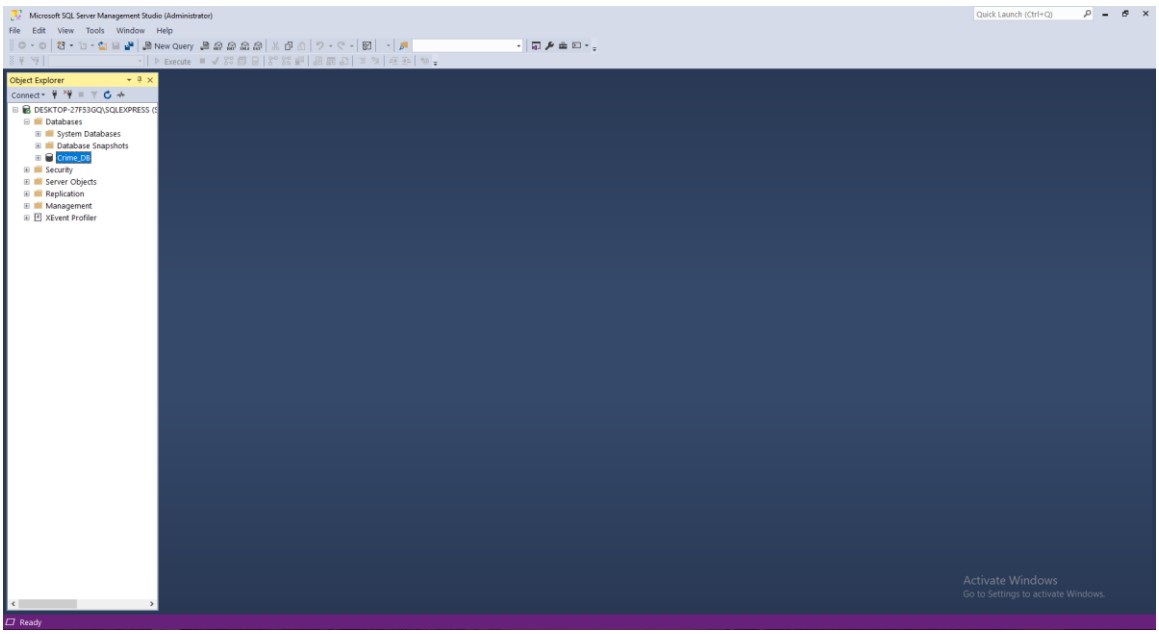
Since the downloading and installation of both programs had been completed, we moved on and set up a connection between SSMS and our SQL Server by carrying out the steps shown in Appendix 12.



Screenshot 9: “Connection with server ready”

3.7. Database creation:

The next step that brought us closer to having a complete database with the appropriate tables and imported data was the creation of a database for our crime_in_la_cleaned.csv file. This was done by executing the steps located at Appendix 13.

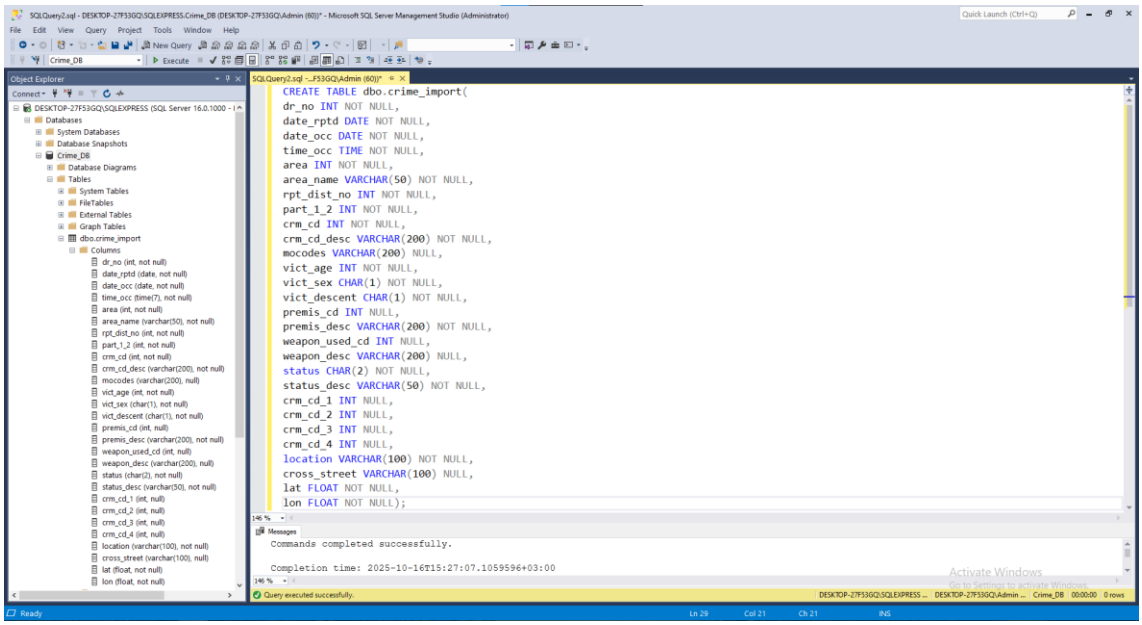


Screenshot 10: “Database created”

3.8. Import table creation:

Regarding the tables, we created our first one using an SQL query. The two simple steps that we used are shown in Appendix 14.

This query created the table and its columns that hold our data (the length of each VARCHAR column was determined by examining the maximum number of characters in each respective column of the CSV file and “NULL/NOT NULL” constraints were assigned based on whether missing values were observed in each column). We confirmed that they had been created successfully by checking the screenshot below.



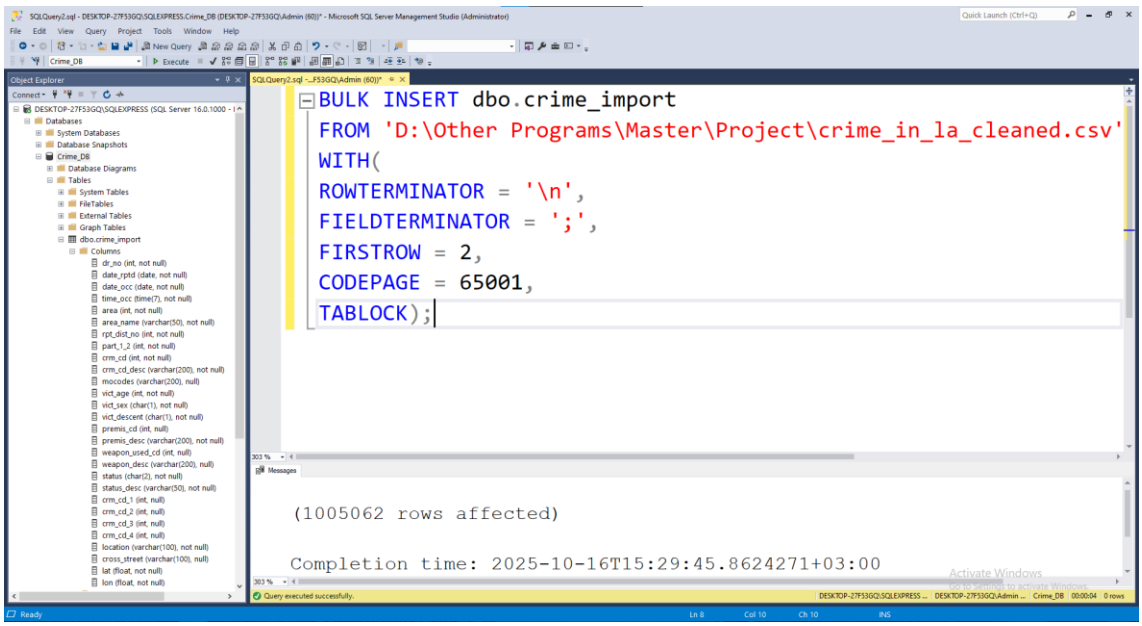
Screenshot 11: “Table and Columns constructed”

3.9. Data importation into the crime_import table:

Having created our database, along with the table and its fields, we proceeded to importing the data from the CSV file into our table. The steps for this are recorded in Appendix 15, in which:

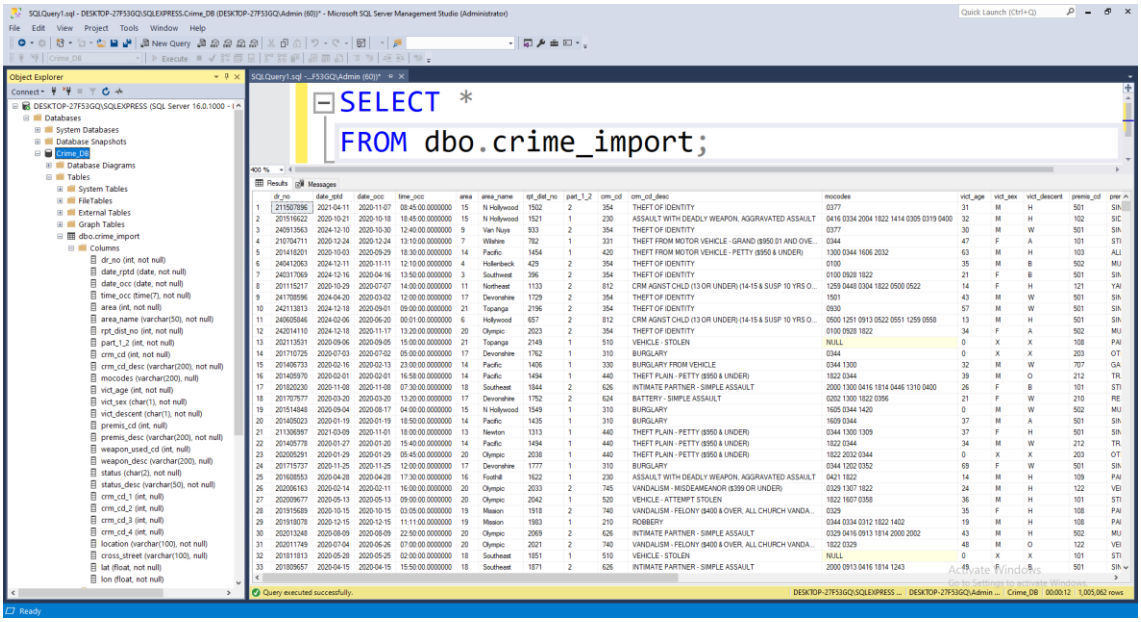
- `FROM` reads the data contained in the complete file path `'D:\Other Programs\Master\Project\crime_in_la_cleaned.csv'` of the CSV.
- `WITH` contains the parameters in parenthesis.

- `ROWTERMINATOR` states that every new record in the file ends with a line feed `'\n'`. This is common in UTF-8 files which are created in modern operating systems e.g., Windows 10.
- `FIELDTERMINATOR` states that columns are segregated with `';'`, instead of a ‘,’. This is mandatory if the CSV has been saved in greek regional settings (where “,” is used as a decimal character).
- `FIRSTROW` indicates that the first line of the file comprises the titles of columns and the actual data begin from line 2.
- `CODEPAGE` determines that the CSV is using UTF-8 coding (`65001`), so that greek or other non-ASCII characters are supported correctly.
- `TABLOCK` uses an exclusive table-level lock during import that improves performance when large amounts of data are imported.



Screenshot 12: “Data imported into crime_import”

Optionally, we confirmed that everything looked decent by using the command shown in Appendix 16, which selected all data from our table. Note that the first five data rows, along with the column headers, are also presented in Appendix 16.



Screenshot 13: “All data presented”

3.10. Creation of fact and dimension tables:

Since the preprocessing of our data, installation of SQL Server 2022 and SSMS, creation of the database and a central table and importation of data into the table were completed, the next thing we did is create a fact table and seven dimension tables using SQL code. These tables were designed so that each would contain fields directly related to it and to support the construction of SQL queries in the next sections. For example, we have columns that indicate geographical information, such as:

- `area`,
- `area_name`,
- `rpt_dist_no`,
- `location`,
- `cross_street`,
- `lat`,
- `lon`.

We also have columns that are relevant to crime information with the names:

- `part_1_2`,
- `crm_cd`,
- `crm_cd_desc`,

- mocodes,
- crm_cd_1,
- crm_cd_2,
- crm_cd_3,
- crm_cd_4,
- status,
- status_desc.

There are columns related to premises with the names:

- premis_cd,
- premis_desc.

Moreover, there are fields that insinuate victim information, such as:

- vict_age,
- vict_sex,
- vict_descent,

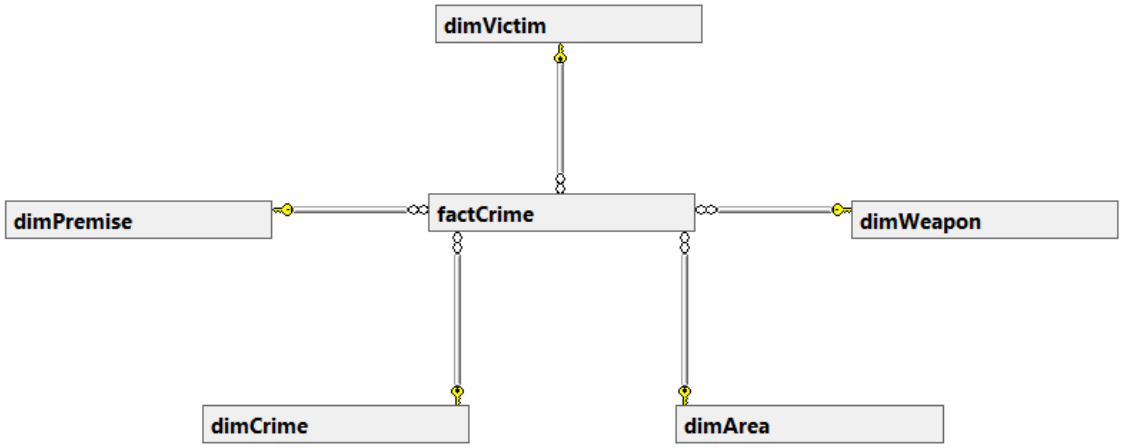
and columns that are relevant to weapons, like:

- weapon_used_cd,
- weapon_desc.

Lastly, there are columns that indicate chronological information and these are:

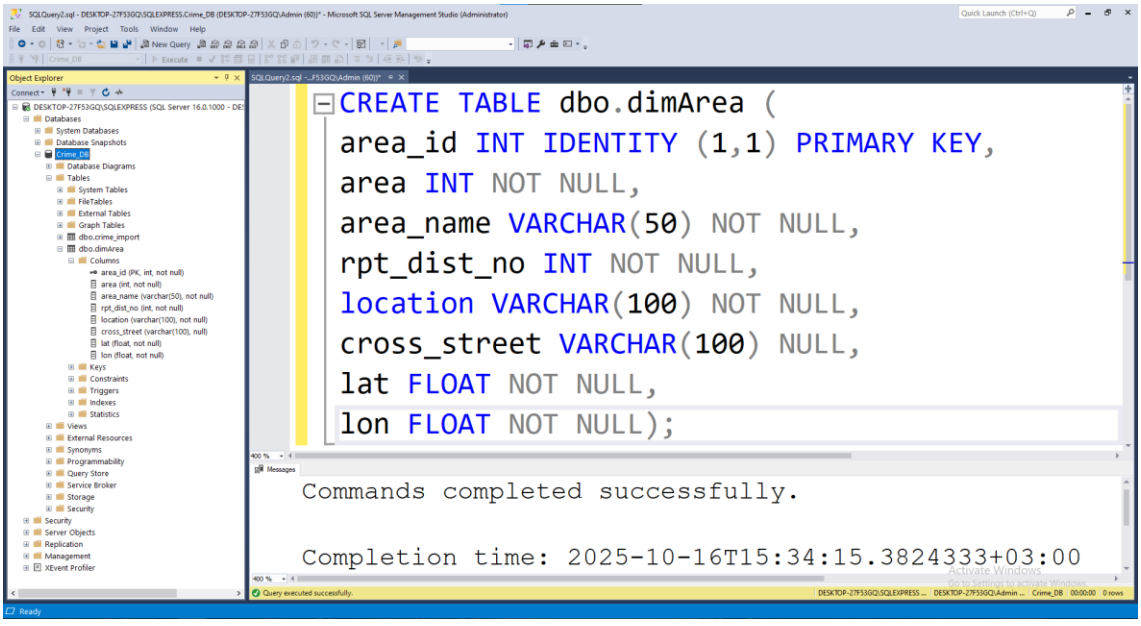
- date_rptd,
- date_occ,
- time_occ,
- dr_no (we include this because the older the crime, the smaller the value of dr_no, since it increments by one every time a crime incident happens).

Before the construction of the fact and dimension tables, a high-level ERD schema is presented, depicting the basic entities of our database. In this schema, only the tables names and the relationships between them are shown, providing a clear overview of the final database structure and preparing the ground for what follows. The complete ERD, including all keys and attributes, is presented later in Section 3.12.



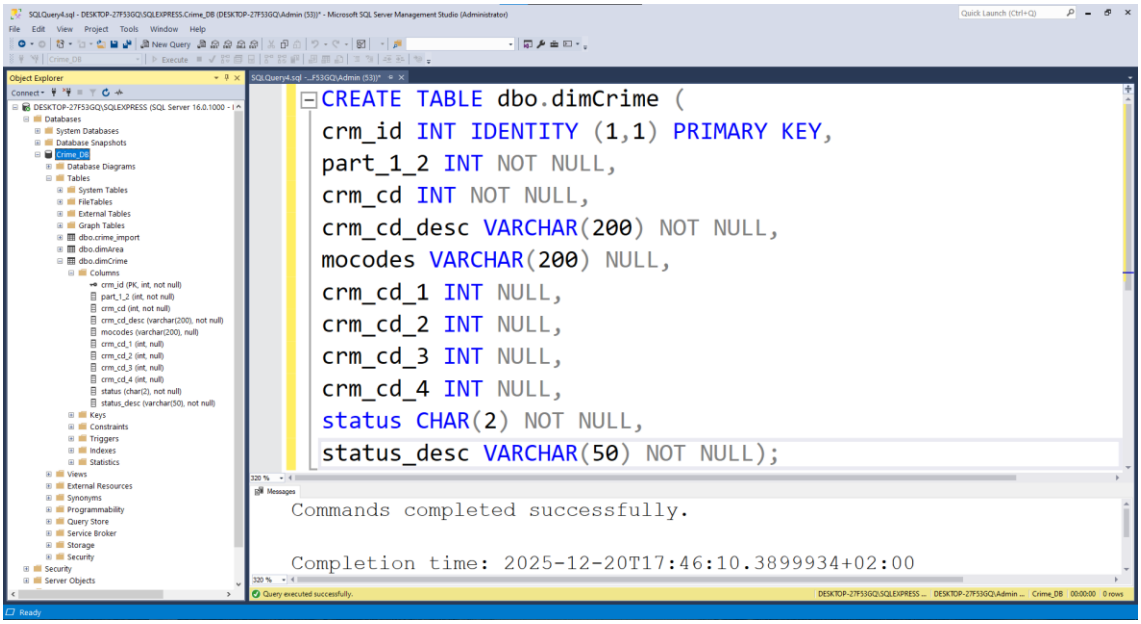
Screenshot 14: “High-level ERD overview”

Starting with the dimension tables, we created the first one called `dimArea` in a couple of steps. The steps are shown in Appendix 17, in which `area_id` is an integer type primary key that is automatically generated with an increasing value, starting from 1. When the data were inserted into the table, all records had a unique combination of values. For example, if our dataset contained two incidents with the same `area`, `area_name`, `rpt_dist_no`, `location`, `cross_street`, `lat`, and `lon` values, both incidents would have the same `area_id` value in the fact table (which would be created later) and this combination of values would only appear once in the `dimArea` table, associated with its own unique `area_id`.



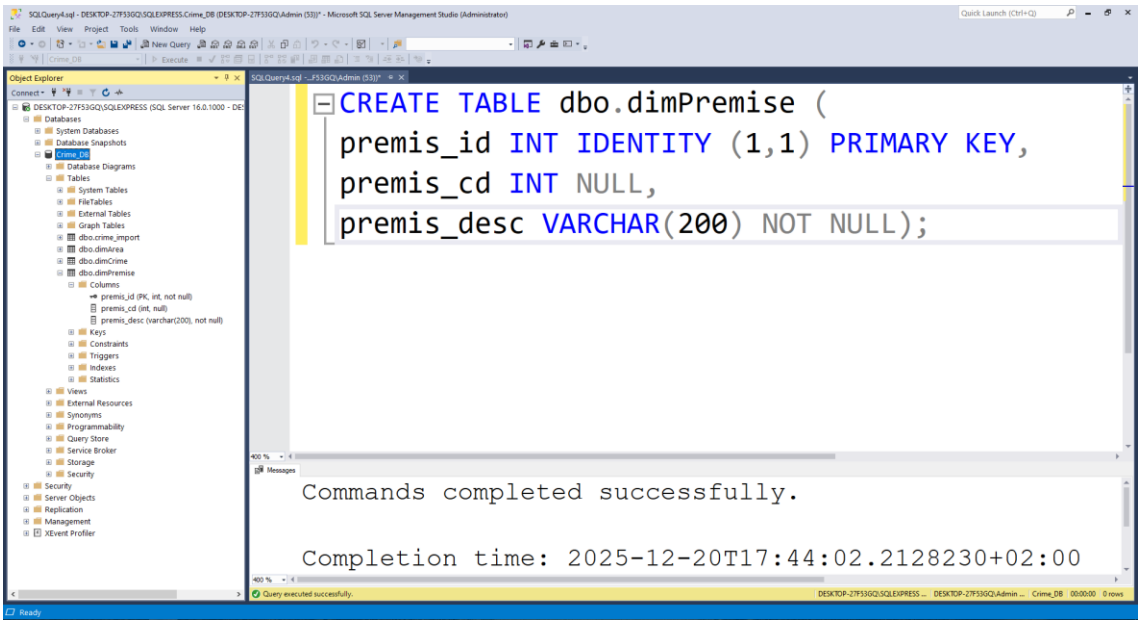
Screenshot 15: “dimArea created”

The next dimension table that we constructed is `dimCrime` by using the same steps as the previously created table, `dimArea`. The code utilized is shown in Appendix 18 and the primary key here that auto-increments is `crm_id`.



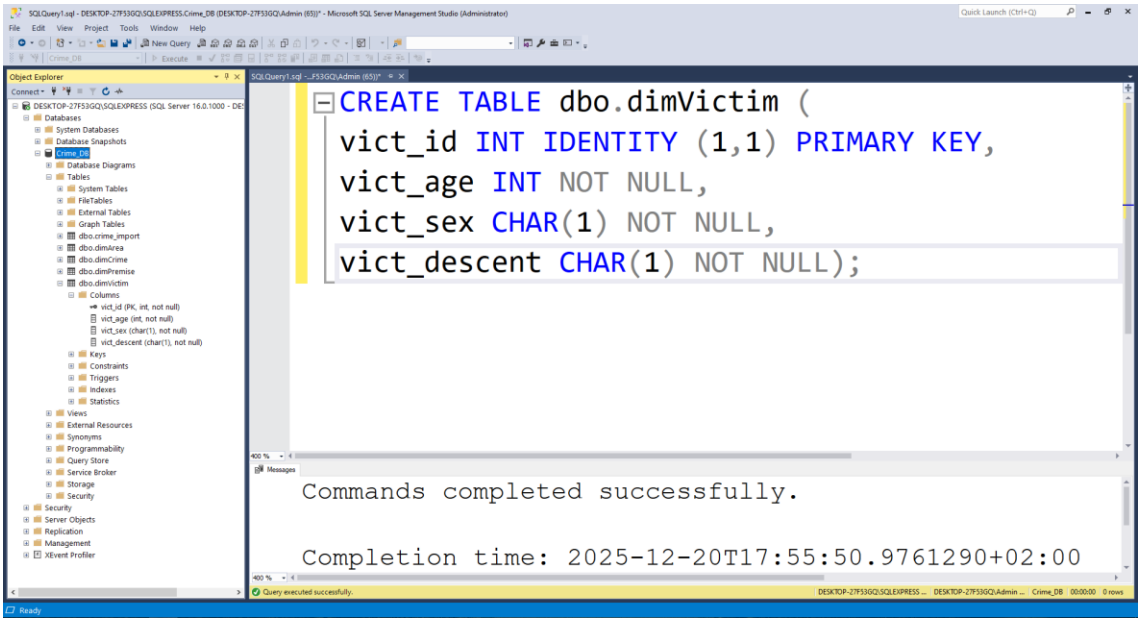
Screenshot 16: "dimCrime created"

The third dimension table that was created is `dimPremise` with the code shown in Appendix 19. The primary key here that auto-increments is `premis_id`.



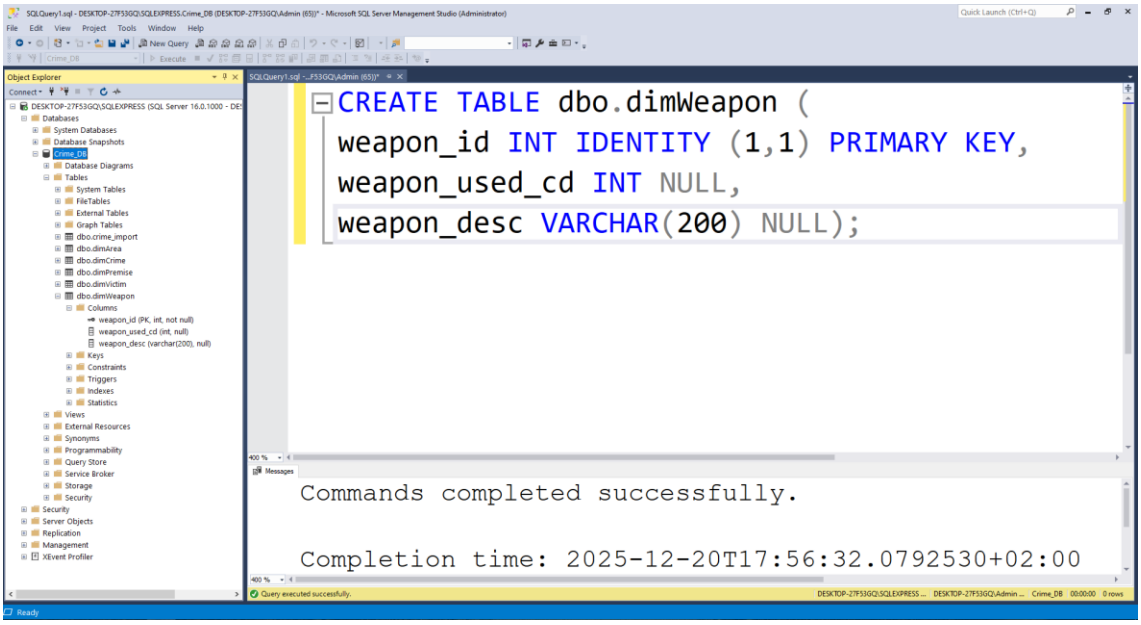
Screenshot 17: "dimPremise created"

The fourth dimension table that we created by writing the command shown in Appendix 20 is `dimVictim`. The primary key here that auto-increments is `vict_id`.



Screenshot 18: "dimVictim created"

The fifth dimension table that was created is `dimWeapon`. The code used is shown in Appendix 21. The primary key here that auto-increments is `weapon_id`. Note that the record with no values in `weapon_used_cd`, and `weapon_desc` would also have a `weapon_id` value.

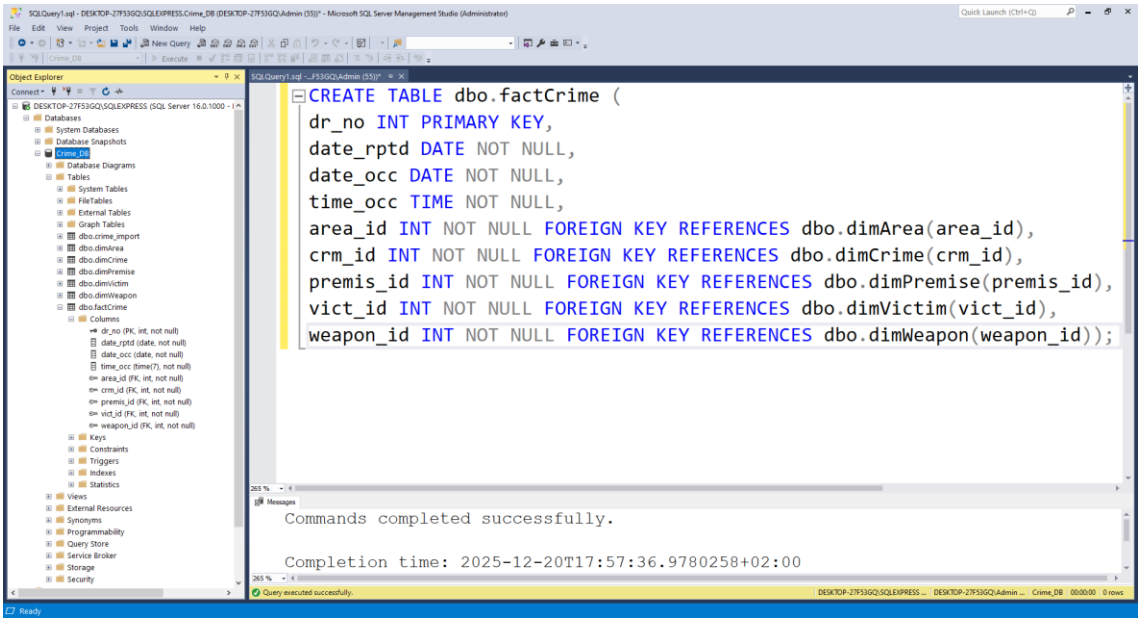


Screenshot 19: "dimWeapon created"

The last table that we created is the fact table. It has the name `factCrime`. The code used is shown in Appendix 22, in which:

- `dr_no` is the primary key.
- `area_id`, `crm_id`, `premis_id`, `vict_id`, and `weapon_id` are added as foreign keys, each pointing to a corresponding dimension table. This results in the connection of the fact table with the respective dimension tables. In this way, the fact table gathers all crime incidents and links them to contextual information stored in the dimension tables (areas, crime types, locations, victims, weapons), forming a star schema. Moreover, all five foreign keys are defined as "NOT NULL", since every crime incident is associated with corresponding entries in the dimension tables.

Note that `date_rptd`, `date_occ`, and `time_occ` fields were added in the fact table and not in any other dimension table, as every incident has its own timestamp. There was no point in adding these three in a dimension table because dimension tables contain a smaller, repetitive list of data, namely locations, weapons, crimes, and victims.



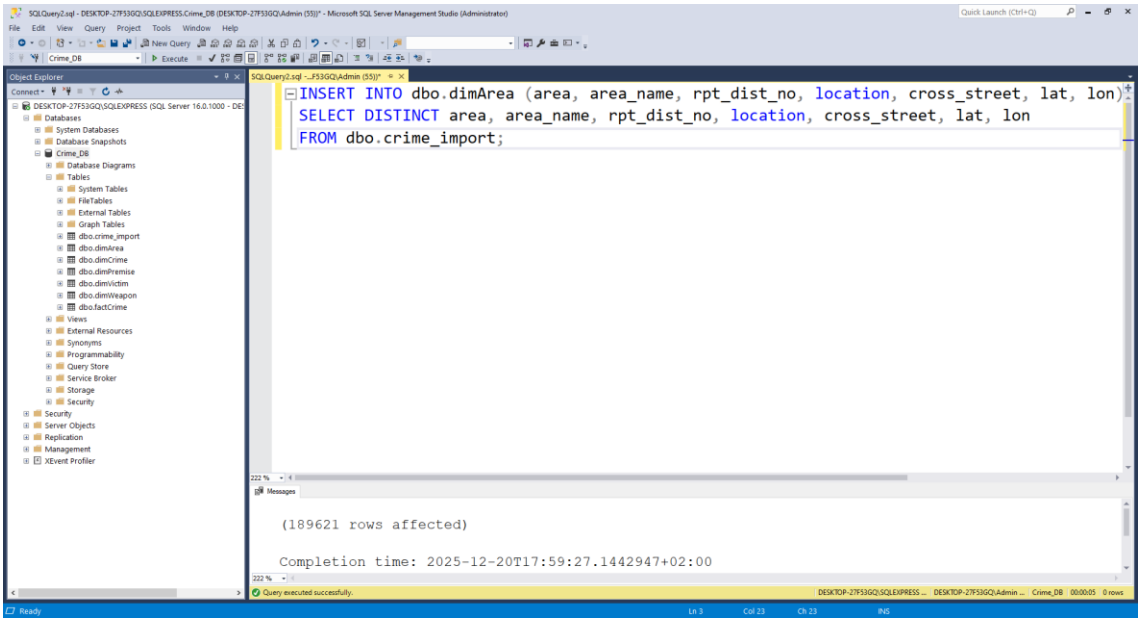
Screenshot 20: "factCrime created"

3.11. Data importation into fact and dimension tables:

The current section of our project is related to the importation of data from the `crime_import` table into the dimension and fact tables we created a while ago. Right after the execution of an import query, a query that selects all data from the respective table is utilized to prove that everything went well, and its results are summarized in the Appendix (column headers and first five data rows).

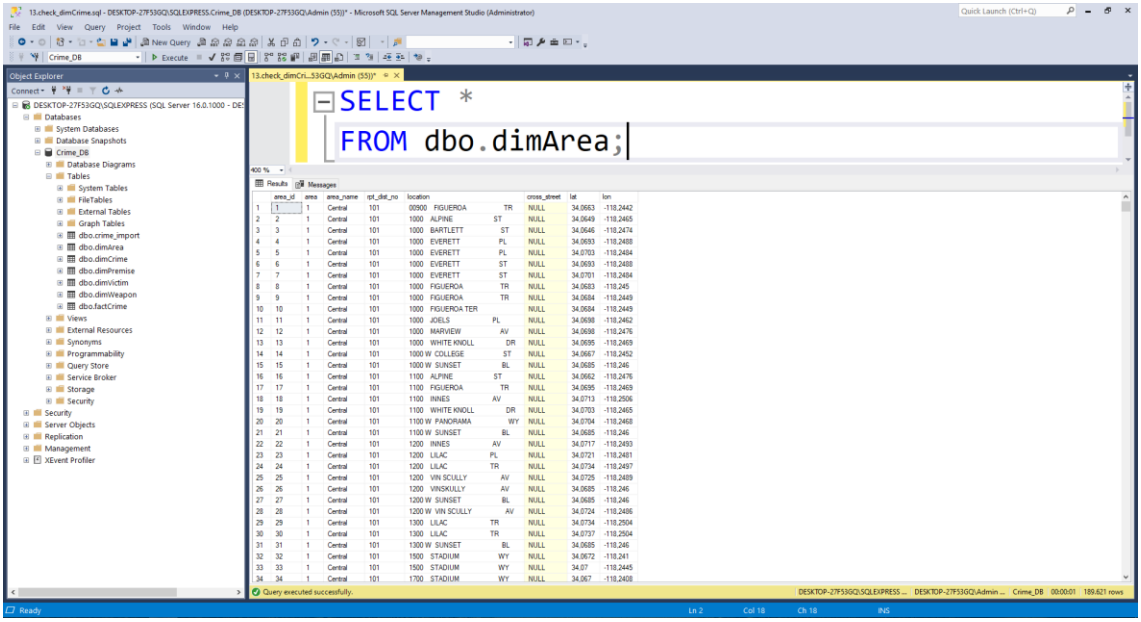
The first table that we imported our data into is `dimArea`. To achieve this, we utilized the chunk of code shown in Appendix 23, in which:

- `INSERT INTO`, the command responsible for the importation, specifies two things:
    1. the table in which we want to insert the data into (`dimArea` in this case),
    2. and its columns that we want to populate (`area`, `area_name`, `rpt_dist_no`, `location`, `cross_street`, `lat`, `lon`).
- `SELECT DISTINCT` then obtains all the unique combinations of the column values `FROM` the `crime_import` table.



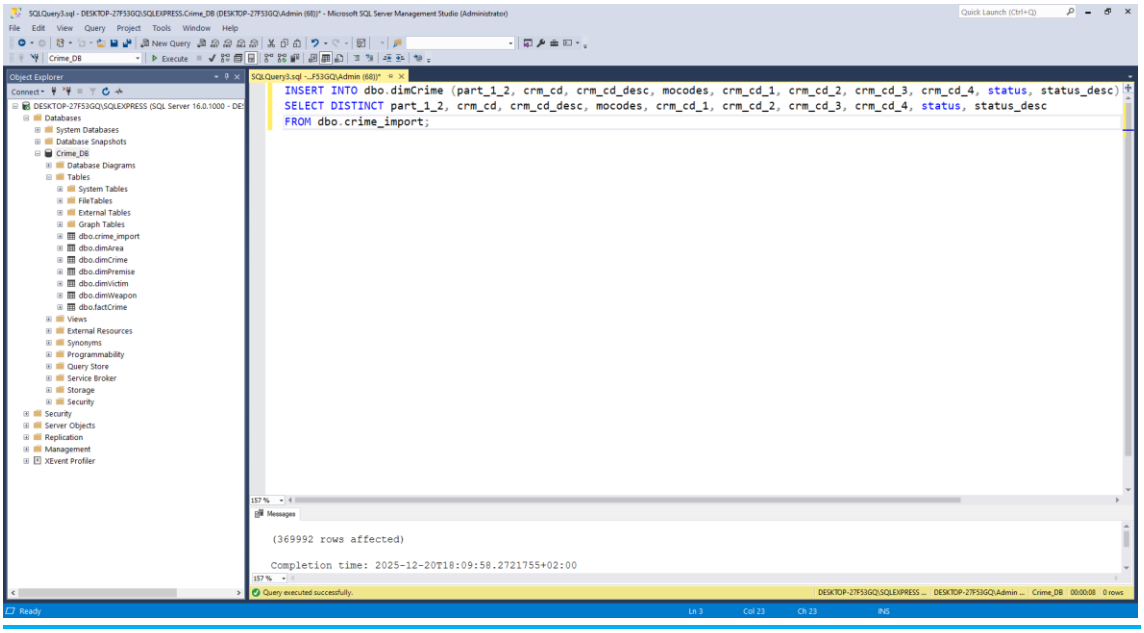
Screenshot 21: “Data imported into dimArea”

Optionally, we used the command shown in Appendix 24 to see that the insertion of data went well. The query output is shown in Appendix 24.



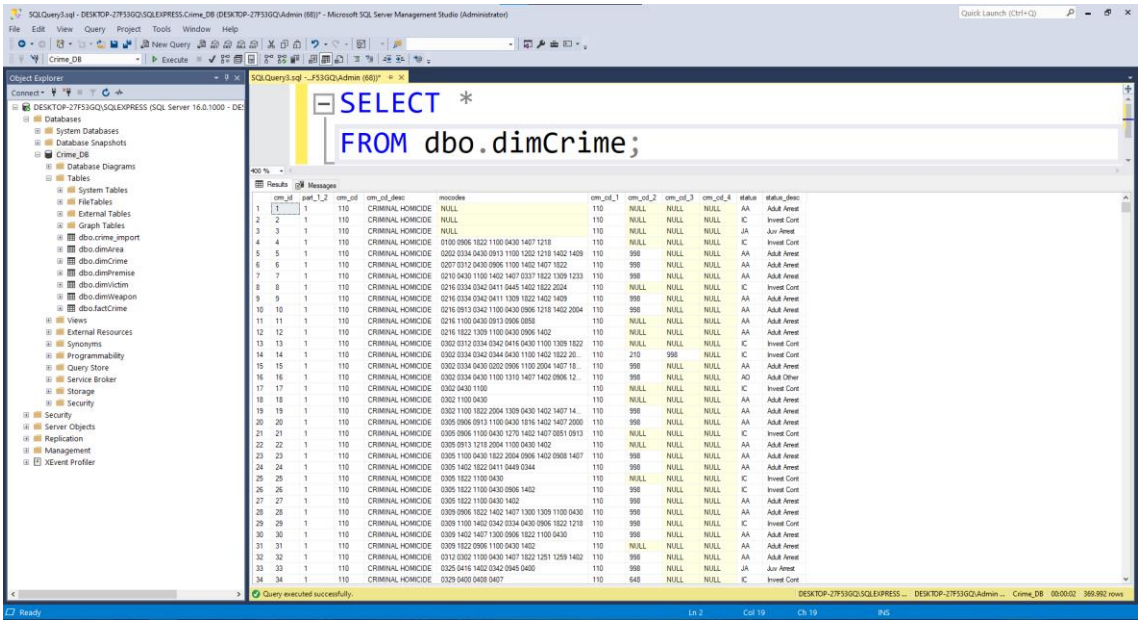
Screenshot 22: “All area related data presented”

The second table that accommodated data relative to it from `crime_import` table is the dimension table `dimCrime`. This took place with the command shown in Appendix 25.



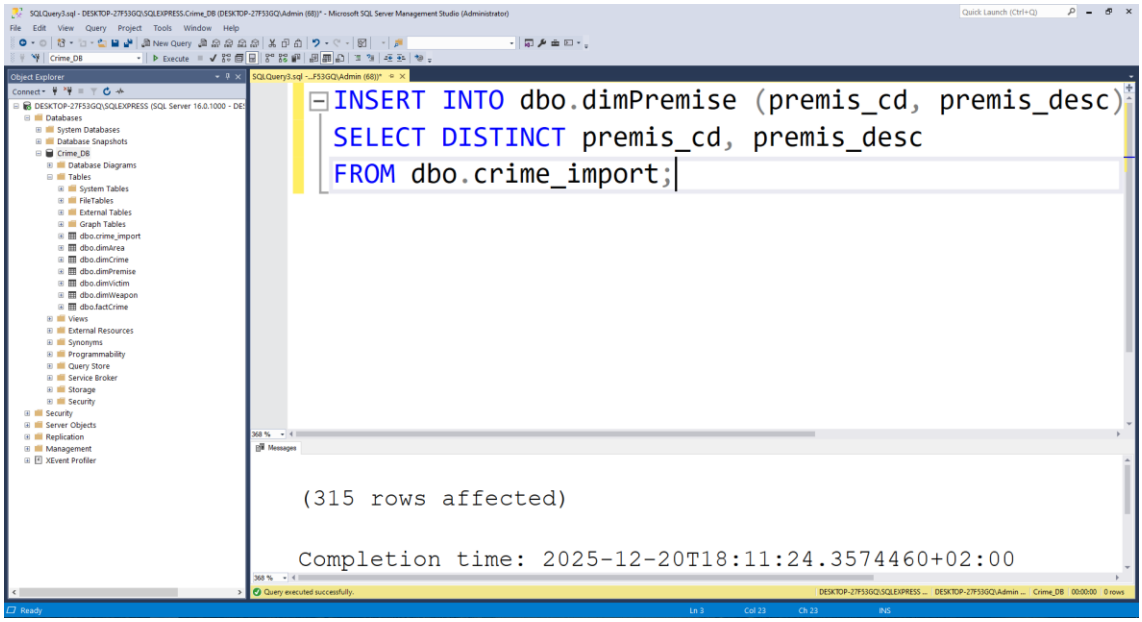
Screenshot 23: “Data imported into dimCrime”

Optionally, the code in Appendix 26 was utilized to make sure that the import was successful. The query output is shown in Appendix 26.



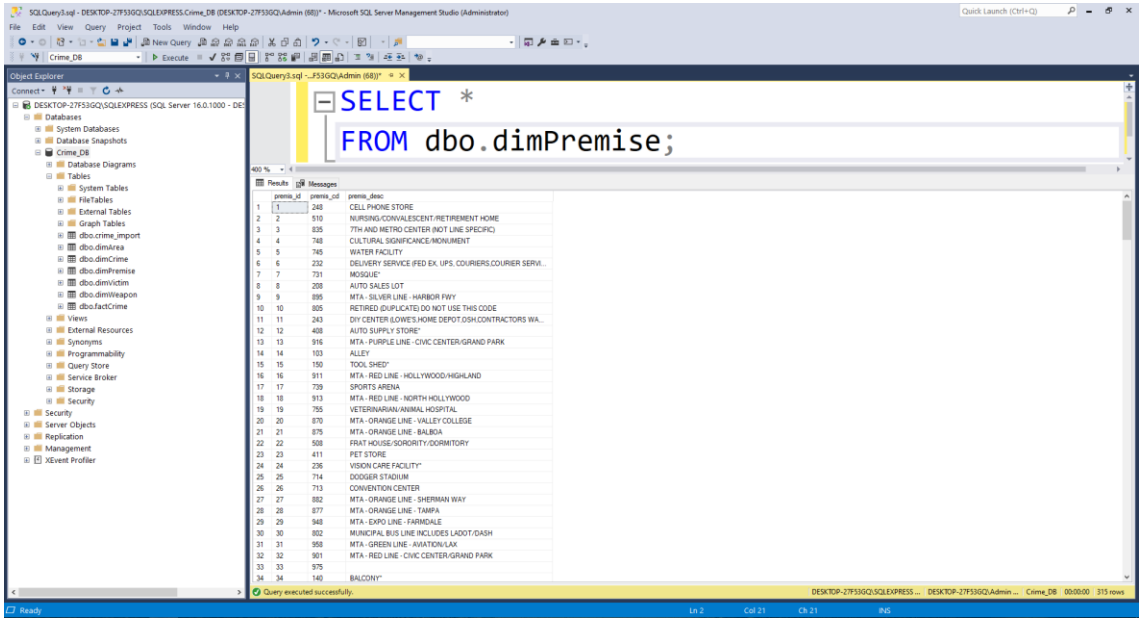
Screenshot 24: “All crime related data presented”

The next table that we inserted data into is the table `dimPremise`. The insertion happened with the code shown in Appendix 27.



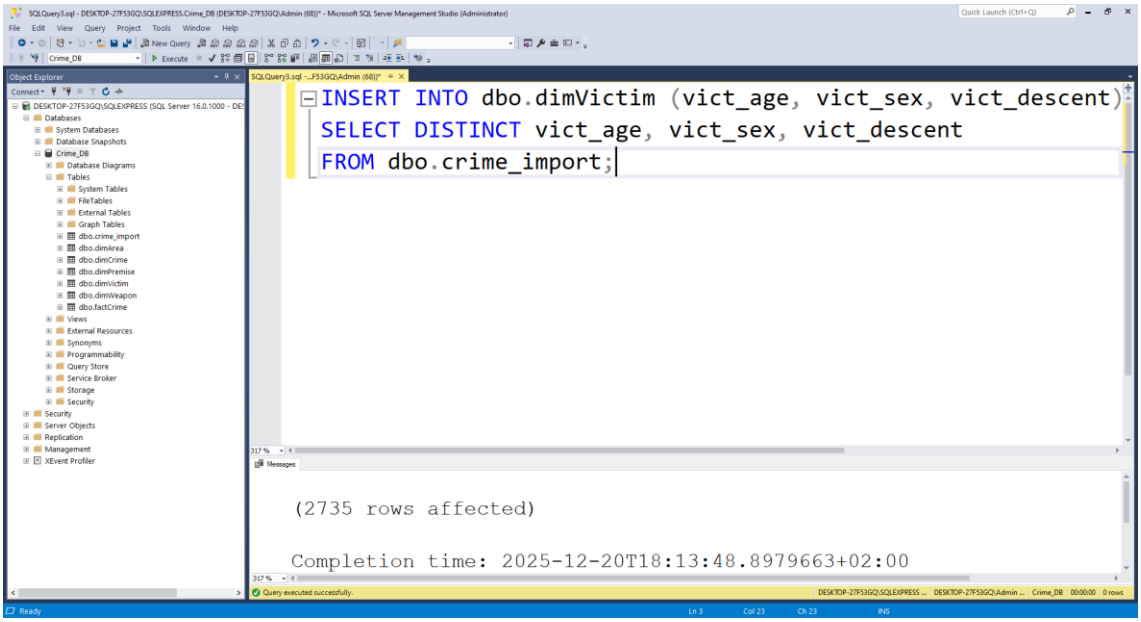
Screenshot 25: “Data imported into dimPremise”

Optionally, we used the code shown in Appendix 28 to make sure everything went well. The query output is shown in Appendix 28.



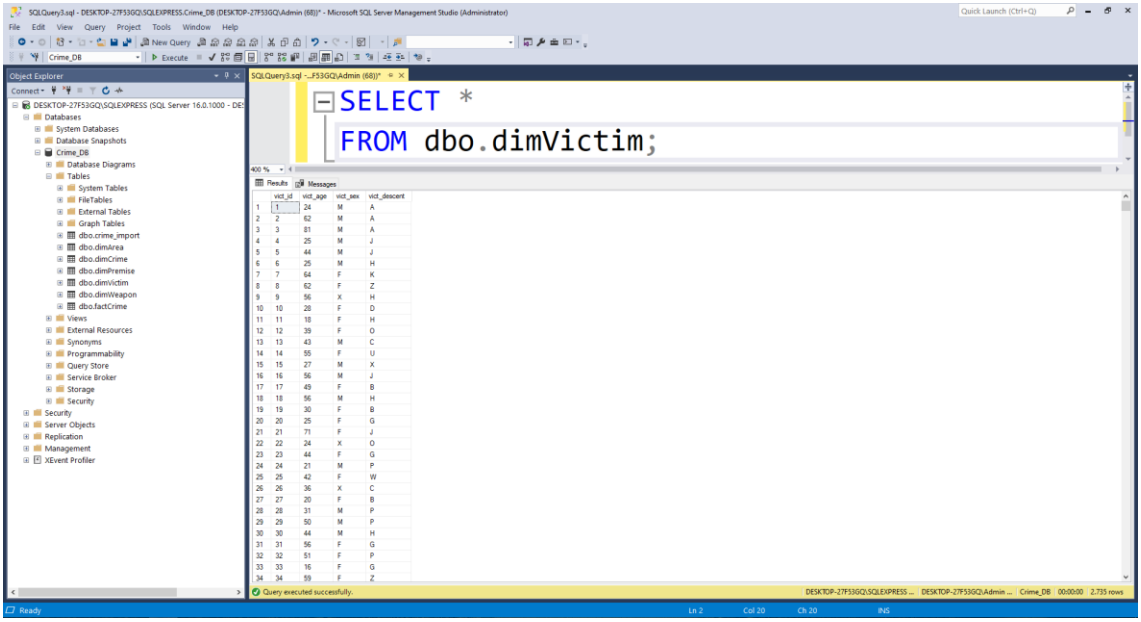
Screenshot 26: “All premise related data presented”

The fourth table that we imported data into is `dimVictim`, by using the query shown in Appendix 29.



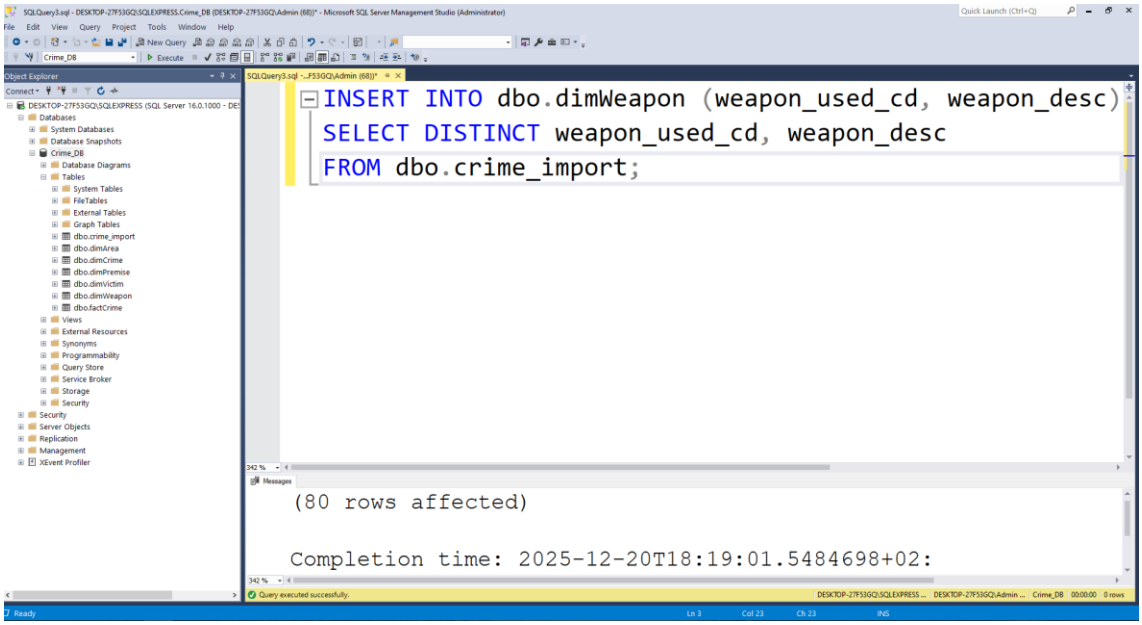
Screenshot 27: “Data imported into dimVictim”

Optionally, the command shown in Appendix 30 was utilized in order to make sure that data import was successful. The query output is shown in Appendix 30.



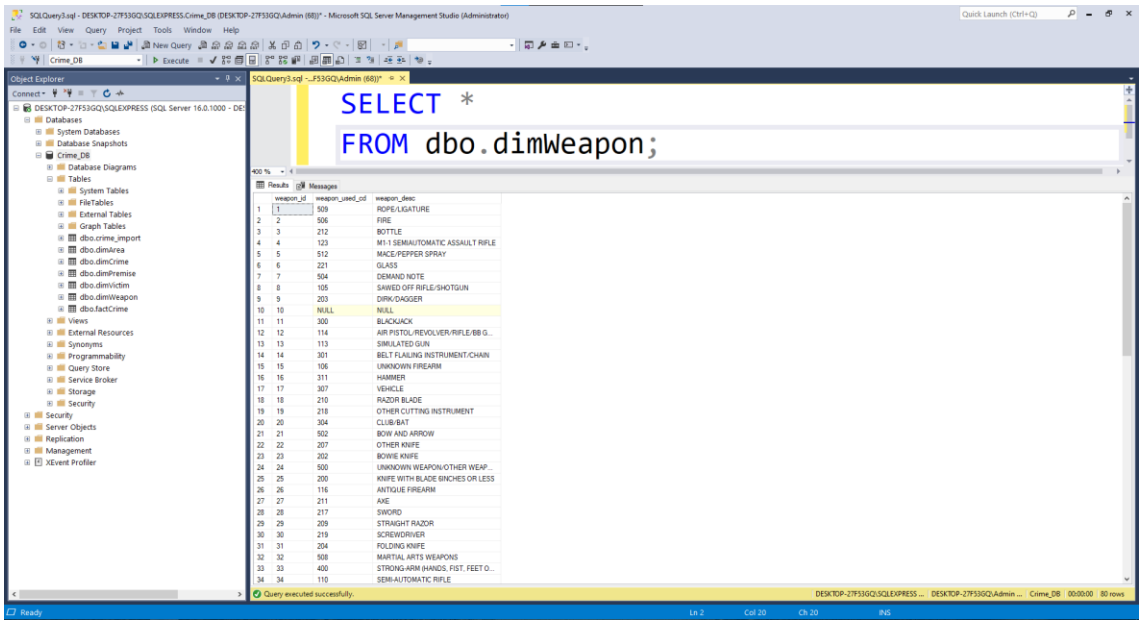
Screenshot 28: “All victim related data presented”

The last dimension table that had data inserted into its columns is the table called `dimWeapon`. The command shown in Appendix 31 will be truly useful for this.



Screenshot 29: "Data imported into dimWeapon"

Optionally, we used the line of code shown in Appendix 32 to make sure that the importation of data in our table looks fine. The query output is shown in Appendix 32.

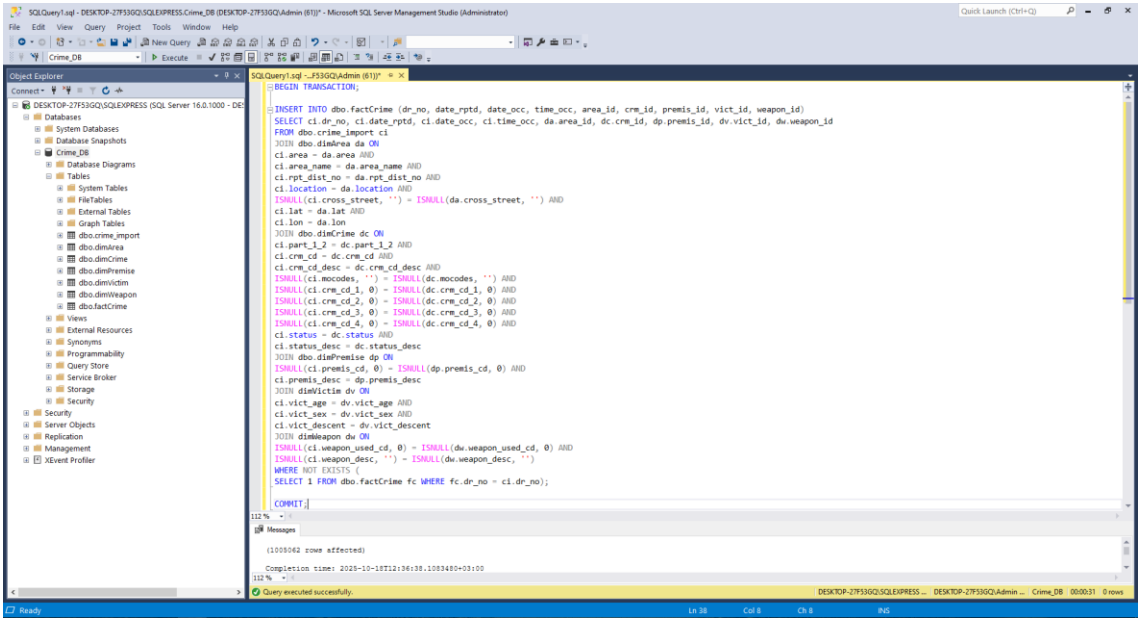


Screenshot 30: "All weapon related data presented"

Since the insertion of the right data into the right dimension tables had been completed, it was time for us to import the appropriate data into the fact table as well. We achieved our purpose with the command shown in Appendix 33, in which:

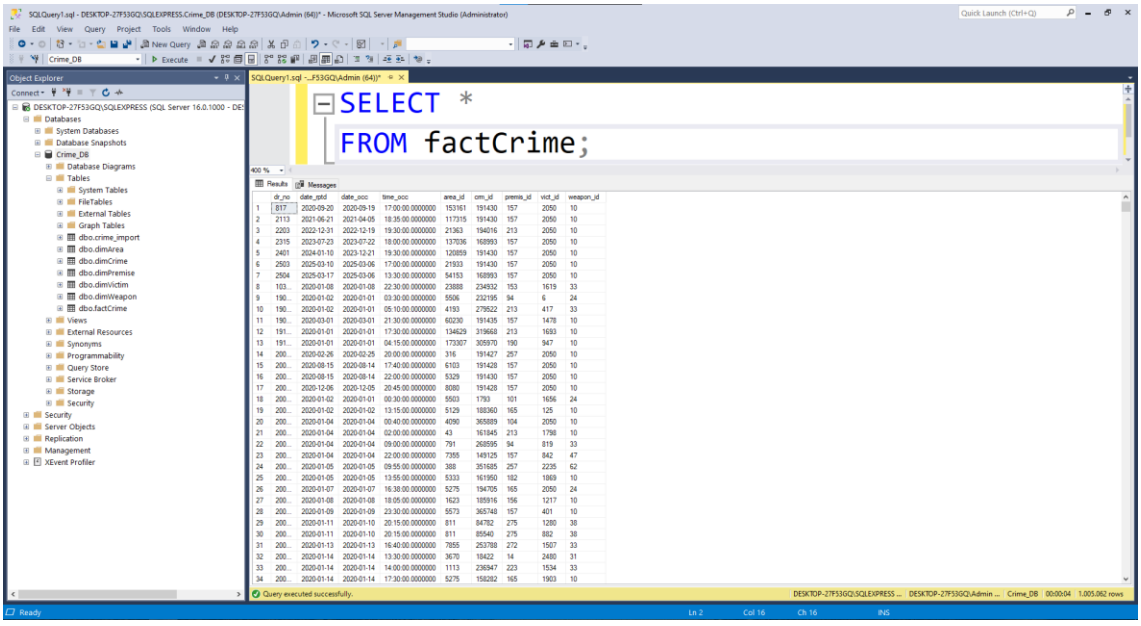
- `BEGIN TRANSACTION` does exactly what its name says. It begins a transaction, which means that all the following commands will be executed together as a whole. It is a precaution measure because it gives us the opportunity to examine if our code provides the desired result, without changing the structure of our database. If it does, we can commit to it and if it does not, we can roll back like nothing ever happened.
- `INSERT INTO` states the `factCrime` table's fields that we want to insert data into, such as `dr_no`.
- `SELECT` chooses the data of certain columns. These columns are referring their own table with an alias at the start of their names, for example, `dv.vict_id` refers to its own table, `dimVictim`, as `dv`.
- `FROM` contains the name of the table with an alias that we are going to extract the data from, the `crime_import` table in our case. It collaborates closely with the `JOIN…ON` clause which contains a dimension table's name with an alias between `JOIN` and `ON` and column names after `ON`. These columns that are also aliased after `crime_import` and the dimension table are common fields between the two tables and are used to create a correlation between them so that all the appropriate dimension table IDs are located for every `crime_import`'s record. For example, in the code shown in Appendix 34:
  1. we create a correlation between the `crime_import`, and `dimArea` tables, using their common fields, such as `area`, `area_name` etc. This correlation allows us to locate the `area_id` key for each corresponding `crime_import`'s record.
  2. the ISNULL() function returns the name of the `cross_street` field if it exists and if it does not exist, it returns `''` (blank). We are using this because, in SQL, null does not equal null and the only way to associate the two void `cross_street` values is to convert both of them to blanks.
- `WHERE NOT EXISTS` secures that a record with the same `dr_no` is not going to be inserted into the `factCrime` table again. It assists in avoiding duplicates.

- `COMMIT` finalizes the transaction. All changes in the `factCrime` table are saved in the database. In case there was a problem, we could have easily used the `ROLLBACK` statement instead.



Screenshot 31: “Data imported into factCrime”

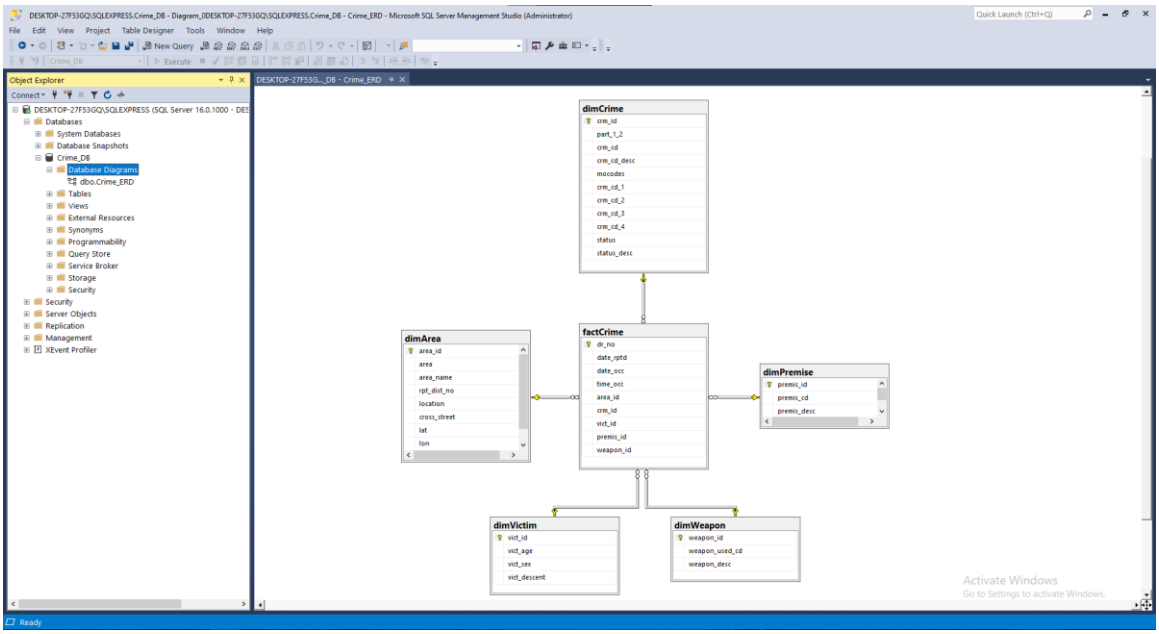
We saw that everything we needed was imported successfully by using the code with the well-known format shown in Appendix 35. The query output is shown in Appendix 35.



Screenshot 32: “All factCrime’s data presented”

3.12. Creation of the ERD:

After the importation of the right data into the right tables, we proceeded to the next step of our project, which was the creation of the ERD, also known as Entity-Relationship Diagram. ERD helped us visualize our database's structure, view the relationships between the fact and dimension tables neatly and validate that the connections (foreign keys) were correct and sensible. In order to construct this notable diagram, we implemented the steps shown in Appendix 36, in which we didn't include the `crime_import` table because all its data were already inside the dimension and fact tables we had created earlier, and this rendered the aforementioned table unnecessary for our diagram.



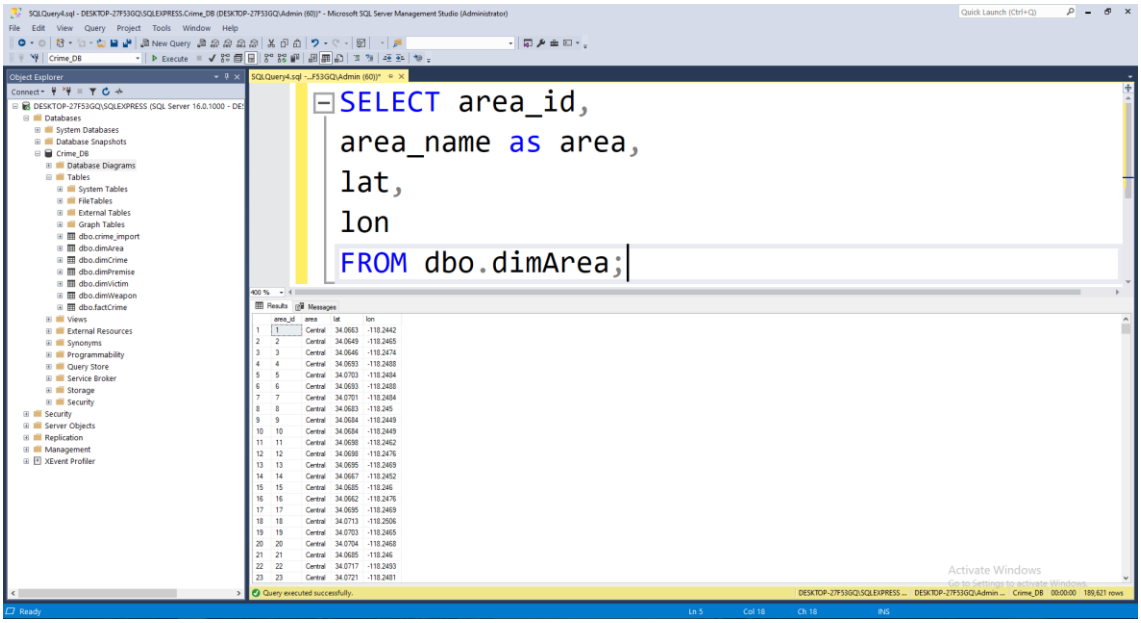
Screenshot 33: "ERD designed and saved"

3.13. SQL Query Construction and Evaluation:

The current section of our thesis is about the construction and testing of simple SQL queries. The queries in subsection 3.13.1 were designed to extract only the columns that would be directly used in Power BI visuals while the queries in subsection 3.13.2 would be exclusively used for validation and testing during the development process. We continued working within SSMS, and the results of each query are summarized in the Appendix (column headers and first five data rows).

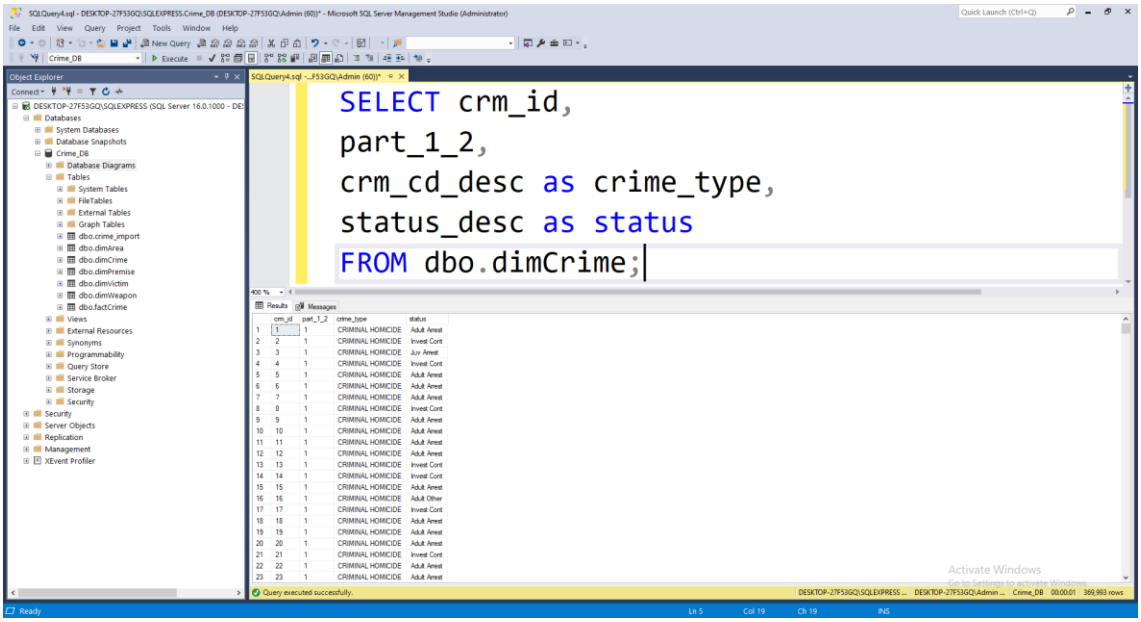
3.13.1. Core Queries for Star Schema:

The first query of this subsection is shown in Appendix 37, along with its output. It selects the key fields of `dimArea`, such as `area_id`, `area_name`, `lat`, and `lon`. `area_id` was included to ensure that the existing relationship between `dimArea` and `factCrime` in our database was also transferred to Power BI's data model. The remaining fields were retained because they would be required in subsequent visualizations.



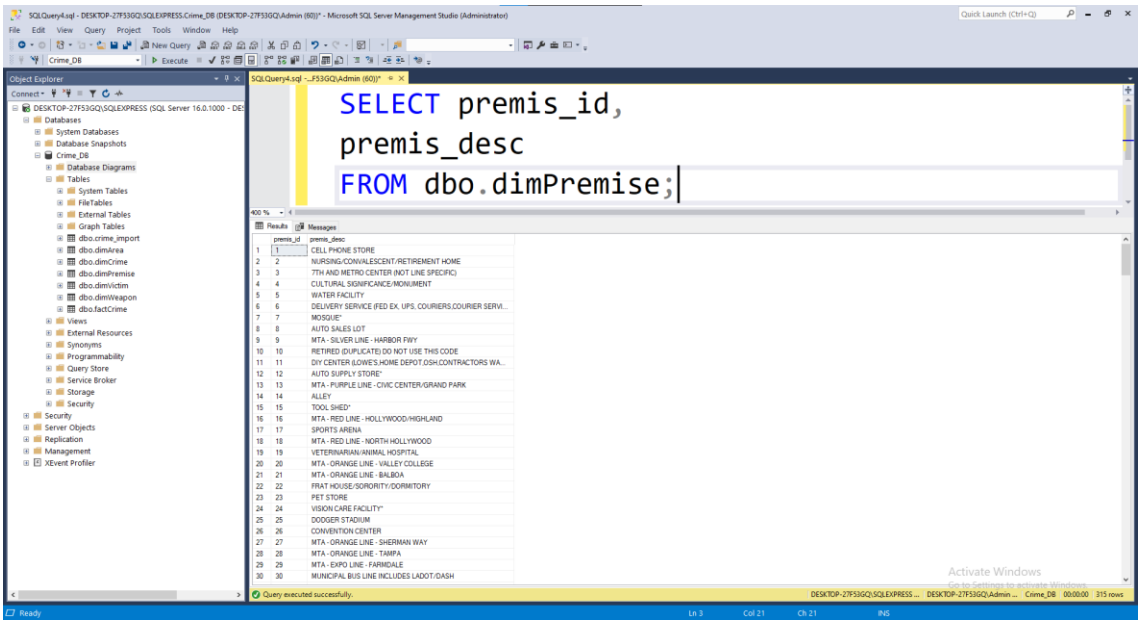
Screenshot 34: "dimArea: key fields returned"

The second query, shown in Appendix 38 along with its output, retrieves key fields from `dimCrime`. The field `crm_id` was included to preserve the relationship with the fact table, while `part_1_2`, `crm_cd_desc`, and `status` were retained for visualization purposes.



Screenshot 35: "dimCrime: key fields returned"

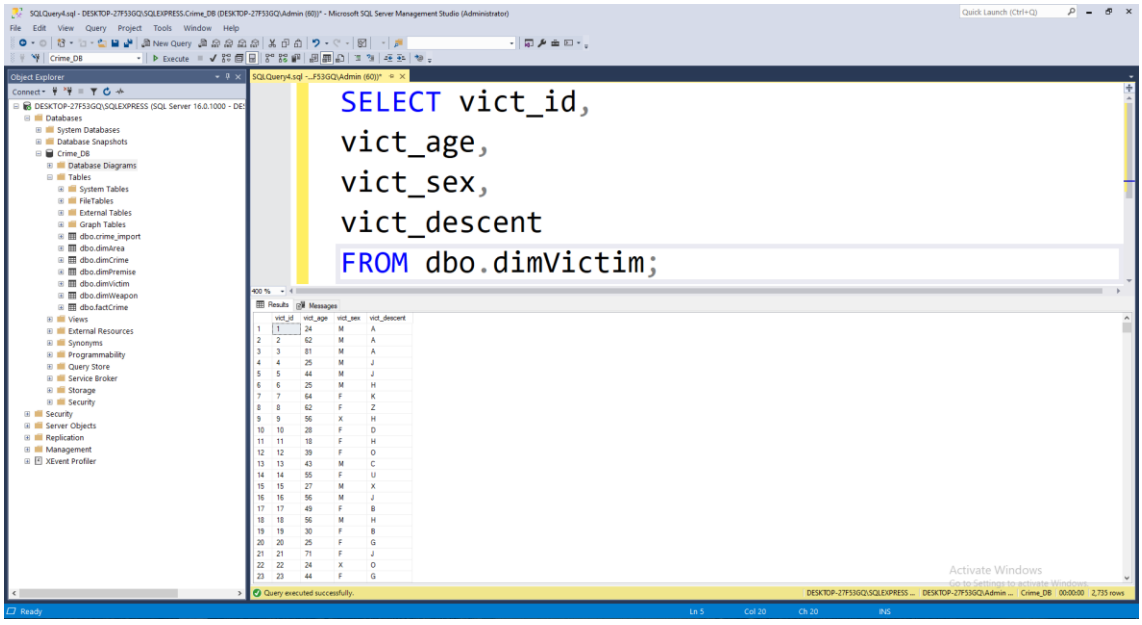
The third query is located in Appendix 39, along with its output. It selects key fields from `dimPremise`, such as `premis_id`, which was included to preserve the relationship with the fact table, while `premis_desc` was retained for visualization purposes.



Screenshot 36: "dimPremise: key fields returned"

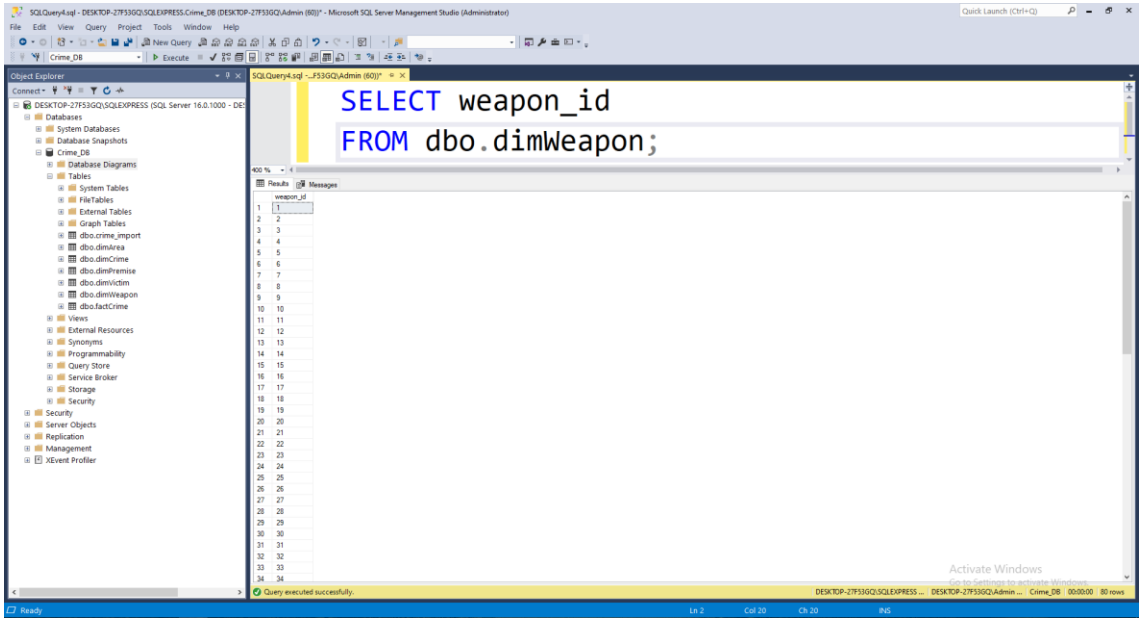
The fourth query, shown in Appendix 40 along with its output, retrieves key fields from `dimVictim`. The field `vict_id` was included to preserve the relationship with `factCrime`, while `vict_age`, `vict_sex`, and `vict_descent`

were retained for visualization purposes.



Screenshot 37: “dimVictim: key fields returned”

The fifth query, shown in Appendix 41 along with its output, retrieves key fields from `dimWeapon`. The field `weapon_id` was included to preserve the relationship with the fact table and will also be used in visualizations.

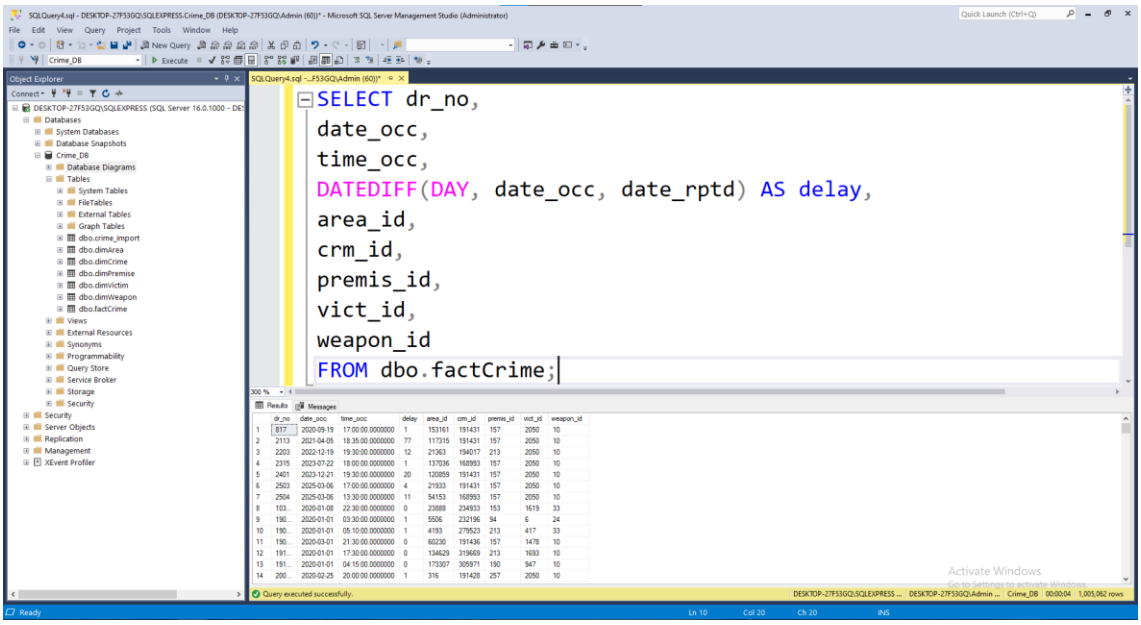


Screenshot 38: “dimWeapon: key fields returned”

The sixth query retrieves key fields from `factCrime`, such as `dr_no` (primary key), and `area_id`, `crm_id`, `premis_id`, `vict_id`, and `weapon_id` (foreign

keys), to preserve the relationship with the dimension tables, while `date_occ`, `time_occ`, and `delay` were included for visualization purposes. `delay` calculates the number of days it took for an incident to be reported after its occurrence. The query and its output are shown in Appendix 42, where:

- the `DATEDIFF` function returns the difference between `date_occ` and `date_rptd`, measured in days.



Screenshot 39: "factCrime: key fields returned"

3.13.2. Exploratory Queries for Validation and Testing:

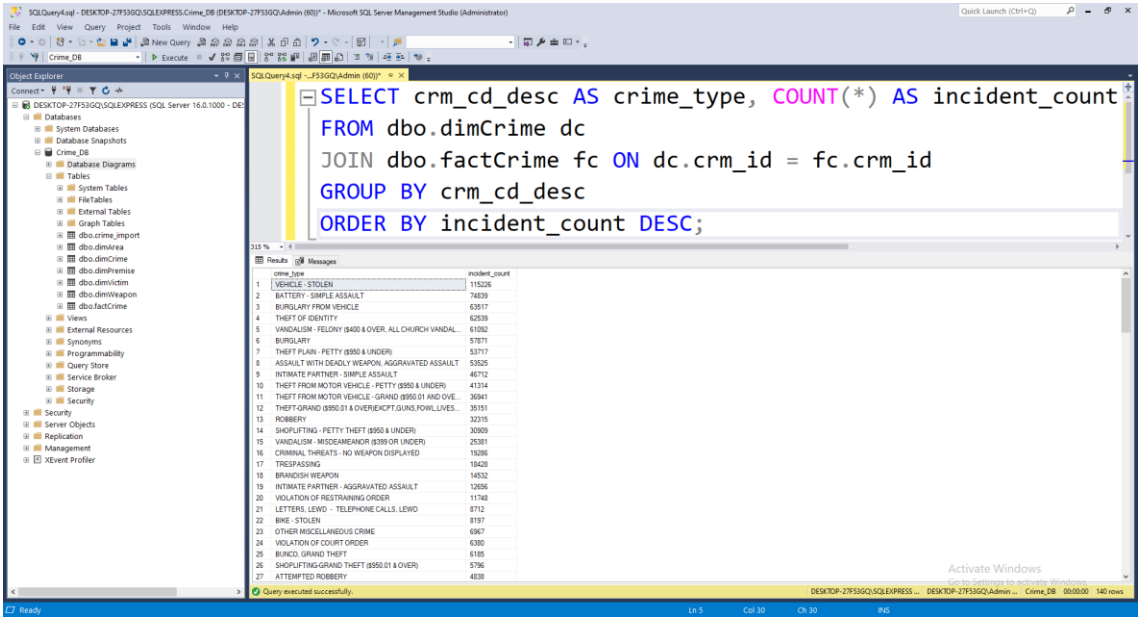
In the current subsection, the first query that we assessed and is about the crime types with the most incidents is shown in Appendix 37, in which:

- the `dimCrime` table joins the `factCrime` table using their common field `crm_id`, which is a primary key in the `dimCrime` table and a foreign key in the `factCrime` table.
- `GROUP BY` is used to group rows that have the same values in the `crm_cd_desc` column.
- after all rows are grouped, `SELECT` chooses the columns that will be returned in the results, in our case the
  1. the `crm_cd_desc` column, aliased as `crime_type` and
  2. `incident_count` column, which uses the aggregate function `COUNT(*)`, which in turn counts the number of rows for all different

`crm_cd_desc` values.

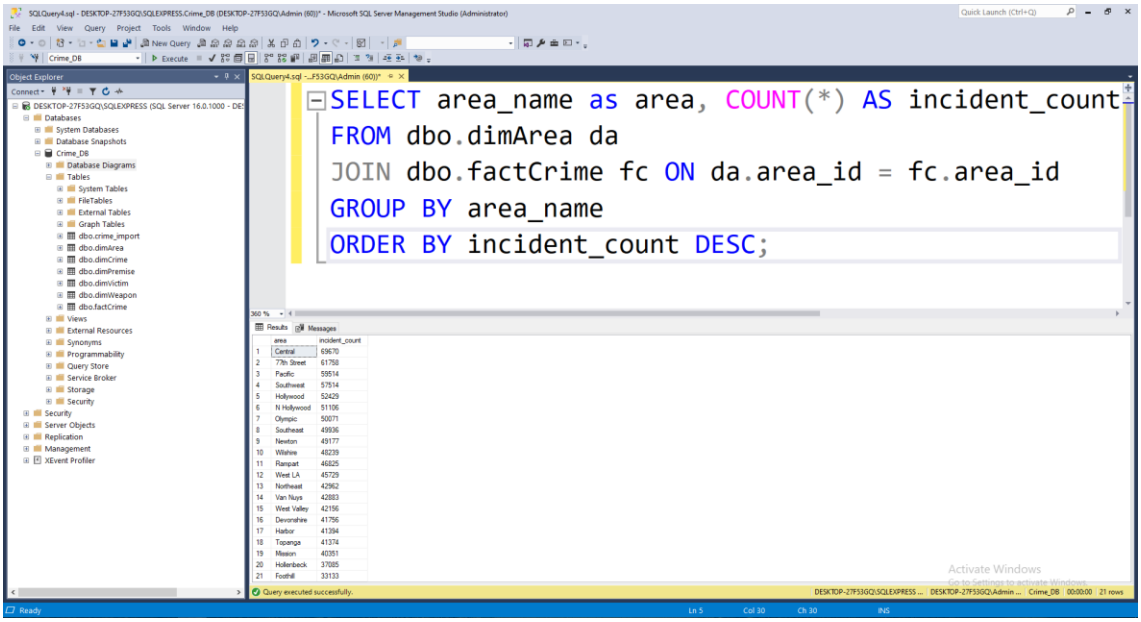
- `ORDER BY` clause sorts the results by the `incident_count` column in descending order, meaning that crime types with the highest number of incidents are going to appear on the top of our query's result.

The query output is shown in Appendix 43.



Screenshot 40: "Incident count by crime type"

The second query, presented in Appendix 44 along with its output, retrieves the area names with the most crimes and is analyzed below. Its structure and logic are similar to those of the first query.

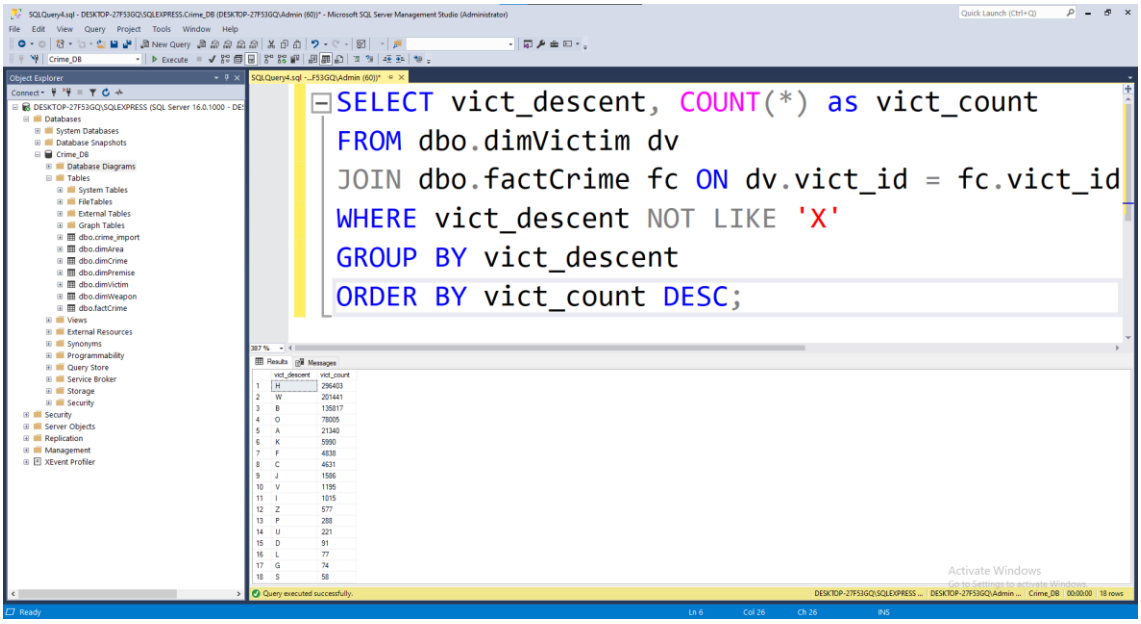


Screenshot 41: "Incident count by area"

The third query that we tested out is about the number of victims per victim descent. The code is shown in Appendix 39. This command is similar to the two commands we wrote previously, apart from the fact that it comprises a parameter in the:

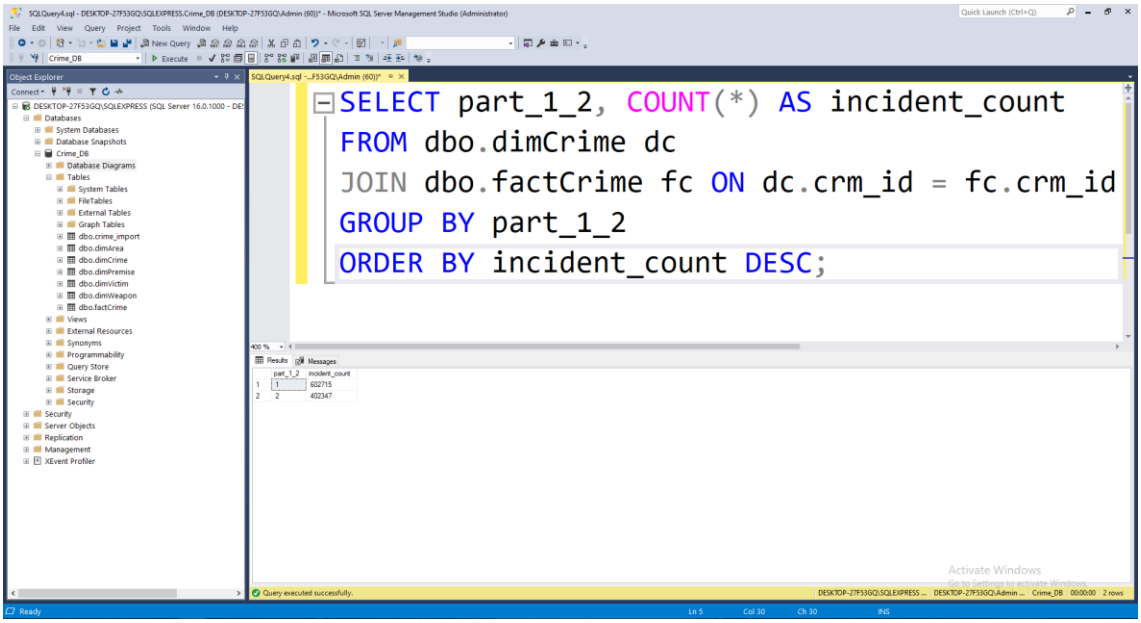
- `WHERE` clause. The parameter states that the result will exhibit all descents that are not equal to unknown ('X').

The query output is shown in Appendix 45.



Screenshot 42: "Victim count by descent"

The fourth query helped us uncover which offense classification has the most incidents, Part 1 or Part 2. The code is presented in Appendix 46, along with its output.

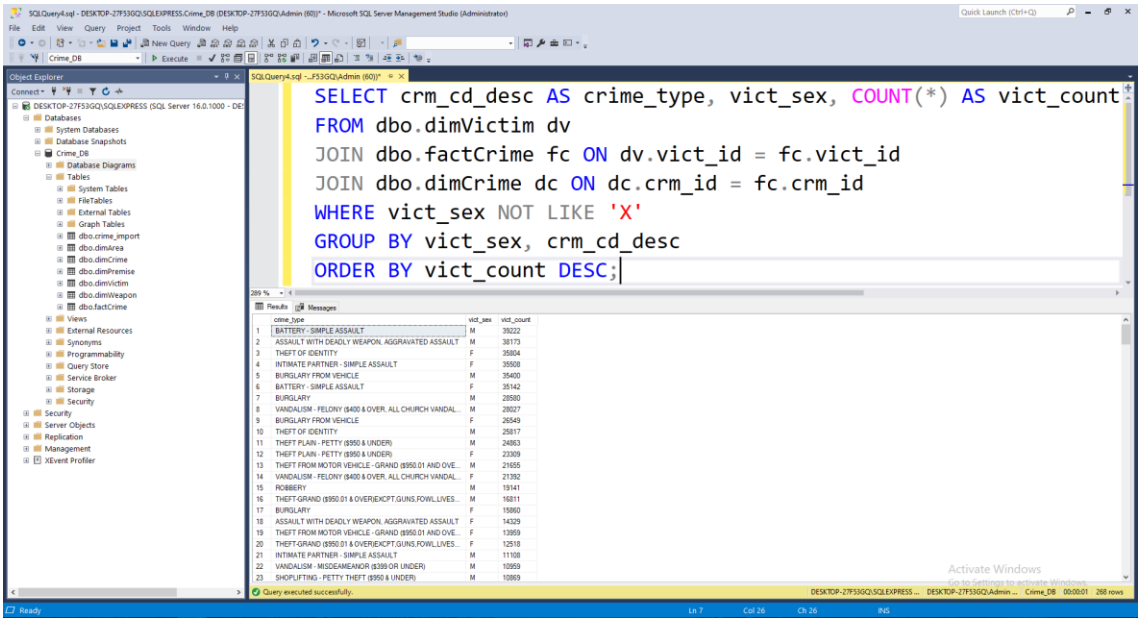


Screenshot 43: "Incident count by part 1 and 2"

The fifth query reveals the number of victims separately for females and males for each crime type. We achieve this result by executing the command shown in Appendix 47, where:

- the `WHERE` clause is used in the same manner as the third query. We are using it to filter out the number of victims with unknown sex ('X') per crime type.

The query output is shown in Appendix 47.

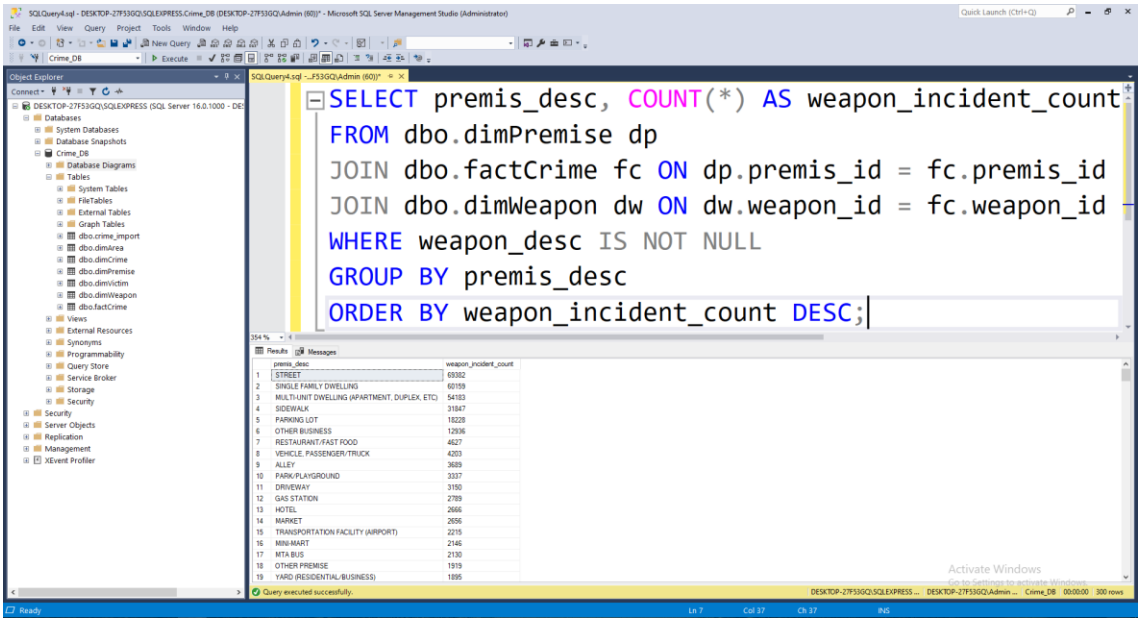


Screenshot 44: “Victim count by sex and crime type”

The sixth query returns all premise types with at least one armed incident and present the number of armed incidents per type. The code is found in Appendix 48, where:

- the `WHERE` clause is used in the same manner as in the third query. It is used in order to exclude from the results all records that do not actually own a weapon.

The query output is shown in Appendix 48.

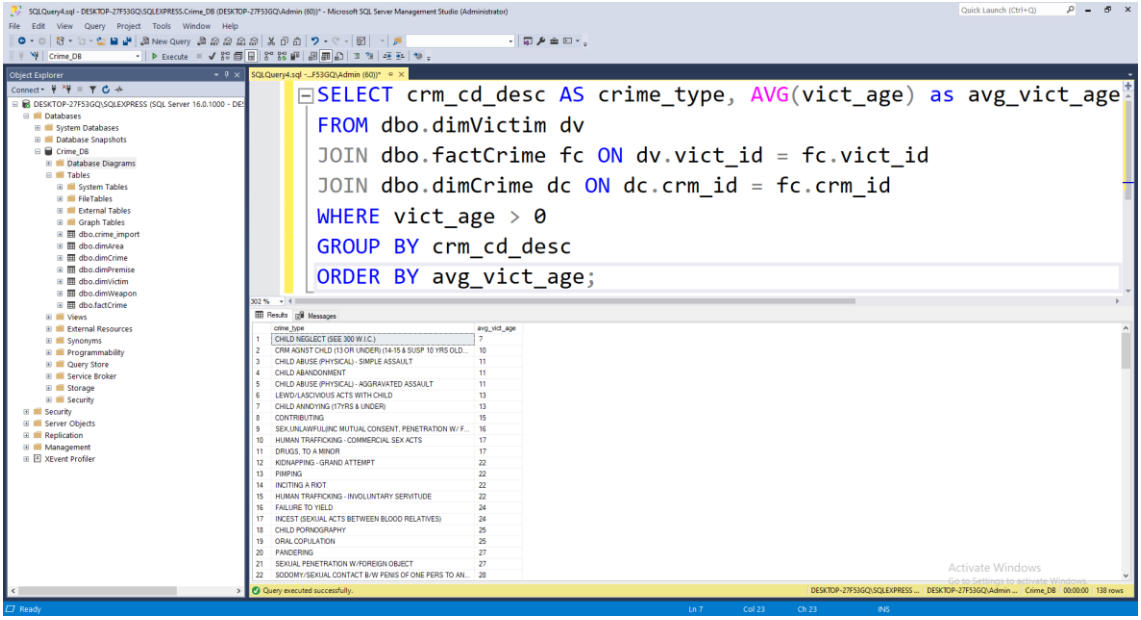


Screenshot 45: "Weapon incident count by premise"

The seventh query we examined reveals the average age of victims by crime type. The code is located at Appendix 49, where:

- in the `WHERE` clause, all victims with `vict_age` = 0 are excluded from the average aggregation.

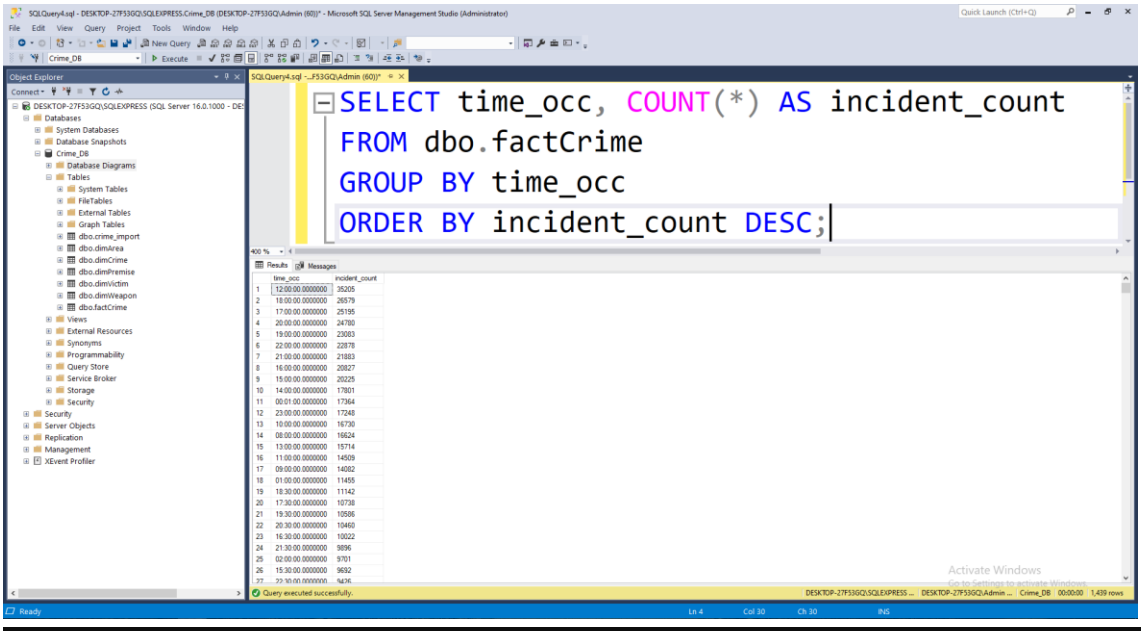
The query output is shown in Appendix 49.



Screenshot 46: "Average victim age by crime type"

The eighth query, presented in Appendix 50 along with its output, illustrates the

number of incidents per time of occurrence.

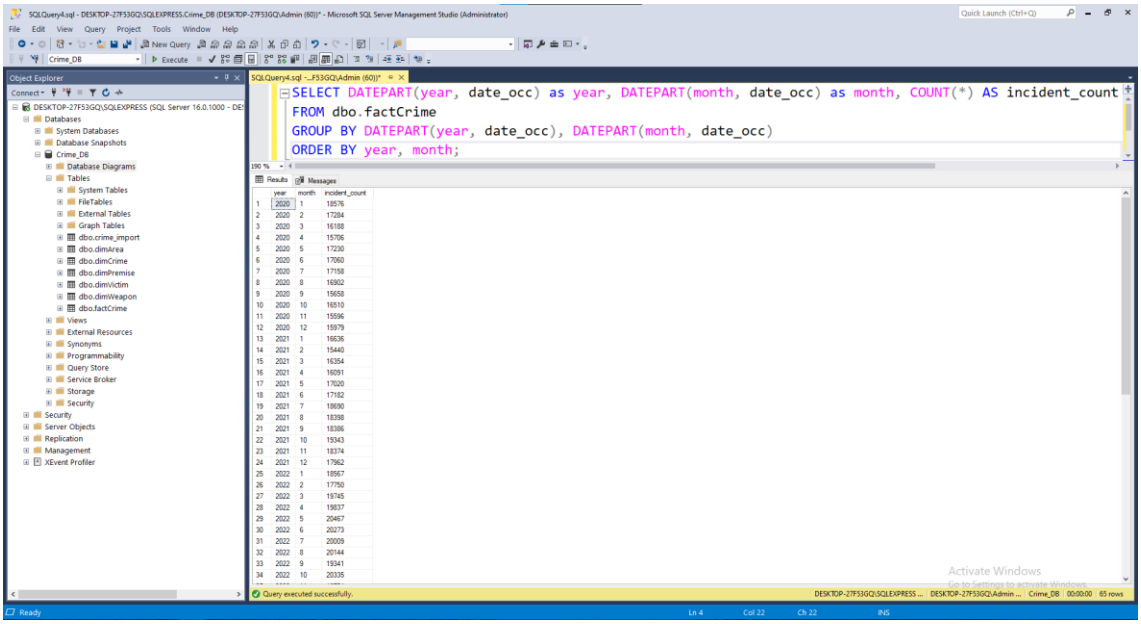


Screenshot 47: "Incident count by occurrence time"

Our ninth query returns the number of crime incidents per combination of year and month. The command can be found in Appendix 51, where:

- the first DATEPART function returns the year part of the date that crimes occurred while the second DATEPART returns the month part.

The query output is shown in Appendix 51.

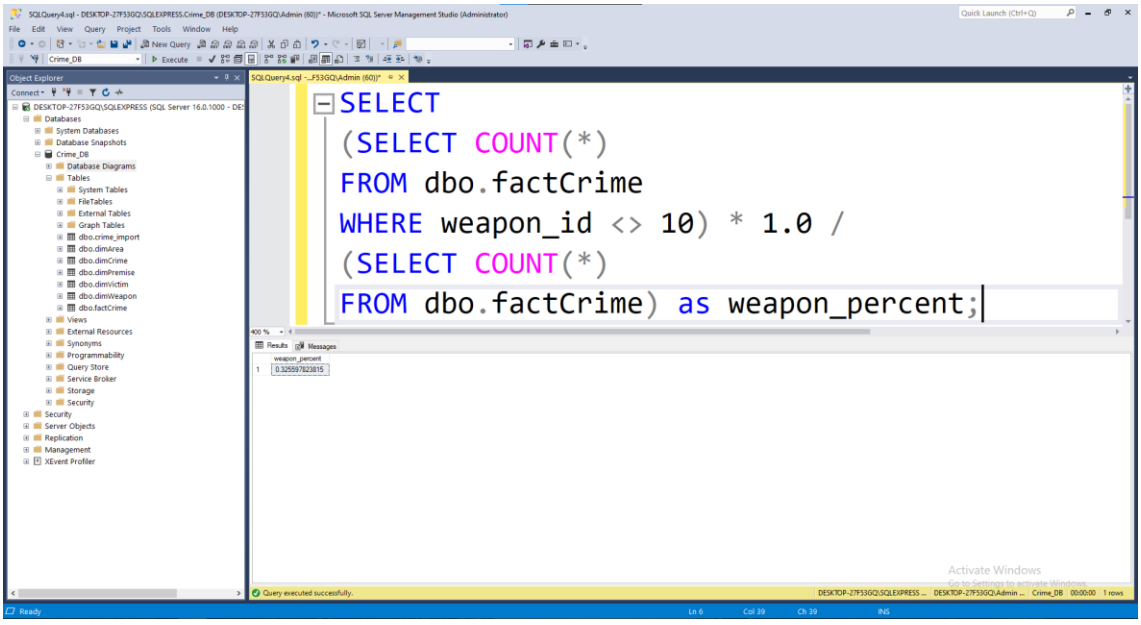


Screenshot 48: "Incident count by year-month"

The tenth query helped us uncover the percentage of crimes that involves a weapon. We will achieve this by executing the code shown in Appendix 52, where:

- the first subquery selects the number of crime incidents where the `weapon_id` is not equal to 10 (note that `weapon_id` = 10 corresponds to null values in both `weapon_cd`, and `weapon_used_cd`).
- the subquery is multiplied by 1.0 to ensure a float/decimal result during the division with the
- second subquery, which acquires the total number of incidents,
- Finally, the result of the division is aliased as `weapon_percent` and presented through the main query.

The query output is shown in Appendix 52.

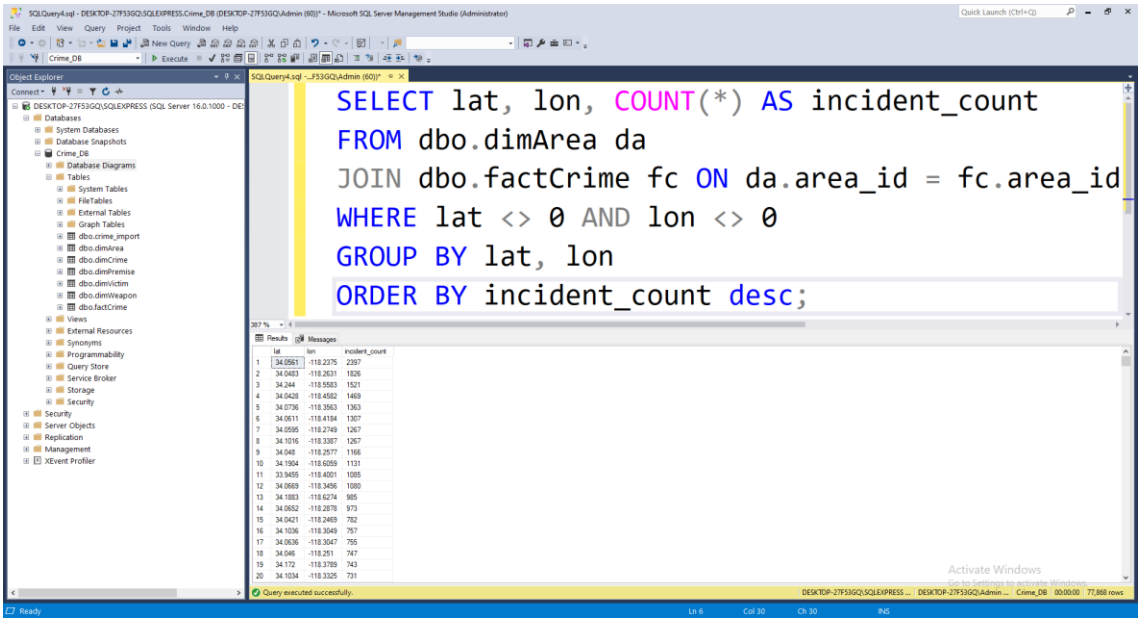


Screenshot 49: "Percentage of armed incidents"

The eleventh query has to do with the number of incidents per unique combination of known coordinates, latitude and longitude. The command is shown in Appendix 53, where:

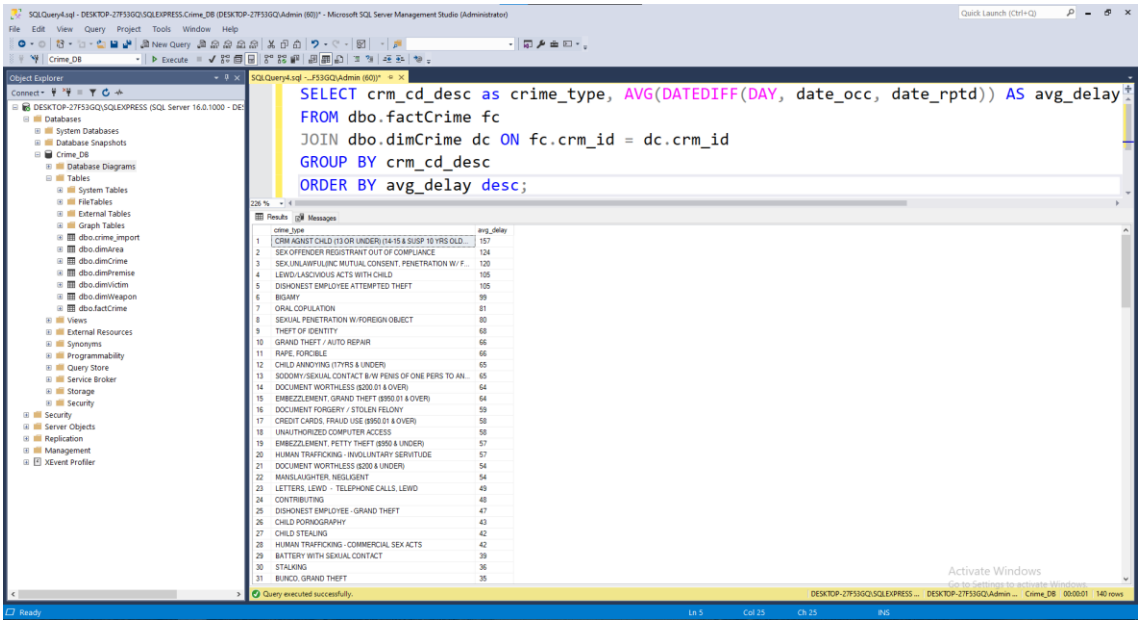
- in the `WHERE` clause, we use a filter to exclude all incidents which hold a combination of `lat` = 0, `lon` = 0. When a record has a value of zero on one or both of the aforementioned, we can be 100% sure that the specific incident has unknown coordinates.

The query output is shown in Appendix 53.



Screenshot 50: “Incident count per lat–lon combination”

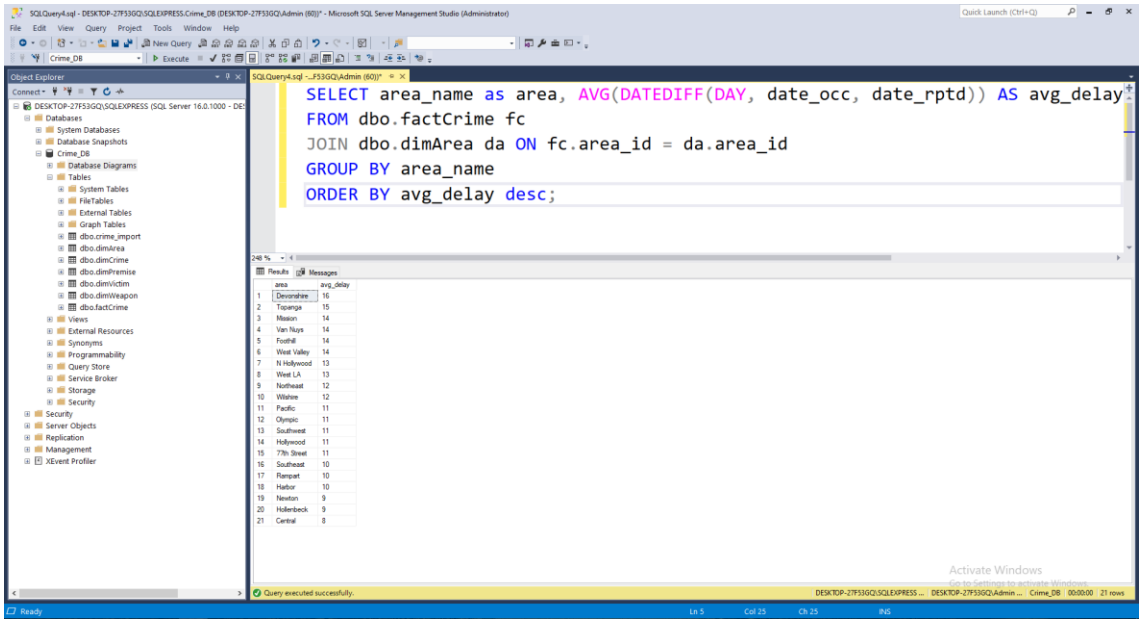
The twelfth query calculates the average number of days it takes for an incident of a specific crime type to be reported after its occurrence. The corresponding code is provided in Appendix 54.



Screenshot 51: “Average delay by crime type”

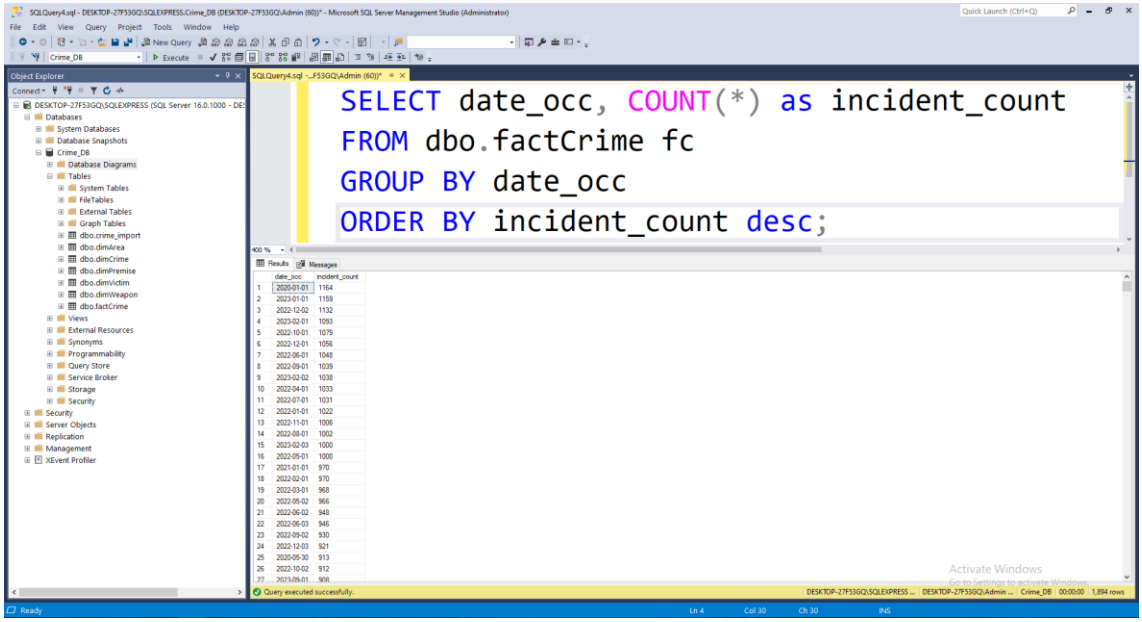
The query’s output is included in Appendix 54.

The thirteenth query calculates the average number of days it takes for an incident in a specific area to be reported after it occurred. The command is shown in Appendix 55 along with its output.



Screenshot 52: "Average delay by area"

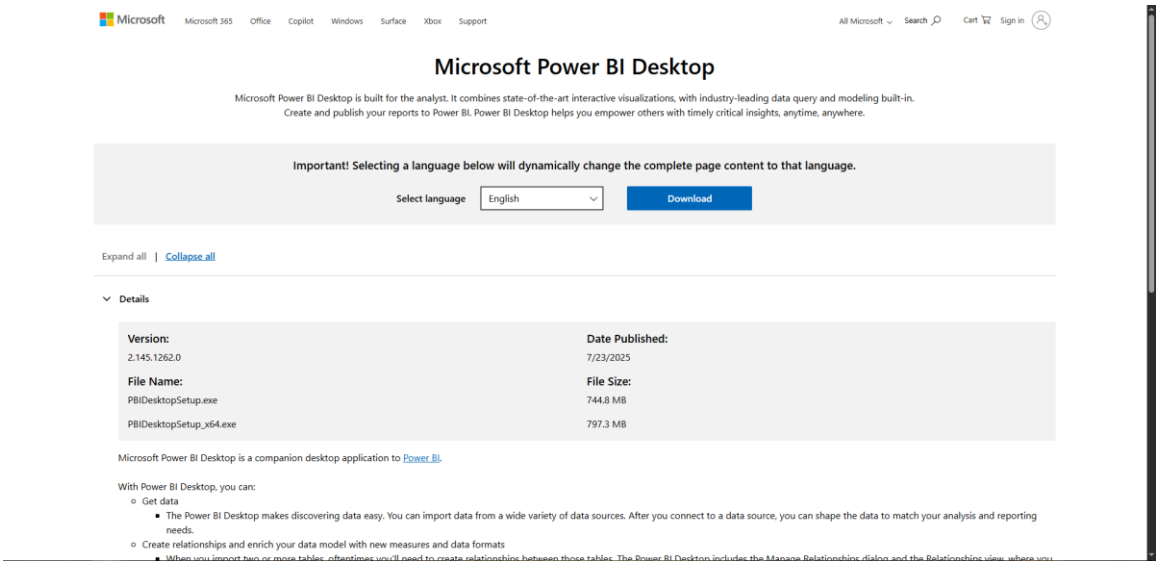
The fourteenth query retrieves the number of incidents per date of occurrence. The command is shown in Appendix 56 along with its output.



Screenshot 53: "Number of incidents by date of occurrence"

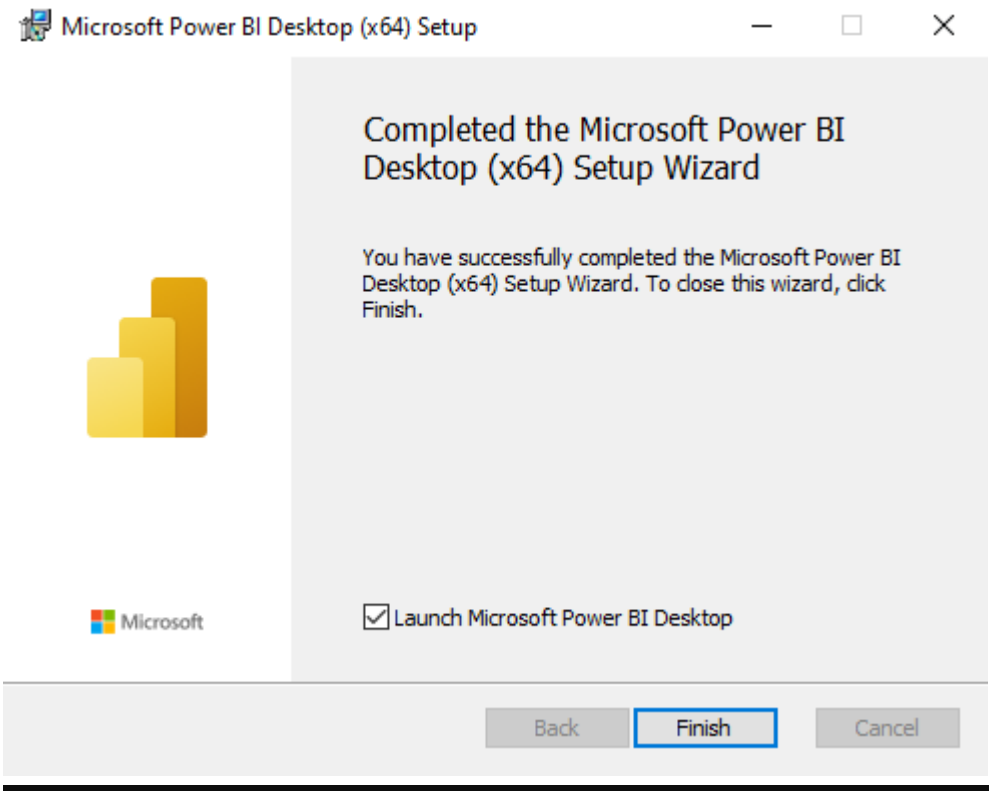
3.14. Installation of Power BI Desktop:

Since we had completed the construction and evaluation of SQL queries — some of which would later be imported into Power BI — we proceeded with downloading and installing Power BI Desktop. Power BI Desktop is a free, downloadable application that allows users to connect to, transform, model and visualize data, creating interactive reports and dashboards. It serves as the primary tool for building reports and dashboards before publishing them to the Power BI service for sharing and collaboration (we will not be using Power BI service in our project).

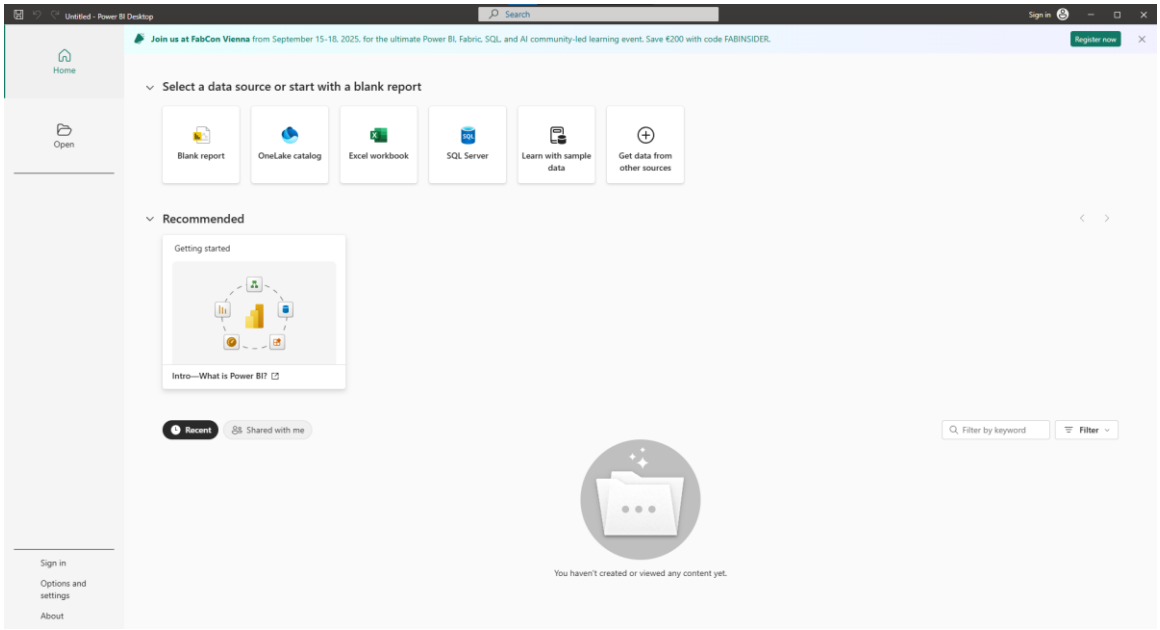


Screenshot 54: "Downloading Power BI Desktop"

Immediately after downloading, it was time to install the software. This took place in a small number of steps, which are shown in Appendix 57.



Screenshot 55: “Power BI Desktop installed”

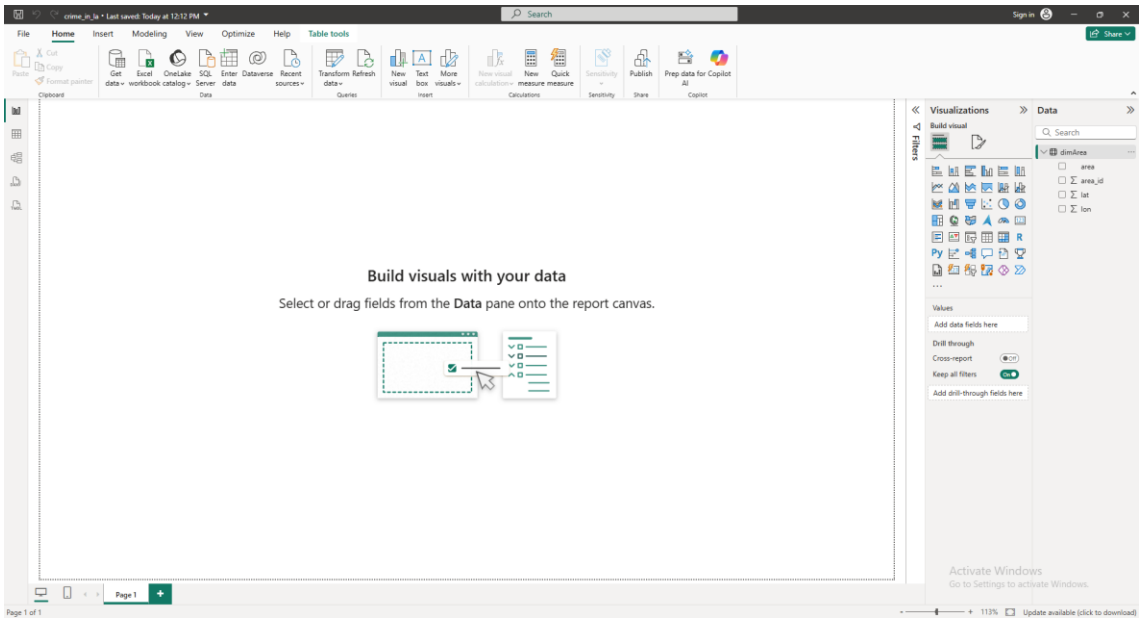


Screenshot 56: “Power BI running”

3.15. Connecting Database to Power BI via SQL Queries:

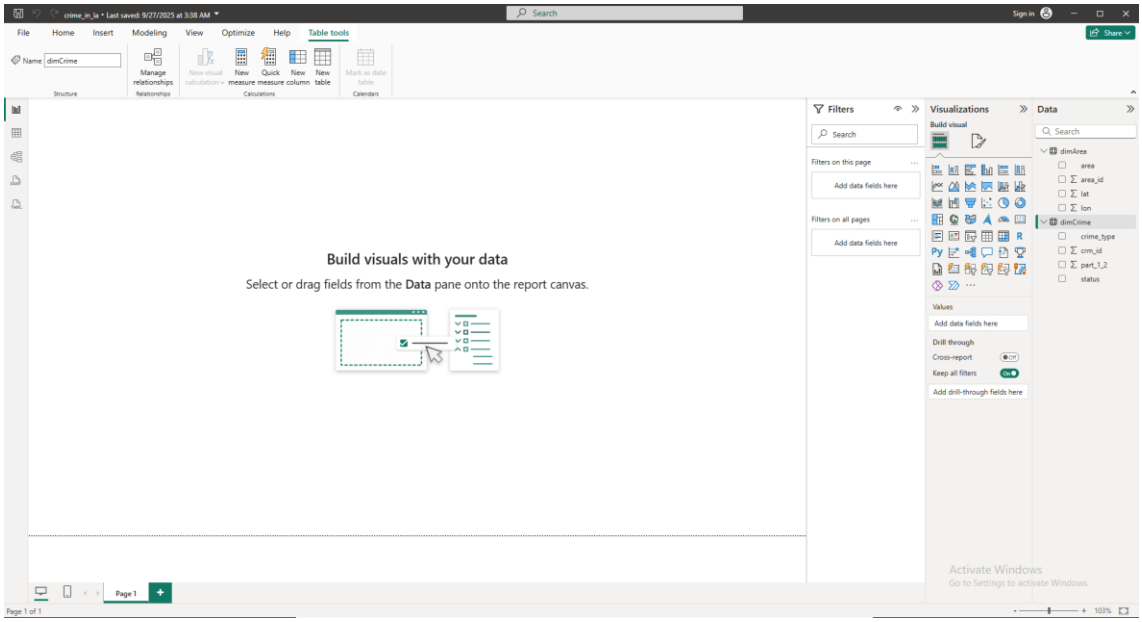
The current section of our project focused on integrating our database, `Crime_DB`, into Power BI's environment using the six SQL Queries we had previously written and evaluated in the "Core Queries for Star Schema" section. Each query was responsible for the importation of a specific dataset into Power BI, helping us achieve an efficient visualization of the information by leaving unnecessary data out of the picture.

Assuming that our Power BI was open from the previous section of our thesis, we established the connection and brought the content of the first query into Power BI by following the steps presented in Appendix 58.

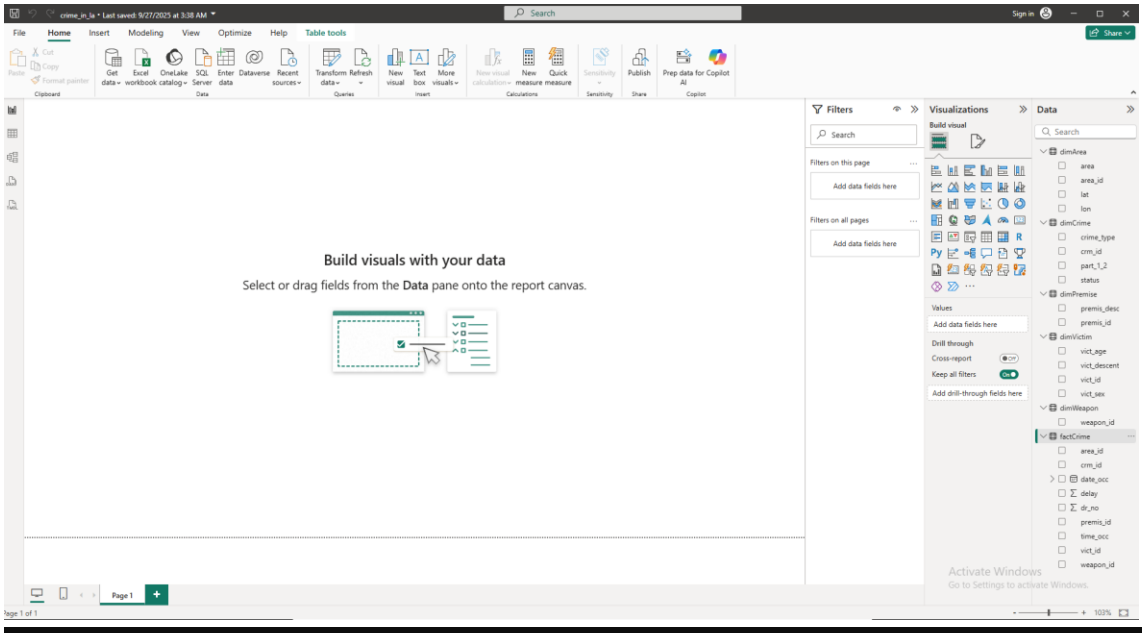


Screenshot 57: "Server's database and Power BI connected — first query's data imported"

Regarding second query's dataset, a similar approach was followed in order for the dataset to be inserted into Power BI. The relative steps are shown in Appendix 59. The importation procedure was exactly the same for the remaining queries of the "Core Queries for Star Schema" section.

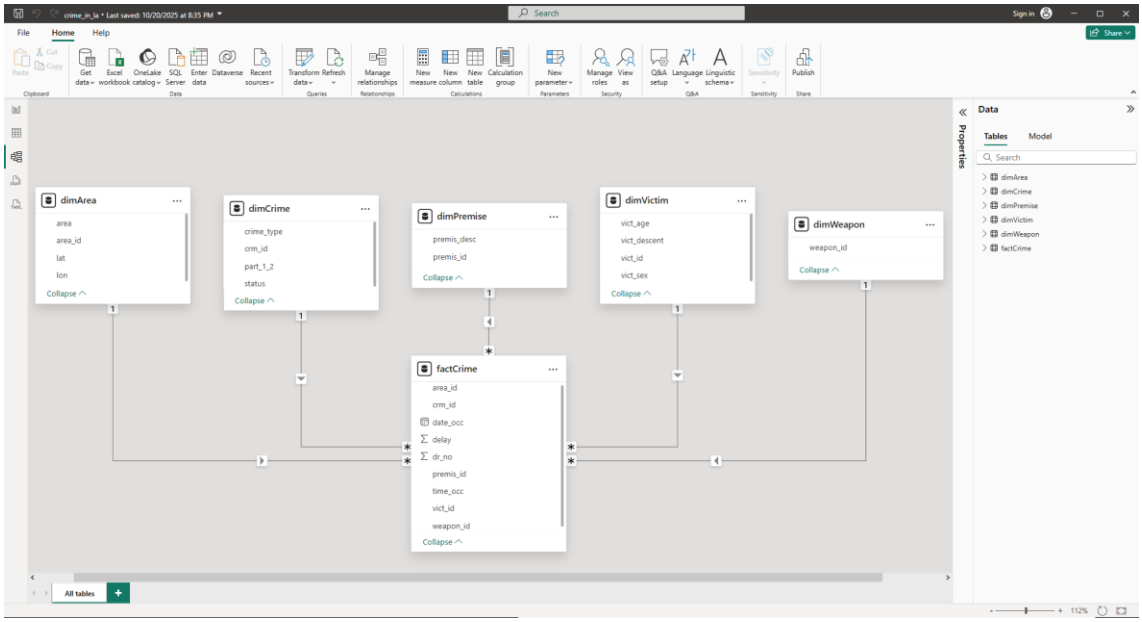


Screenshot 58: “Second query’s data imported”



Screenshot 59: “Data from all tables imported”

To examine the structure of our star schema in Power BI after importing the data, we followed the instructions outlined in Appendix 60.



Screenshot 60: “Model view of the star schema”

3.16. Creation of calculated columns and measures with DAX:

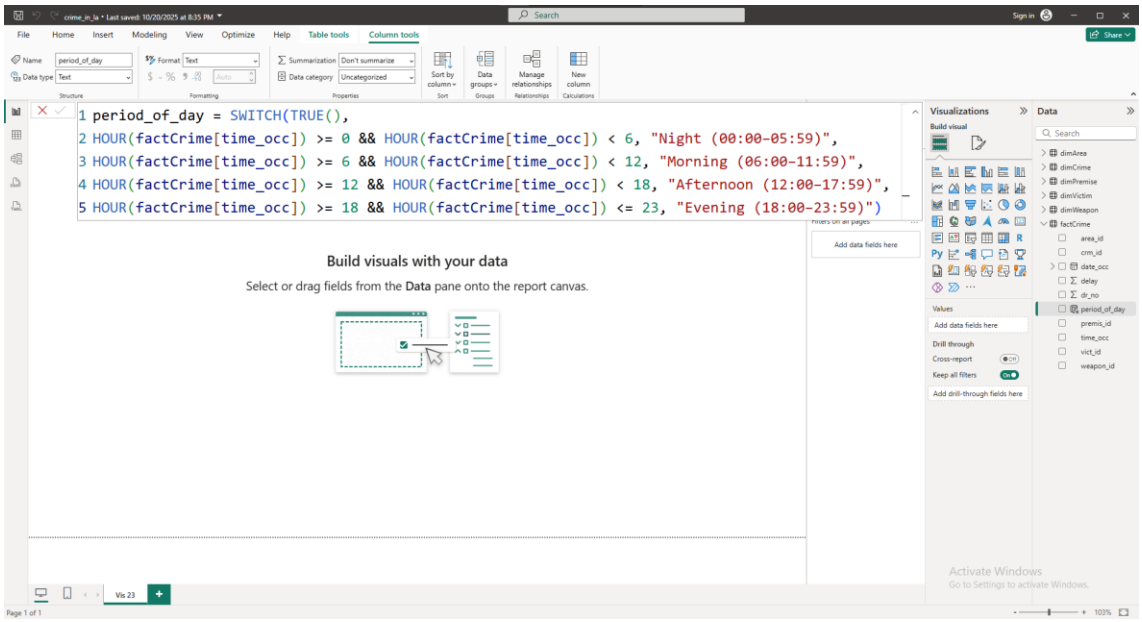
This section focused on creating calculated columns and measures, which would eventually be used in visualizations in the upcoming sections. A calculated column is a new column added to a table using a DAX expression, evaluated row by row, while a calculated measure is a dynamic aggregation computed at query time, typically used in visuals to summarize data. The first calculated column that was created with the use of DAX — Data Analysis Expressions, a formula language consisting of functions and operators used to construct calculations in Power BI, Analysis Services, and Power Pivot in Excel — extracts the time from `time_occ` and converts it into a period of day category, out of four categories in total, including morning, afternoon, evening and night. In order to create this column, the steps located at Appendix 61 were executed, in which:

- `period_of_day` is the name of the new column.
- the `HOUR` function returns the hour as a number from 0 (12:00:00 A.M.) to 23 (23:59:59 P.M.).
- `TRUE()` returns the logical value “TRUE”.
- `SWITCH` is frequently used instead of multiple nested if statements. Its purpose is to find the first expression that is true out of the defined four and return the corresponding result. When the first true

expression is found, it stops there and moves on to evaluating `time_occ`'s value of the next `factCrime`'s record. As mentioned, we have four expressions/categories. These are

1. "Night (00:00–05:59)", which consists of hours between 00:00 and 05:59,
2. "Morning (06:00–11:59)", which consists of hours between 06:00 and 11:59,
3. "Afternoon (12:00–17:59)", which consists of hours between 12:00 and 17:59,
4. "Evening (18:00–23:59)", which consists of hours between 18:00 and 23:59.

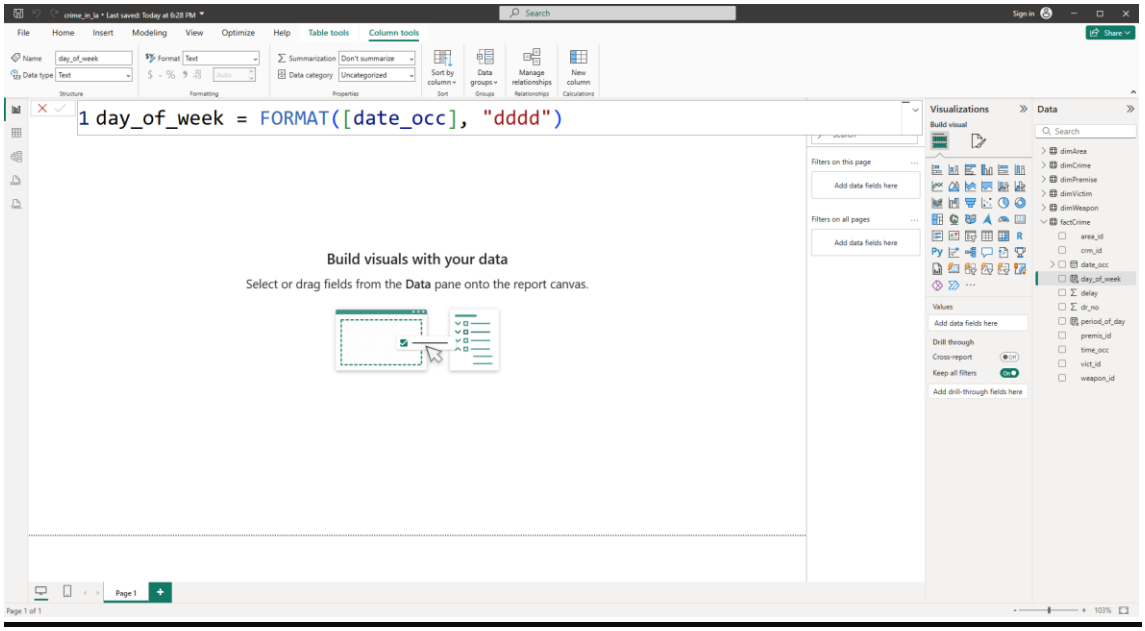
For better understanding of how the code truly works, here's a brief example. If there's a record with 18:15:00 as the time of occurrence, `SWITCH` classifies 18:15:00 in the "Evening (18:00–23:59)" category.



Screenshot 61: "Period of day column created"

The second calculated column extracts the day from `date_occ` column and converts it into corresponding day of the week, ranging from Monday to Sunday. The steps that created this column are listed in Appendix 62, in which:

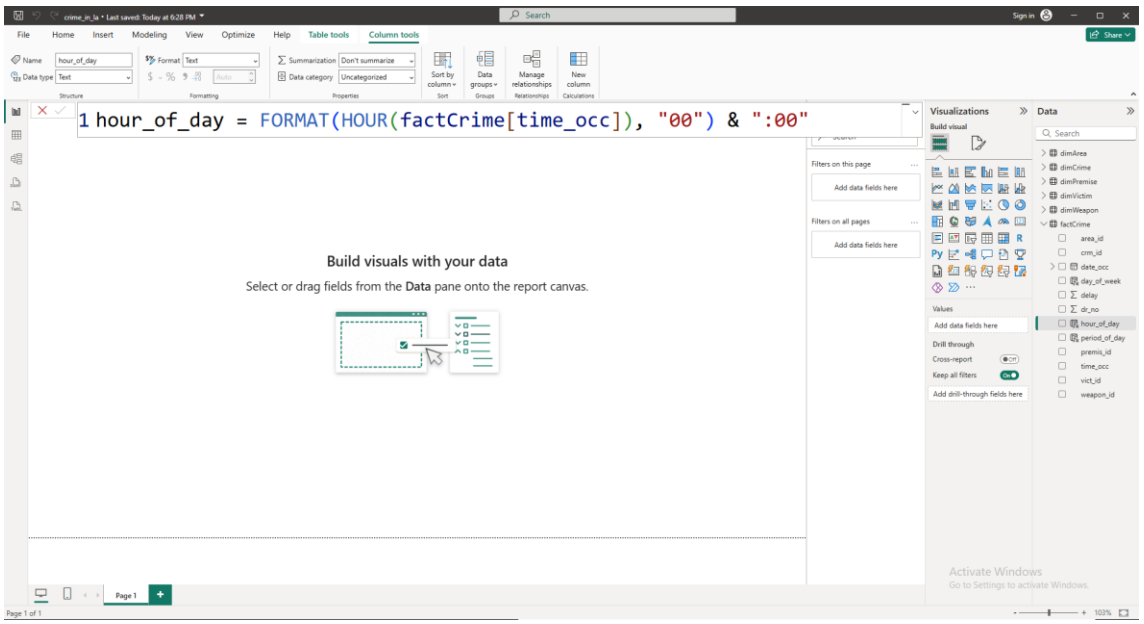
- the `FORMAT` function converts occurrence date (value) to a verbal representation of the day of the week.



Screenshot 62: "Day of week column created"

The third calculated column extracts the hour from `time_occ` column and converts it into corresponding hour of the day, ranging from 00:00 to 23:59. The steps that created this column are listed in Appendix 63, in which:

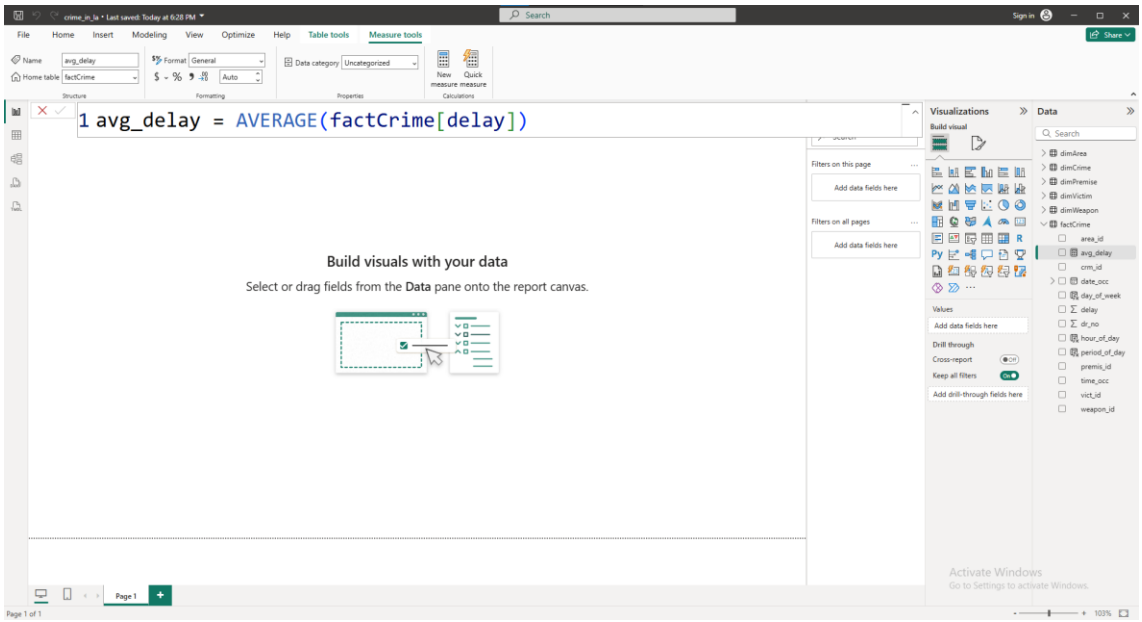
- the `HOUR` function extracts the hour from `time_occ` column.
- the `FORMAT` function converts the extracted hour into a string with the format HH:00.



Screenshot 63: "Hour of day column created"

The first calculated measure returns the average reporting delay of our dataset. To create it, the steps shown in Appendix 64 were executed, where:

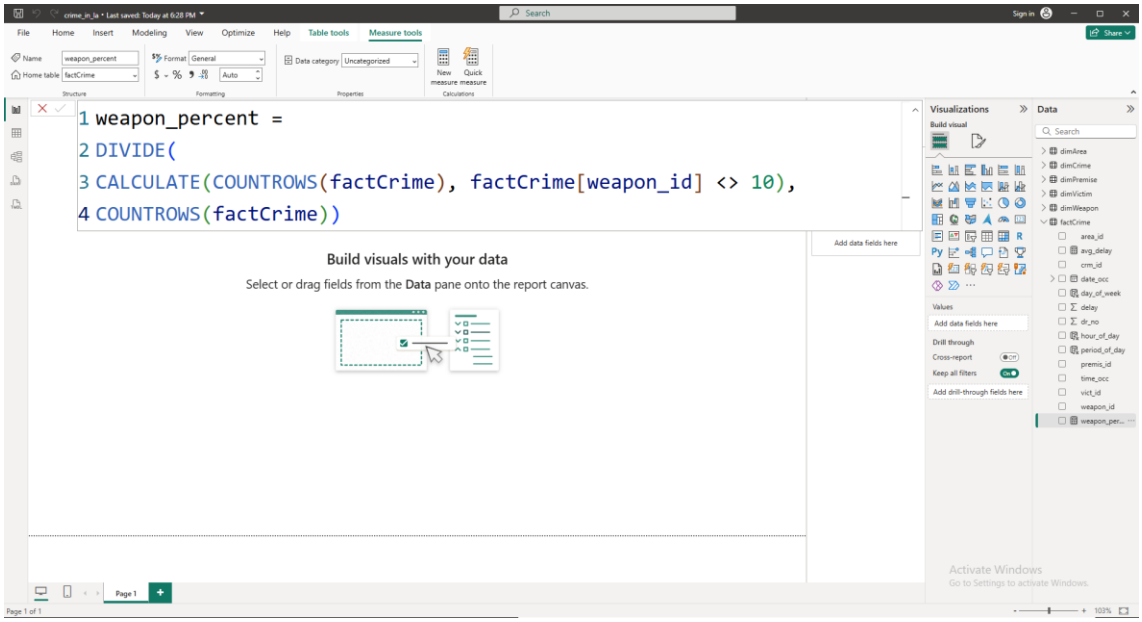
- `avg_delay` is the name of the new measure.
- the `AVERAGE` function calculates the average of `delay` column.



Screenshot 64: "Average delay measure created"

The second calculated measure retrieves the percentage of crimes that involve a weapon. The steps that created it are listed in Appendix 65, where:

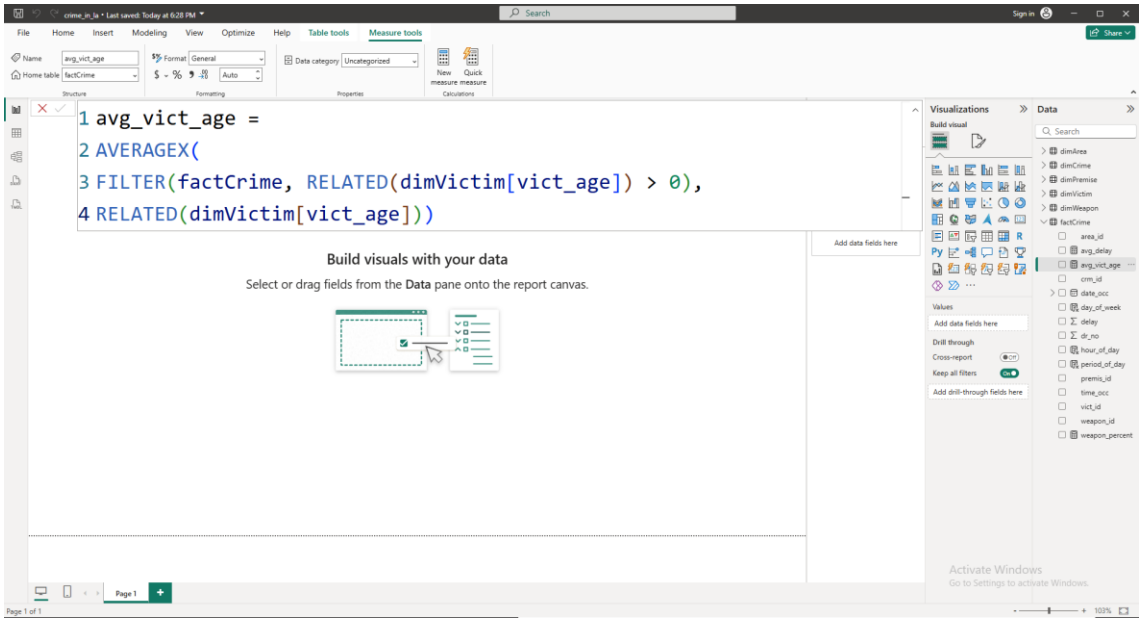
- the `CALCULATE` function contains two components:
  1. the `COUNTROWS` function counts the number of records in `factCrime`.
  2. the filter `[weapon_id] <> 10` excludes all records where no weapon was involved (`weapon_id` = 10).
- the `DIVIDE` function divides the result of `CALCULATE` by the total number of records.



Screenshot 65: "Weapon percent measure created"

The third calculated measure returns the average victim age of our dataset. To create it, the steps shown in Appendix 66 were carried out, where:

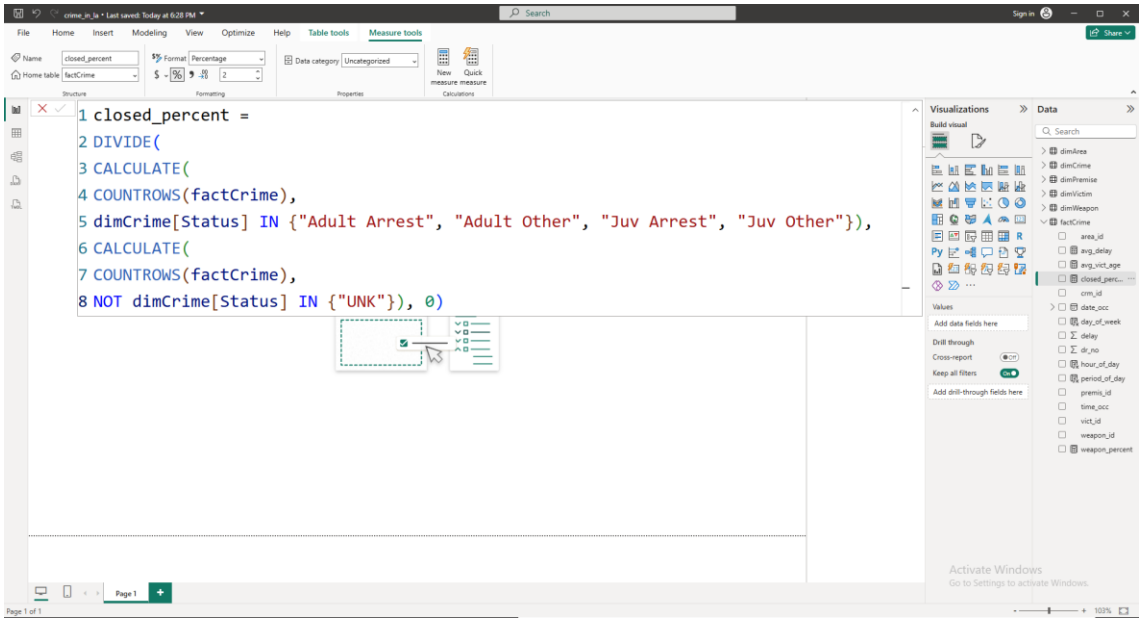
- the `FILTER` function creates a temporary table that contains only the records from `factCrime` where `vict_age` of the corresponding victim is greater than 0.
- the `RELATED` function retrieves the value of `vict_age` from `dimVictim` for each record in `factCrime`, using the relationship between the two tables.
- the `AVERAGEX` function calculates the average of the values returned by the `RELATED` function, iterating over each record in the temporary table produced by `FILTER`.



Screenshot 66: "Average victim age measure created"

The fourth calculated measure retrieves the percentage of crimes that are closed/resolved. The steps that created it are listed in Appendix 67, where:

- the first `CALCULATE` function contains two components:
    1. the `COUNTROWS` function counts the number of records in `factCrime`.
    2. the filter `dimCrime[Status] IN {"Adult Arrest", "Adult Other", "Juv Arrest", "Juv Other"}` includes only records with a closed/resolved status (e.g., Adult Arrest).
- the second `CALCULATE` function contains two components:
    1. the `COUNTROWS` function counts the number of records in `factCrime`.
    2. the filter `NOT dimCrime[Status] IN {"UNK"}` excludes only records with an unknown status (UNK).
- the `DIVIDE` function divides the result of the first `CALCULATE` by the result of the second `CALCULATE`. If the result of the denominator equals 0, the function returns 0 to avoid a division error.



Screenshot 67: “Closed percent measure created and formatted”

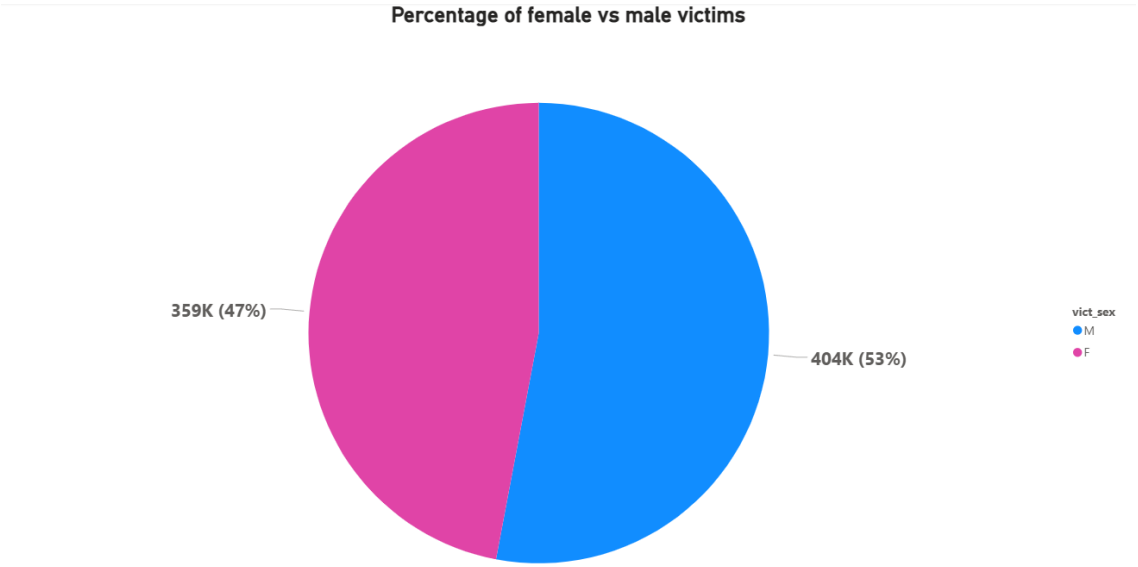
4. Data Analysis and Interpretation of Findings

4.1. Visualization of forensic data:

For many individuals, the creation of data visualizations is considered one of the most intriguing aspects of data science – if not the most interesting. Data visualization refers to the practice of presenting information and datasets through graphical formats such as charts, graphs, and maps. These visual elements make it easier to identify trends, highlight anomalies, and recognize underlying patterns in the data. Beyond analysis, visualization also serves as a clear communication tool, allowing complex information to be conveyed to non-technical audiences in a simple and intuitive way. In the context of large-scale or complex datasets, effective visualization becomes essential for enabling deeper insights and supporting data-driven decision-making.

Our first visualization, which uses `dimVictim`'s and `factCrime`'s datasets, reveals which sex is the most frequent victim in our dataset, men or women. To create it, we followed the instructions shown in Appendix 68.

After carrying out each step with attention to detail, we plotted a well-designed chart, which shows that our dataset has more male victims than females. Specifically, 53% of victims are males (403,878 individuals) while the rest 47% are females (358,580 individuals).

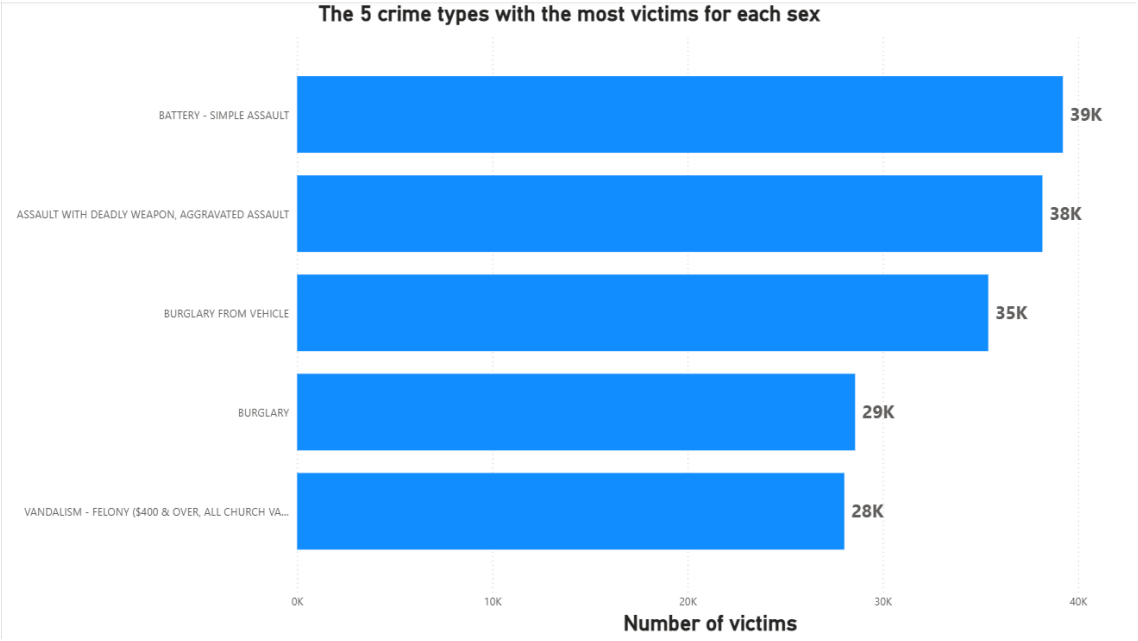


Screenshot 68: “Incidents by sex”

The second visualization that we created uses `dimCrime`’s, `dimVictim`’s and `factCrime`’s datasets. Its objective is to disclose the five crime types with the highest number of male victims and the five crime types with the highest number of female victims. The detailed guide presented in Appendix 69 helped us create it.

By looking at the bar chart we created, we note that the crime type with the most male victims is BATTERY – SIMPLE ASSAULT, which accounts for over 39,000 male victims. Our complete results, ordered by the victim count in descending order, are:

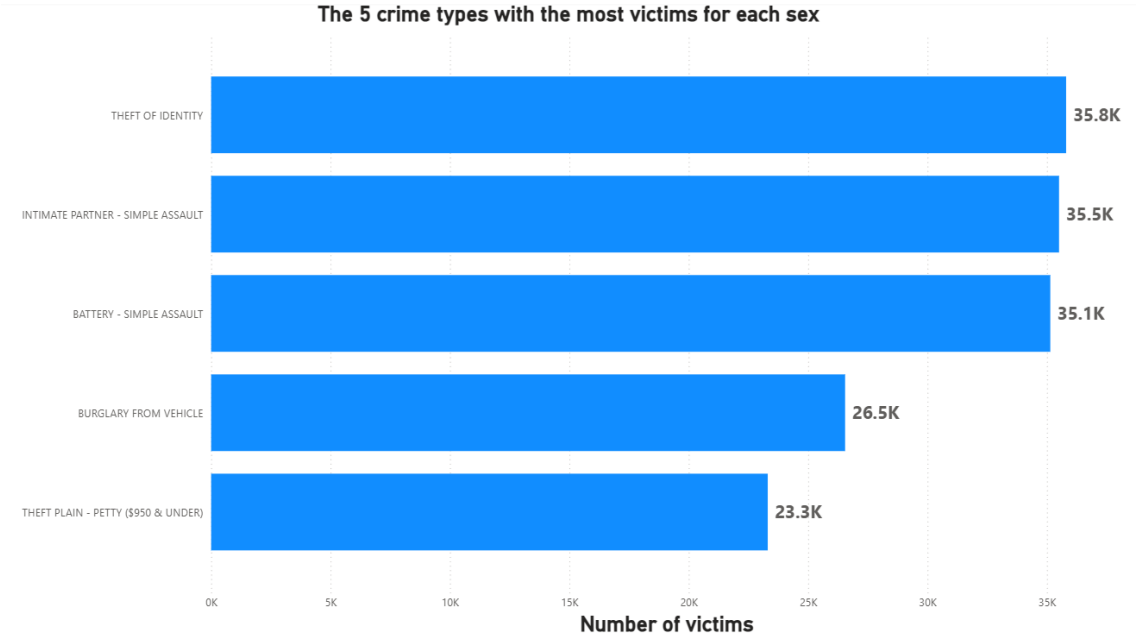
1. BATTERY — SIMPLE ASSAULT (39,222 victims),
2. ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT (38,173 victims),
3. BURGLARY FROM VEHICLE (35400 victims),
4. BURGLARY (28,580 victims),
5. VANDALISM — FELONY ($400 & OVER, ALL CHURCH VANDALISMS) (28,027 victims).



Screenshot 69.1: "Incidents by crime type for males: Top 5"

In contrast to males, females tend to be more victimized in the THEFT OF IDENTITY category, which account for over 35,000 victims. Our complete results below are:

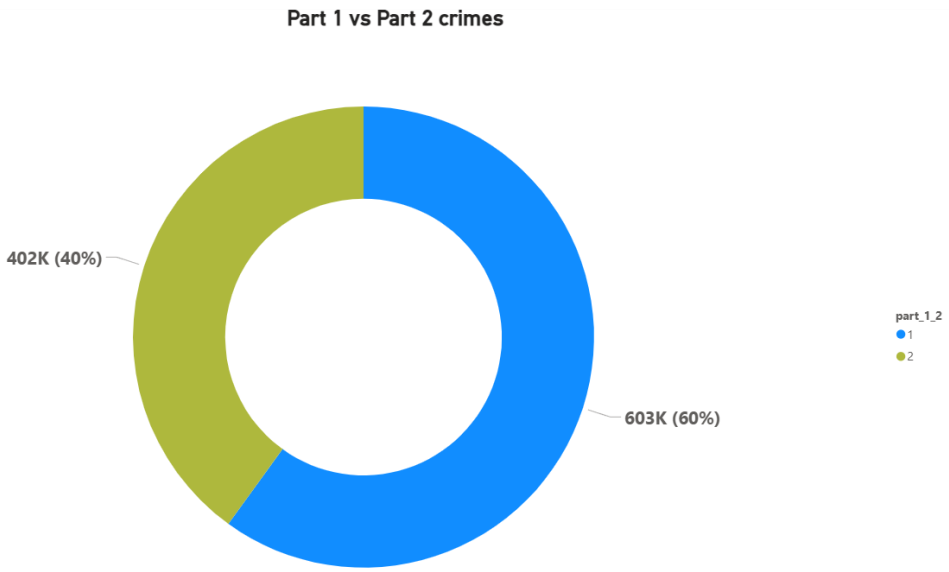
1. THEFT OF IDENTITY (35,804 victims),
2. INTIMATE PARTNER — SIMPLE ASSAULT (35,508 victims),
3. BATTERY — SIMPLE ASSAULT (35,142 victims),
4. BURGLARY FROM VEHICLE (26,549 victims),
5. THEFT PLAIN — PETTY ($950 & UNDER) (23,309 victims).



Screenshot 69.2: “Incidents by crime type for females: Top 5”

The third visualization uses `dimCrime`’s and `factCrime`’s datasets and shows us which offense classification had the most incidents between Part 1 and Part 2. The steps shown in Appendix 70 were followed to design it.

After examining our donut chart, we observe that Part 1’s incidents — serious crimes — are the 6/10 of all crime incidents (602,715 incidents), while Part 2’s incidents — less serious crimes — are the 4/10 (402,347 incidents).

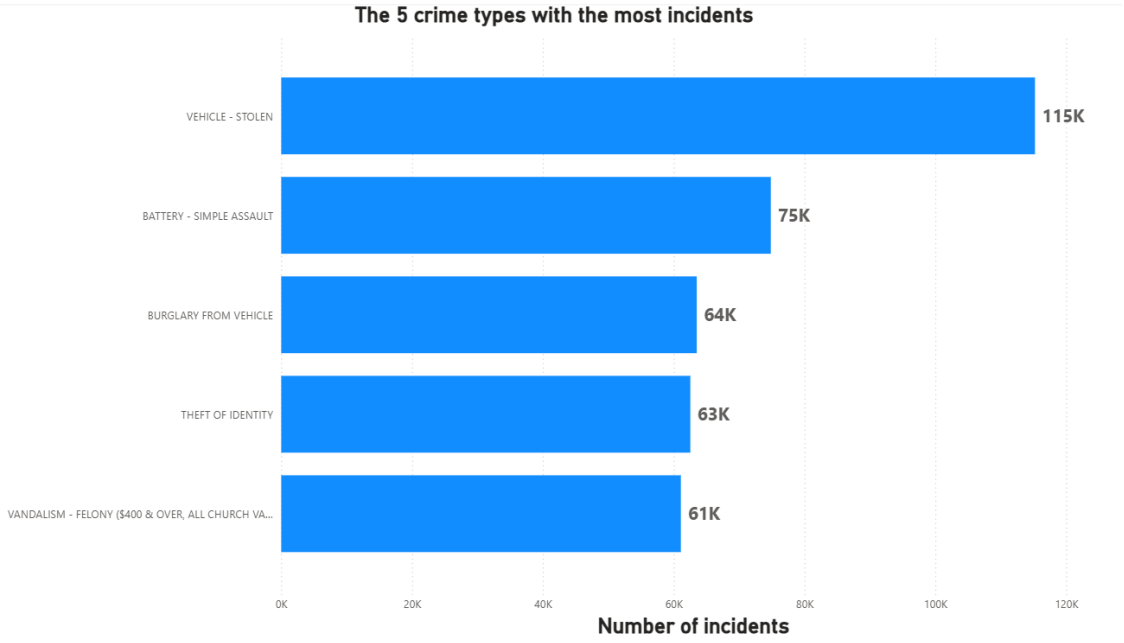


Screenshot 70: "Part 1 vs Part 2 crimes"

The fourth visualization, which uses `dimCrime`'s and `factCrime`'s datasets, depicts the five most frequent crime types. To construct it, the steps presented in Appendix 71 were executed.

After a brief examination of our bar chart's results, we see that the crime type with the most incidents is VEHICLE – STOLEN, with a total of more than 115,000 incidents. Analytically, the five crime types with the most crimes, sorted by the number of incidents in descending order, are:

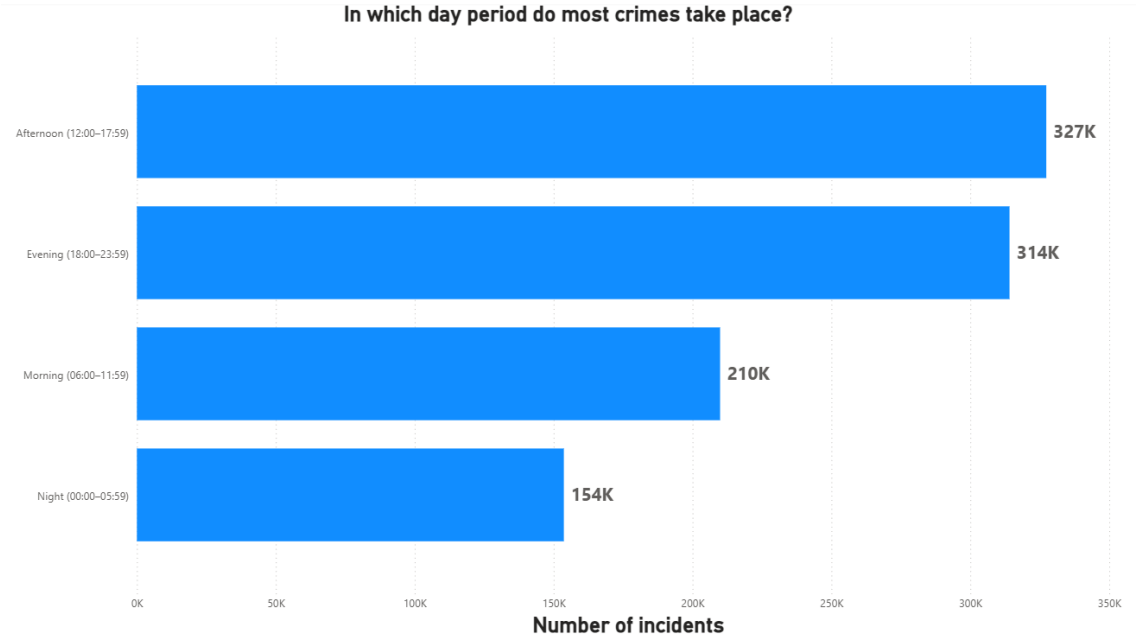
1. VEHICLE — STOLEN (115,226 incidents),
2. BATTERY — SIMPLE ASSAULT (74,839 incidents),
3. BURGLARY FROM VEHICLE (63,517 incidents),
4. THEFT OF IDENTITY (62,539 incidents),
5. VANDALISM — FELONY ($400 & OVER, ALL CHURCH VANDALISMS) (61,092 incidents).



Screenshot 71: “Incidents by crime type: Top 5”

The fifth visualization uses `factCrime`’s dataset and portrays the four-day periods we created earlier, ordered from the period with the most incidents to the one with the least. With the steps located at Appendix 72, this bar chart was designed.

It is apparent that “Afternoon (12:00–17:59)” is the period of day in which most incidents happen, having 327,341 incidents. Second comes “Evening (18:00–23:59) with 314,110 incidents while third and last place are claimed by “Morning (06:00–11:59)” and “Night (00:00–05:59)”, both of which have 209,927 and 153,684 incidents, respectively.

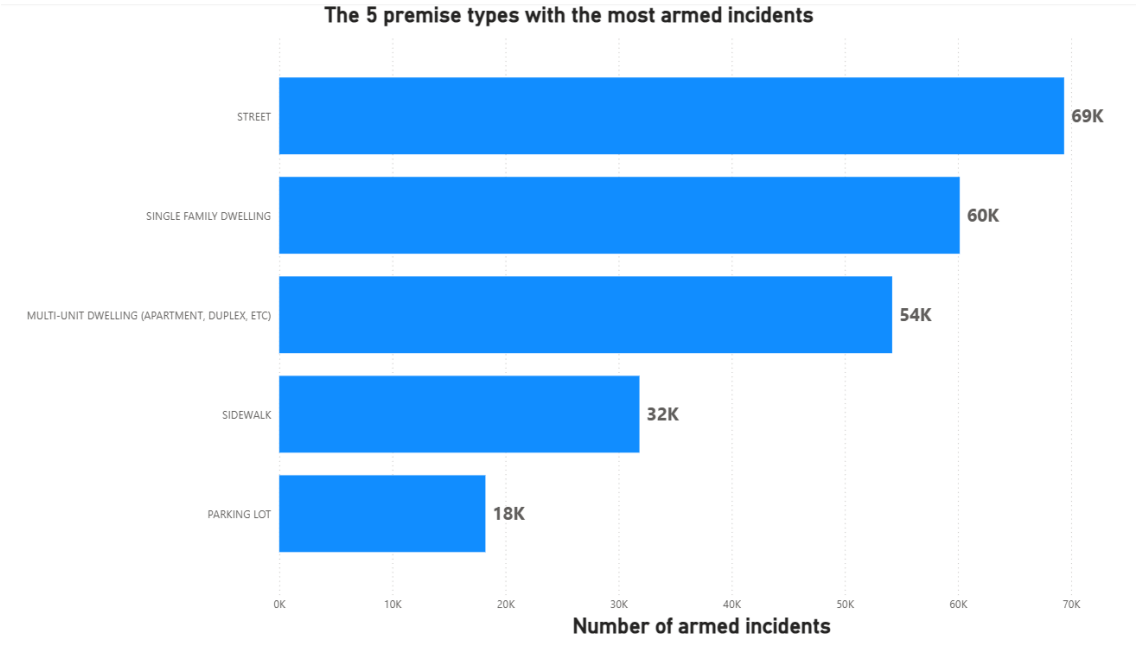


Screenshot 72: “Incidents by day period”

The sixth visualization presents the five premise types with the most armed incidents. By using sixth `dimPremise`’s and `factCrime`’s datasets, and following the instructions presented in Appendix 73, the chart was generated.

The chart uncovers that the premise type with the highest number of crimes is STREET, which accounts for over 69,000 incidents. From an analytical perspective, the five premise types with the most armed incidents, ordered by the number of armed incidents in descending order, are the following:

1. STREET (69,382 incidents),
2. SINGLE FAMILY DWELLING (60,159 incidents),
3. MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC) (54,183 incidents),
4. SIDEWALK (31,847 incidents),
5. PARKING LOT (18,228 incidents).

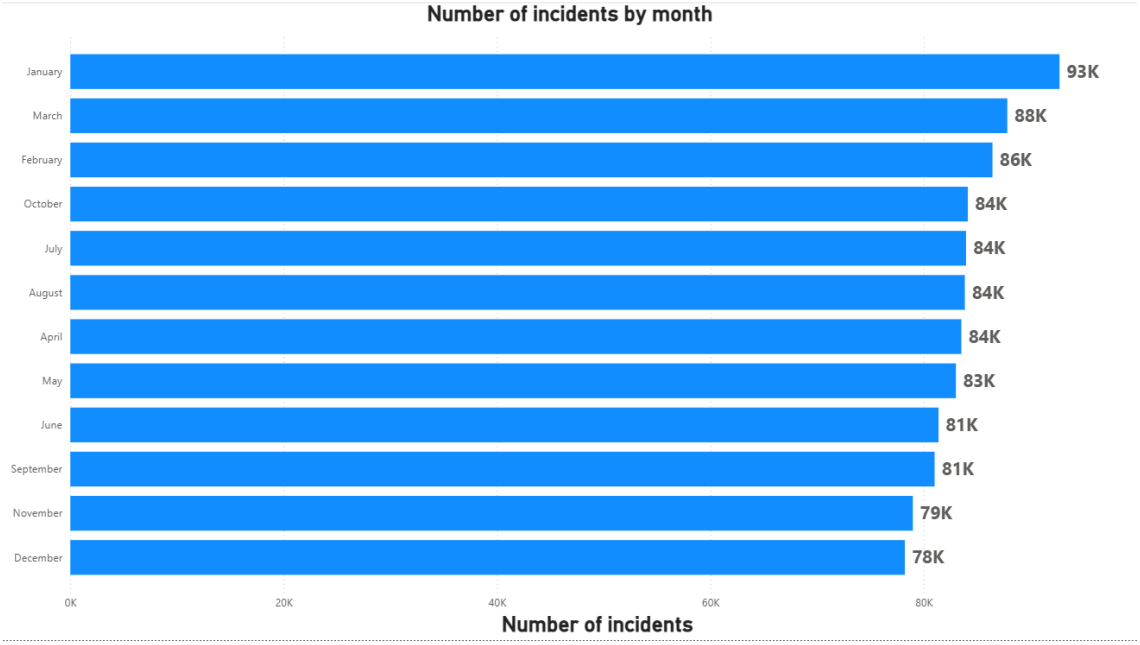


Screenshot 73: "Armed incidents by premise: Top 5"

The seventh visualization uses `factCrime`'s dataset. It uncovers which months of the year in general have the most incidents by showing us the total number of incidents for each month. The guide shown in Appendix 74 facilitated its creation.

According to the chart, January recorded the highest number of incidents, with more than 92,000 crimes. Our results in detail, sorted from the month with the highest number of incidents to the one with the lowest number of incidents, are:

1. January (92,724 incidents),
2. March (87,829 incidents),
3. February (86,440 incidents),
4. October (84,127 incidents),
5. July (83,962 incidents),
6. August (83,850 incidents),
7. April (83,518 incidents),
8. May (83,013 incidents),
9. June (81,382 incidents),
10. September (81,015 incidents),
11. November (78,976 incidents),
12. December (78,226 incidents).

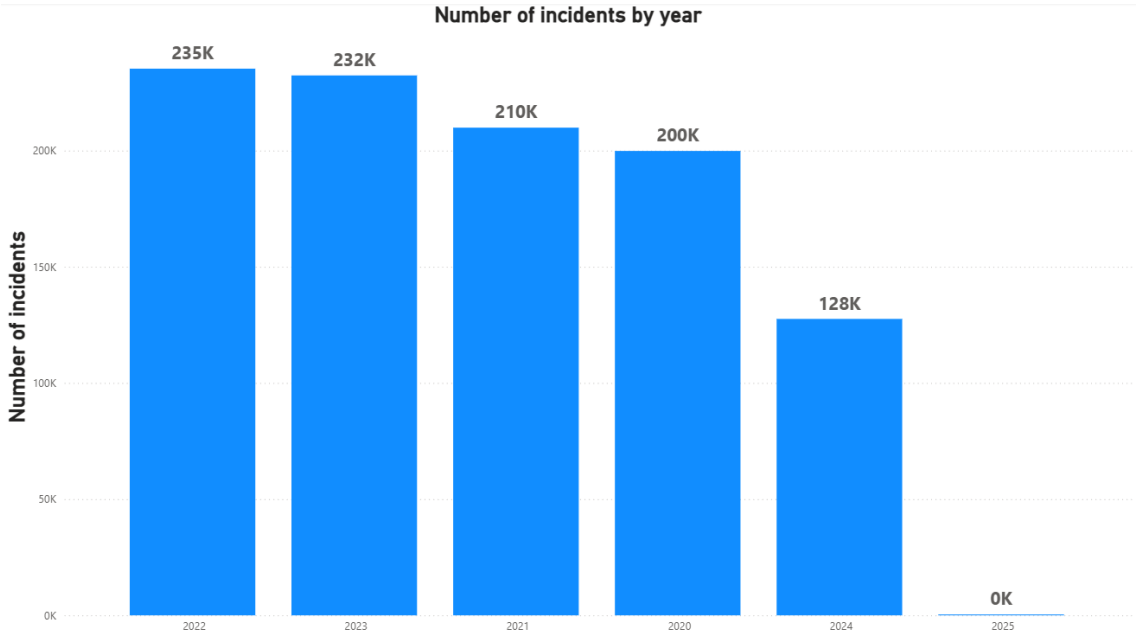


Screenshot 74: “Incidents by month”

Visualization eight is similar to the last, but shows yearly crime data instead of monthly. We built this chart by using `factCrime`’s data and the instructions shown in Appendix 75.

The chart unveils that 2022 stands out as the year with the most incidents — over 235,000 crimes. Our complete results in descending order are:

1. 2022 (235,258 incidents),
2. 2023 (232,345 incidents),
3. 2021 (209,876 incidents),
4. 2020 (199,847 incidents),
5. 2024 (127,567 incidents),
6. 2025 (169 incidents).



Screenshot 75: “Incidents by year”

The ninth visualization, which is based on `factCrime`’s dataset, portrays the average age of all victims. The steps presented in Appendix 76 were undertaken to create it.

According to our data, the average age of victims is approximately 39.4 years.

Screenshot 76: “Average victim age”

The tenth visualization, which uses `factCrime`’s dataset, presents to us the

percentage of crime incidents that includes a weapon. To design it, the steps located at Appendix 77 were executed.

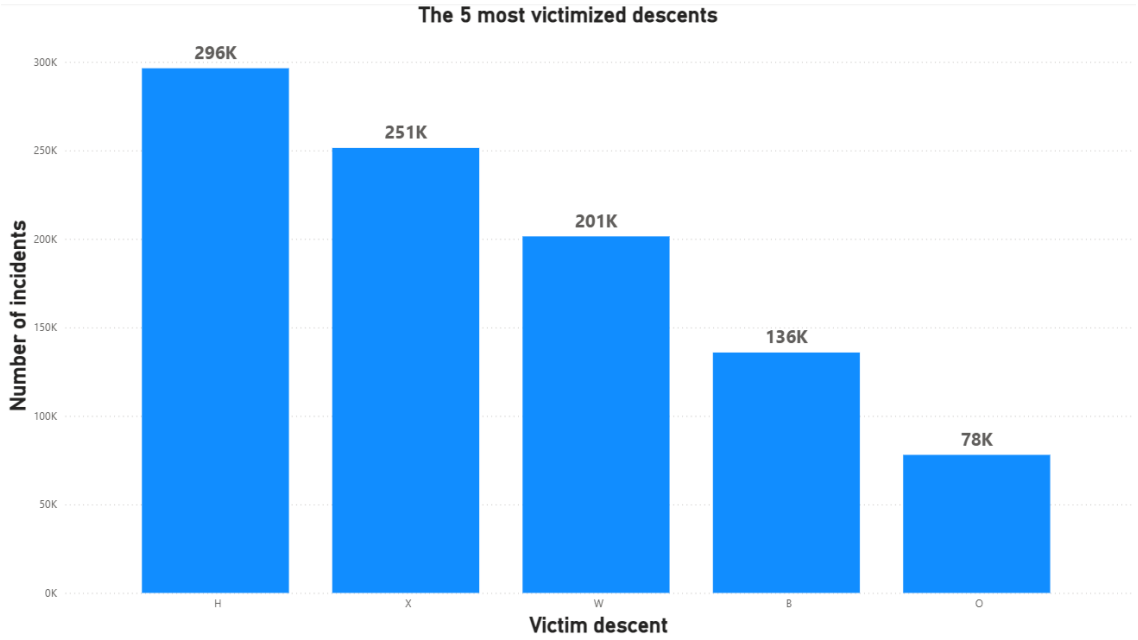
The percentage of crime incidents that involves a weapon is 32.56%.

Screenshot 77: “Percentage of armed incidents”

Our eleventh visualization uses `dimVictim`’s, and `factCrime`’s datasets. It displays the five descents with the most victims. To create it, the steps shown in Appendix 78 were carried out.

The chart shows that H (Hispanic/Latin/Mexican) is the descent with the most victims. Statistically speaking, our results in descending order are:

1. H (Hispanic/Latin/Mexican) (296,403 victims),
2. W (White) (201,441 victims),
3. B (Black) (135,817 victims),
4. O (Other) (78,005 victims),
5. A (Other Asian) (21,340 victims).

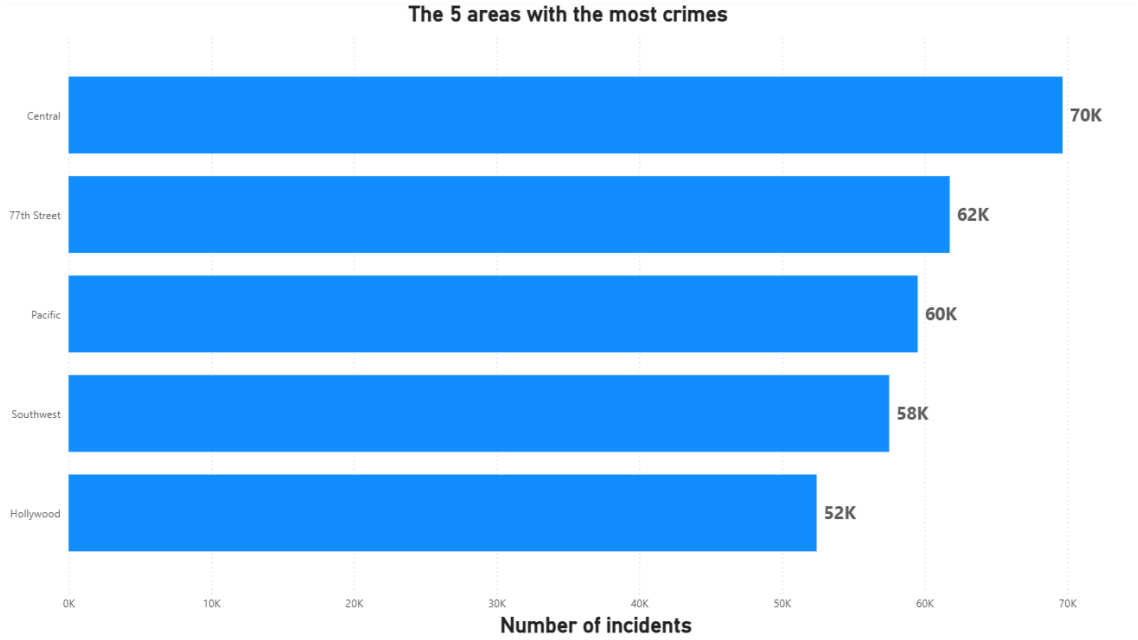


Screenshot 78: “Incidents by descent: Top 5”

The twelfth visualization illustrates the five areas with the most crimes. It uses the dataset from `dimArea`, and `factCrime`. We followed the guide located at Appendix 79 to build it.

According to our chart, Central is the area of Los Angeles with the highest number of incidents, with a total of more than 69,000 crimes. Our results in detail, sorted in descending order, are the following:

1. Central (69,670 incidents),
2. 77th Street (61,758 incidents),
3. Pacific (59,514 incidents),
4. Southwest (57,514 incidents),
5. Hollywood (52,429 incidents).

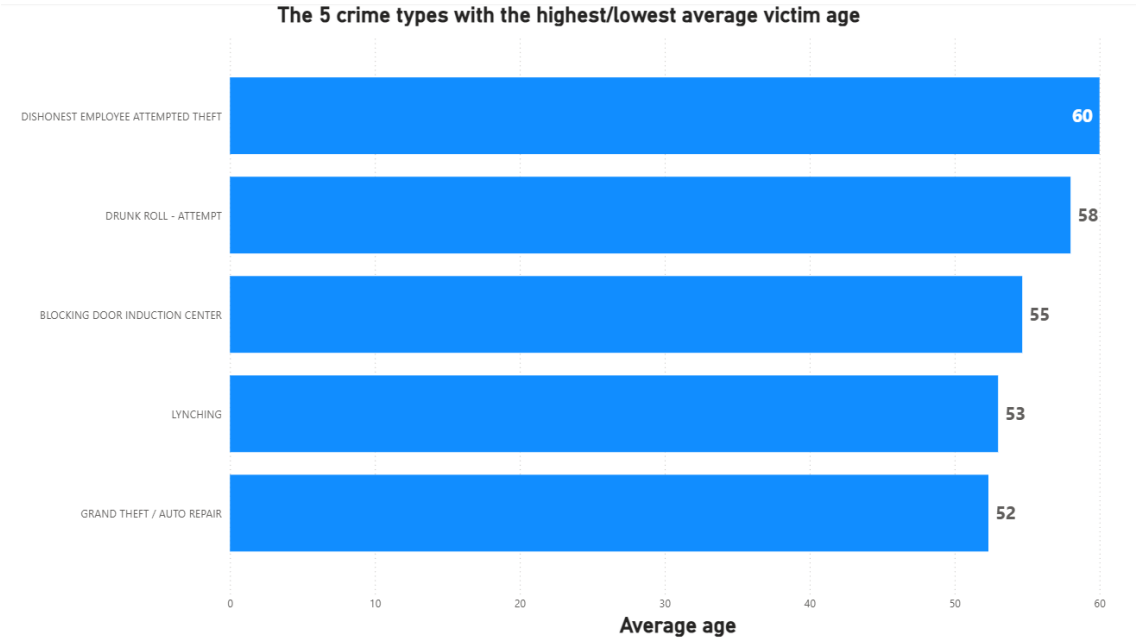


Screenshot 79: “Incidents by area: Top 5”

The thirteenth visualization that we created is based on `factCrime`’s, and `dimCrime`’s datasets. It depicts the five crime types with the lowest and the five with the highest average victim age. The steps shown in Appendix 80 were followed for the creation of this chart.

Examining the bar chart, we observe that the crime type with the highest average victim age is DISHONEST EMPLOYEE ATTEMPTED THEFT, with an age of 60 years. Based on the data, the five crime types with the highest average age, ordered by the victim age in descending order, are:

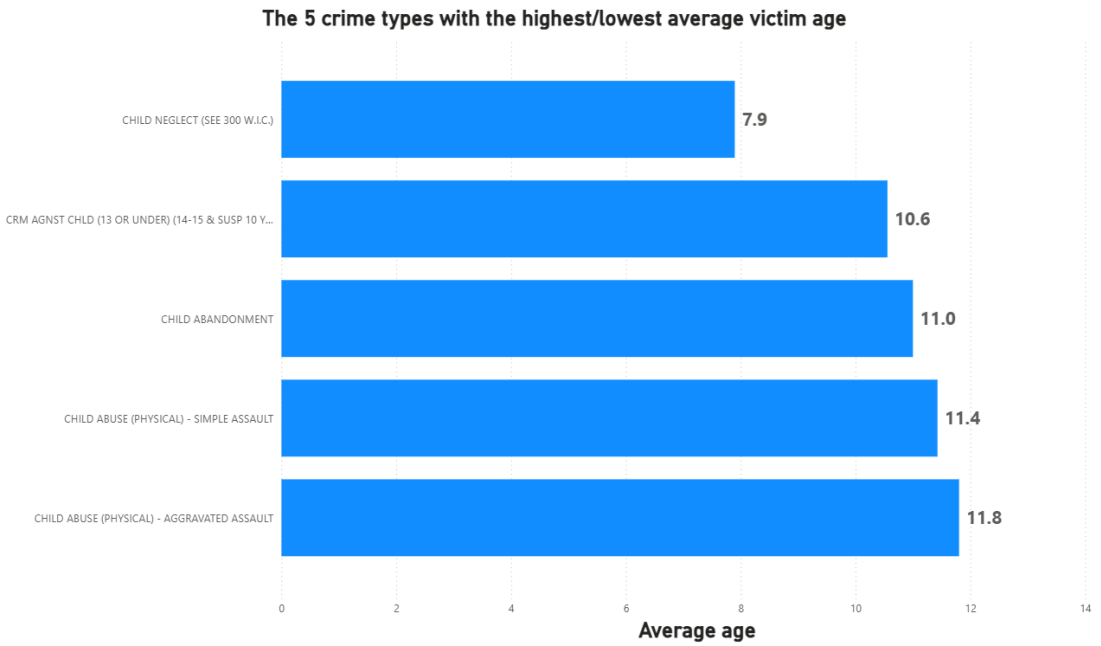
1. DISHONEST EMPLOYEE ATTEMPTED THEFT (60 years),
2. DRUNK ROLL — ATTEMPT (58 years),
3. BLOCKING DOOR INDUCTION CENTER (55 years),
4. LYNCHING (53 years),
5. GRAND THEFT/AUTO REPAIR (52 years).



Screenshot 80.1: “Average victim age by crime type: Top 5”

In contrast, the crime type with the lowest average age is CHILD NEGLECT (SEE 300 W.I.C.), with an average age of 7,9 years. Data-wise, the five crime types with the lowest average age, listed in ascending order, are:

1. CHILD NEGLECT (SEE 300 W.I.C.) (7,9 years),
2. CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER) (10,6 years),
3. CHILD ABANDONMENT (11 years),
4. CHILD ABUSE (PHYSICAL) — SIMPLE ASSAULT (11,4 years),
5. CHILD ABUSE (PHYSICAL) — AGGRAVATED ASSAULT (11,8 years).



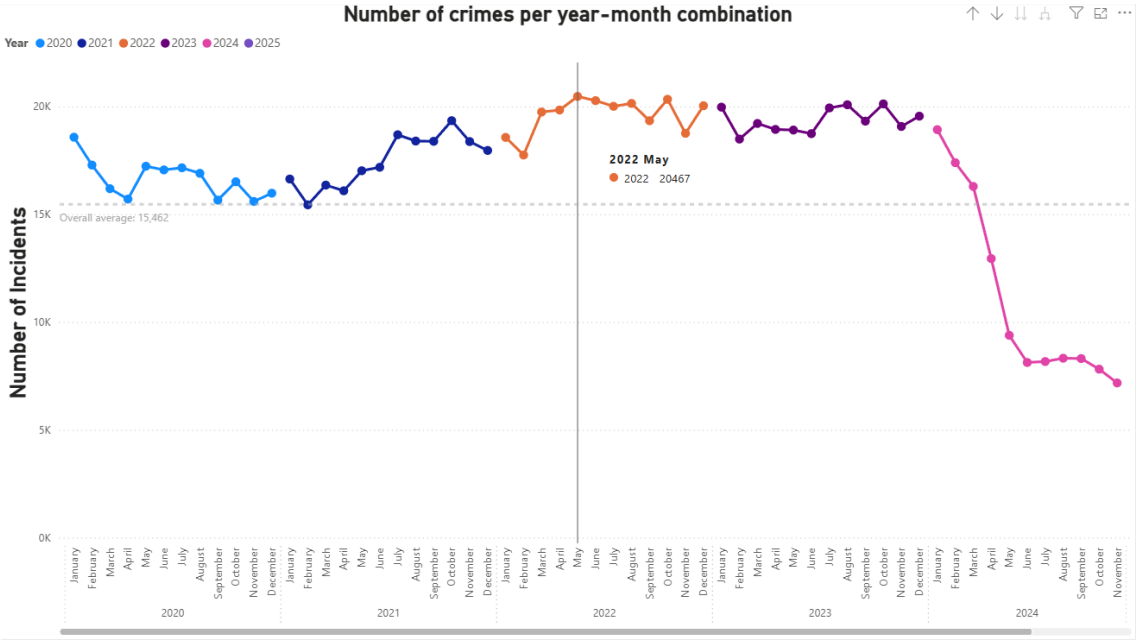
Screenshot 80.2: "Average victim age by crime type: Bottom 5"

The fourteenth visualization is based on the dataset from `factCrime`. It displays the number of crimes per year–month combination, sorted in ascending chronological order, with older combinations appearing on the left side of the chart. Additionally, each data point is compared to the overall average using an average line, highlighting deviations from the norm. To create it, the guide presented in Appendix 81 was useful.

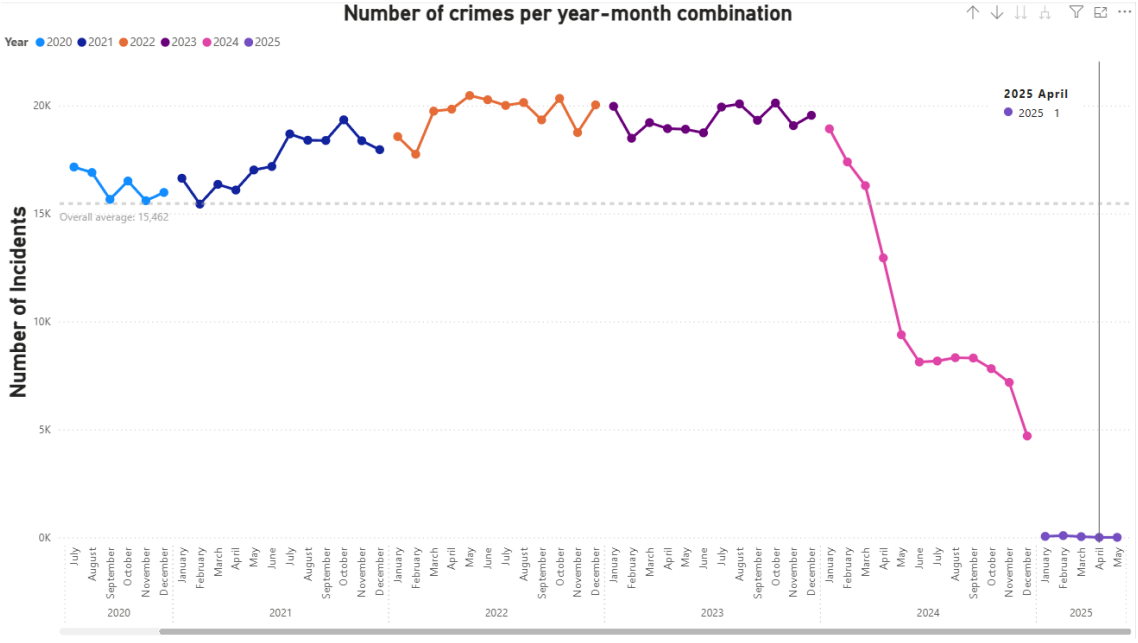
At a glance, the line chart tells us that the number of incidents has:

- steadily increased the period 2020–2022,
- remained stable throughout 2023,
- abruptly decreased during 2024,
- remained relatively stable with minor fluctuations in the first five months of 2025.

By hovering over each data point, we can analytically observe the monthly incident counts for each year compare them to the overall average of 15,462 incidents.



Screenshot 81.1: “Number of crimes per year–month”

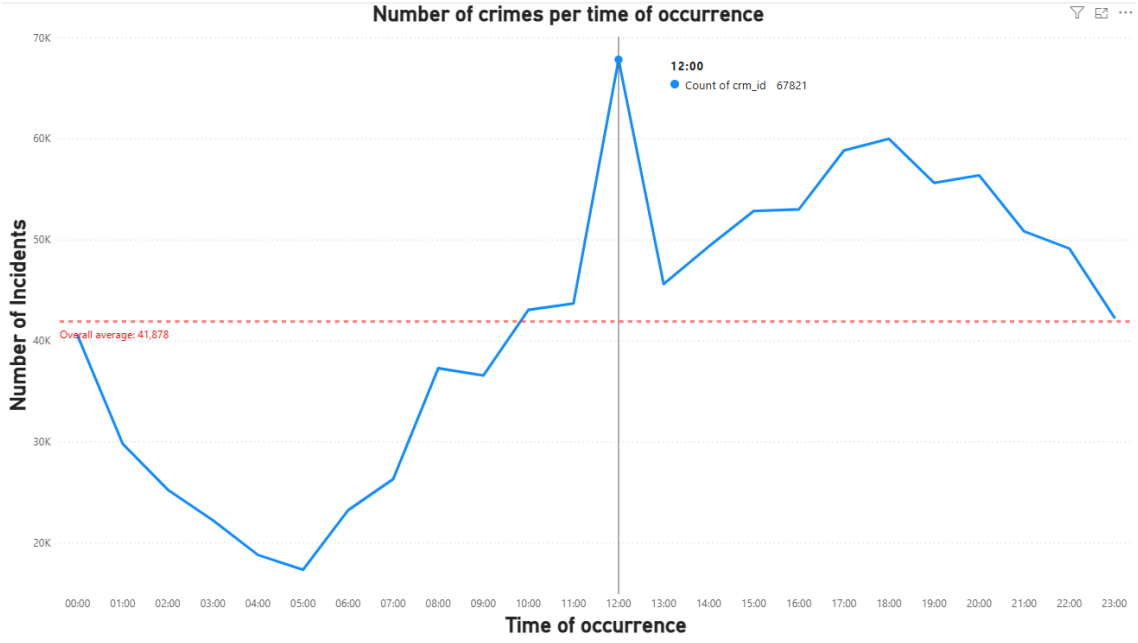


Screenshot 81.2: “Number of crimes per year–month”

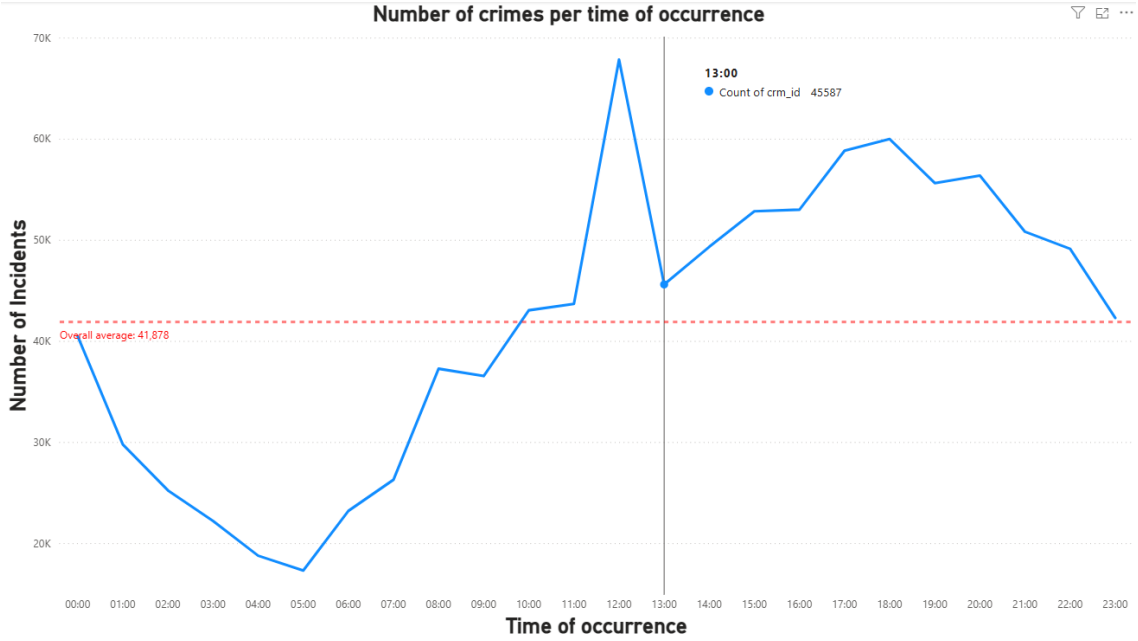
The fifteenth visual that we created portrays the number of incidents per time of occurrence and compares each data point to the overall average. We utilized the dataset from `factCrime` and the steps detailed in Appendix 82 in order to build it.

Based on the results, we observe that 12 PM marks the peak time for criminal activity (67,821 incidents) compared to the overall average (41,878 incidents)

with incidents tending to be more frequent after midday — especially after 1 PM.



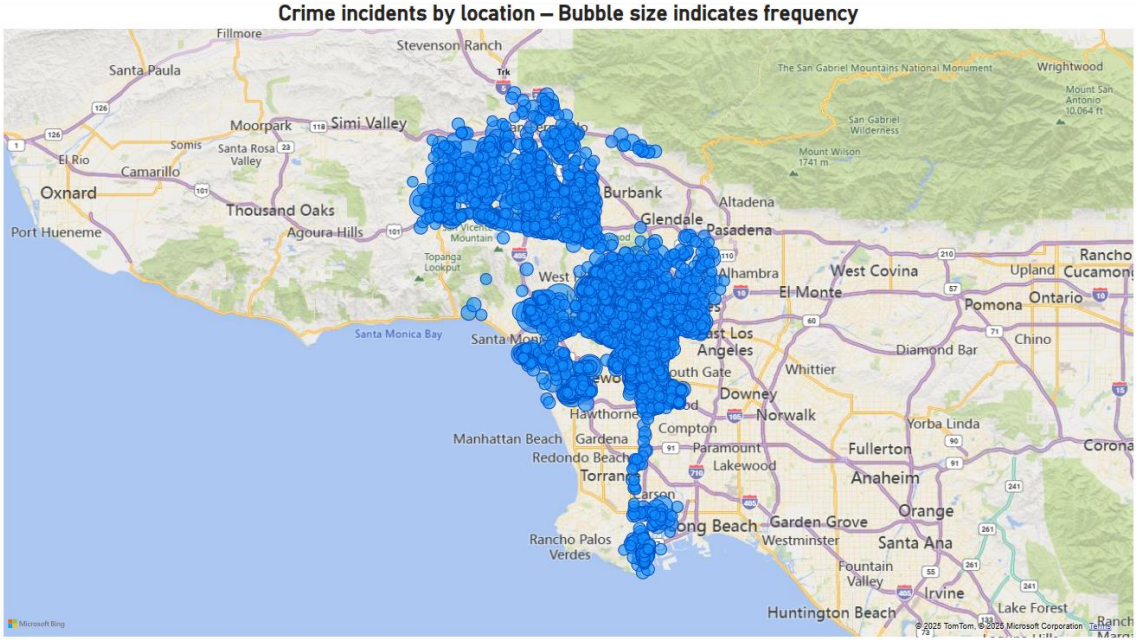
Screenshot 82.1: “Number of crimes per occurrence time”



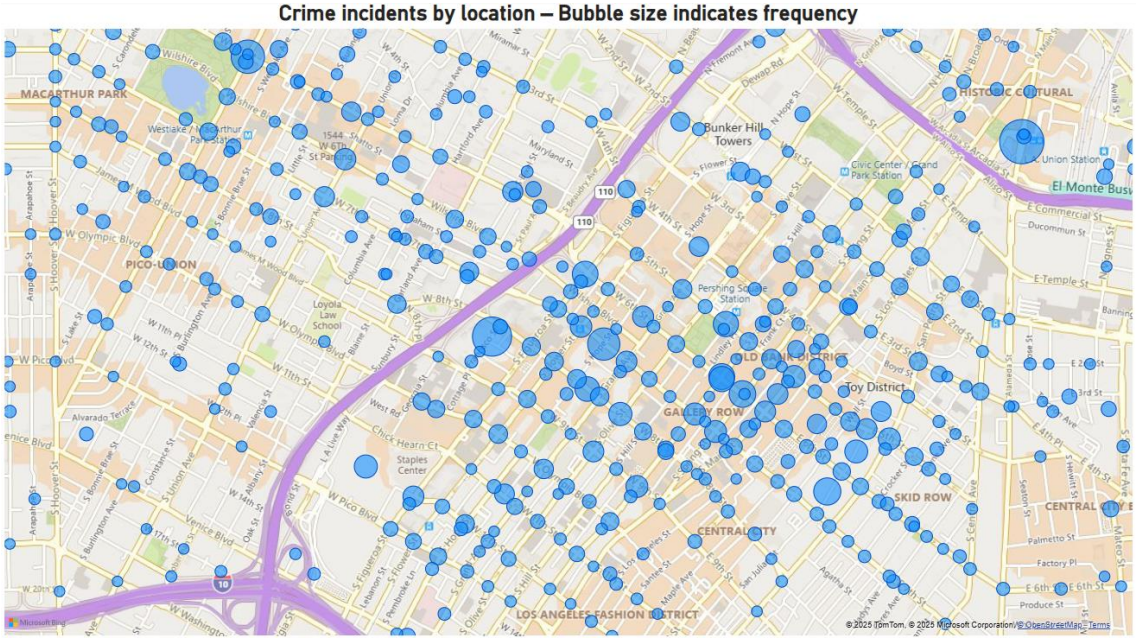
Screenshot 82.2: “Number of crimes per occurrence time”

The sixteenth visualization, based on the datasets from `dimArea`, and `factCrime`, displays on a map all exact locations — unique latitude and longitude combinations — where crimes have occurred. Each location is represented by a bubble, with its size proportional to the number of incidents

recorded at the spot. We generated it using the steps outlined in Appendix 83.



Screenshot 83.1: "All crime locations"

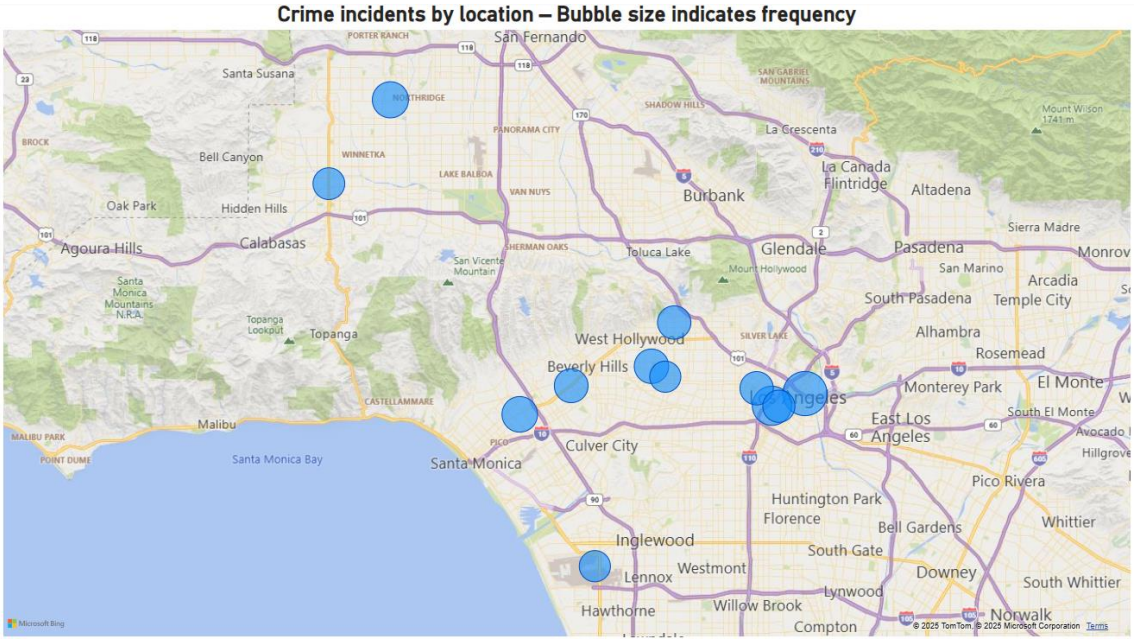


Screenshot 83.2: "Crime locations in zoomed view"

The chart unveils that the combination with lat: 34.0561, lon: -118.2375 is the map location with the highest number of incidents, amounting to nearly 2,400 incidents. The twelve combinations with more than 1000 incidents, ordered by the number of incidents in descending order, are:

1. lat: 34.0561, lon: -118.2375 (2,397 incidents),

2. lat: 34.0483, lon: -118.2631 (1,826 incidents),
3. lat: 34.244, lon: -118.5583 (1,521 incidents),
4. lat: 34.0428, lon: -118.4582 (1,469 incidents),
5. lat: 34.0736, lon: -118.3563 (1,363 incidents),
6. lat: 34.0611, lon: -118.4184 (1,307 incidents),
7. lat: 34.0595, lon: -118.2749 (1,267 incidents),
8. lat: 34.1016, lon: -118.3387 (1,267 incidents),
9. lat: 34.048, lon: -118.2577 (1,166 incidents),
10. lat: 34.1904, lon: -118.6059 (1,131 incidents),
11. lat: 33.9455, lon: -118.4001 (1,085 incidents),
12. lat: 34.0669, lon: -118.3456 (1,080 incidents).



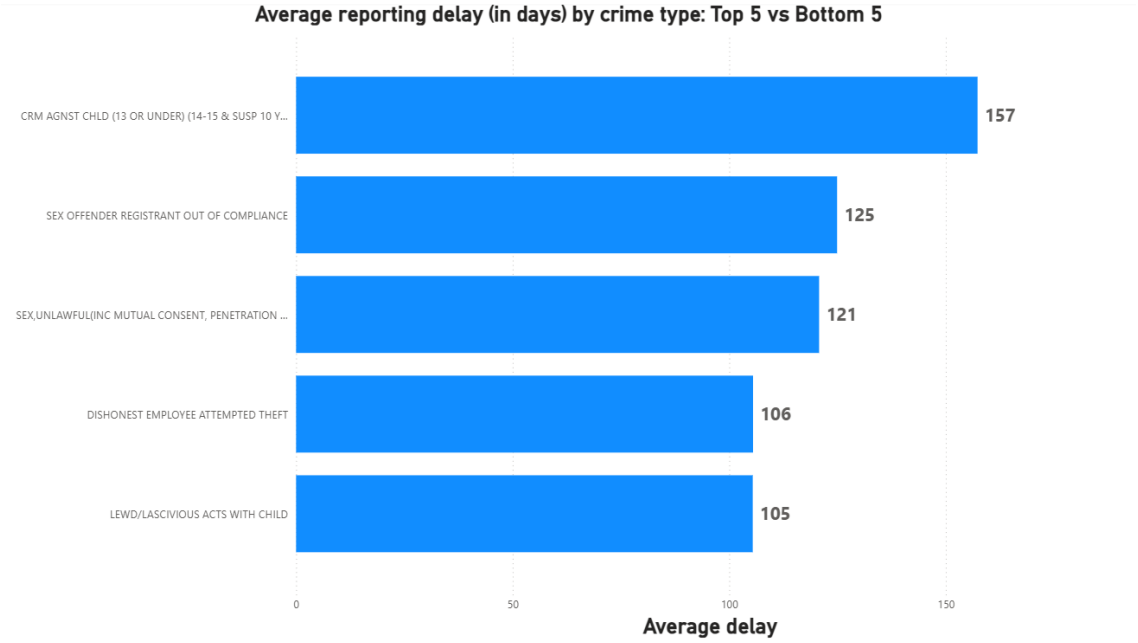
Screenshot 83.3: "Crime locations with > 1000 incidents"

The seventeenth visualization uses the datasets from `dimArea`, and `factCrime`. It highlights the five crime types with the highest and all the crime types with the lowest average number of days between the occurrence of an incident and its reporting. We designed it with the steps located at Appendix 84.

From the bar chart we created, we observe that the crime type with the highest average reporting delay is CRM AGNST CHLD (13 OR UNDER) (14–15 & SUSP 10 YRS OLDER), with a delay of 157 days. From a data-driven standpoint, the five crime types with the highest average delay, sorted by the

average delay in descending order, are:

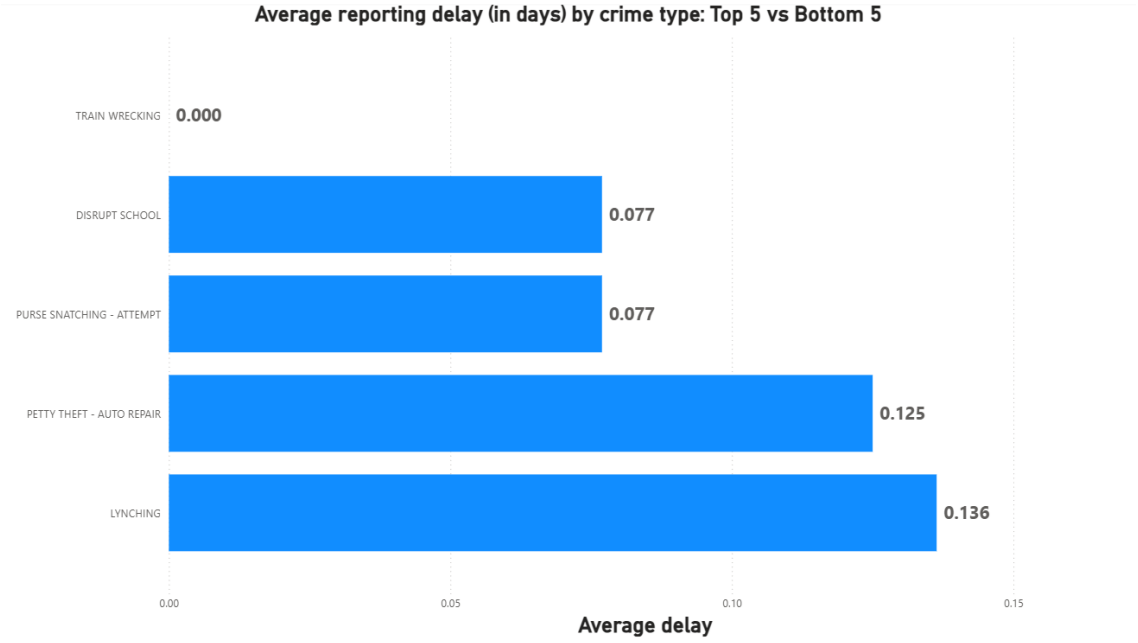
1. CRM AGNST CHLD (13 OR UNDER) (14–15 & SUSP 10 YRS OLDER) (157 days),
2. SEX OFFENDER REGISTRANT OUT OF COMPLIANCE (125 days),
3. SEX, UNLAWFUL (INC MUTUAL CONSENT, PENETRATION W/ FRGN OBJ) (121 days),
4. DISHONEST EMPLOYEE ATTEMPTED THEFT (106 days),
5. LEWD/LASCIVIOUS ACTS WITH CHILD (105 days).



Screenshot 84.1: "Average reporting delay by crime type: Top 5"

In contrast, the crime type with the lowest average delay is TRAIN WRECKING. Analytically, the five crime types with the lowest average delay are:

1. TRAIN WRECKING (0 days),
2. DISRUPT SCHOOL (0.077 days),
3. PURSE SNATCHING — ATTEMPT (0.077 days),
4. PETTY THEFT — AUTO REPAIR (0.125 days),
5. LYNCHING (0.136 days).

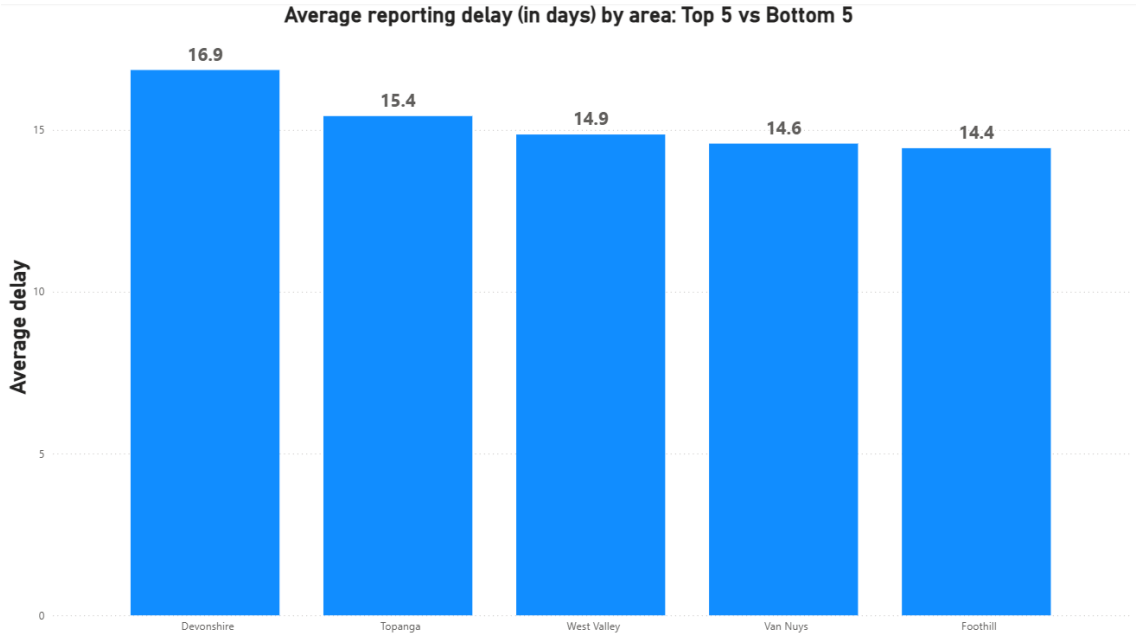


Screenshot 84.2: "Average reporting delay by crime type: Bottom 5"

The eighteenth visualization, which is based on the datasets from `factCrime` and `dimArea`, presents the five areas with the highest and the five areas with the lowest average number of days between the occurrence of an incident and its reporting. We created it using the steps listed in Appendix 85.

The area with the lowest average reporting delay is Central, with a delay of 8.7 days. According to the plot, the five areas with the lowest average delay, sorted by the average delay in ascending order, are:

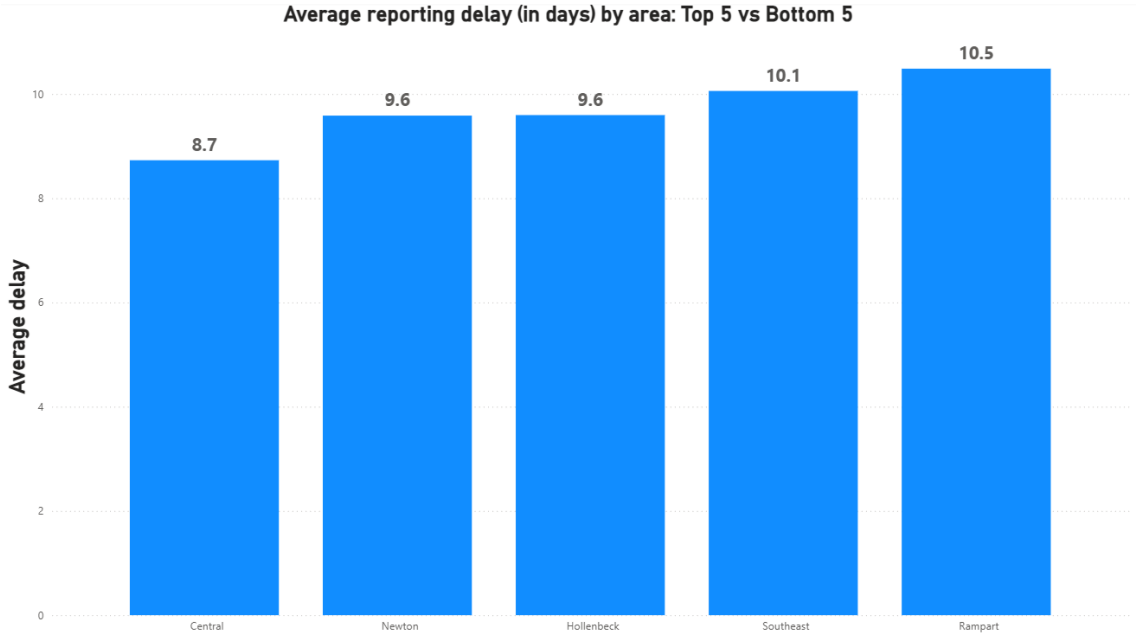
1. Central (8.7 days),
2. Newton (9.6 days),
3. Hollenbeck (9.6 days),
4. Southeast (10.1 days),
5. Rampart (10.5 days).



Screenshot 85.1: “Average reporting delay by area: Top 5”

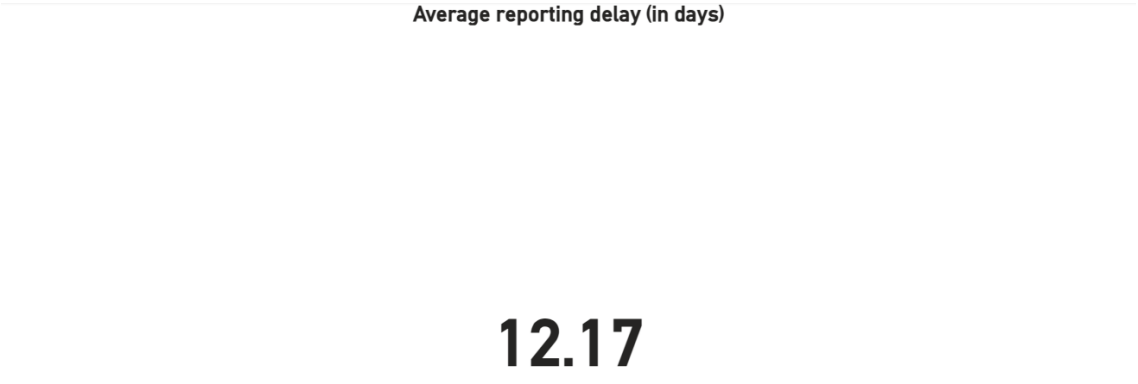
On the contrary, the area with the highest average reporting delay is Devonshire, with a delay of 16.9 days. In detail, the five areas with the highest average delay, sorted by the average delay in descending order, are:

1. Devonshire (16.9 days),
2. Topanga (15.4 days),
3. West Valley (14.9 days),
4. Van Nuys (14.6 days),
5. Foothill (14.4 days).



[Screenshot 85.2: “Average reporting delay by area: Bottom 5”]

Our nineteenth visualization utilizes the dataset from `factCrime` and reveals the average number of days between the occurrence of an incident and its reporting. We created it by performing the steps located at Appendix 86. According to our data, the average reporting delay is approximately 12.2 days.



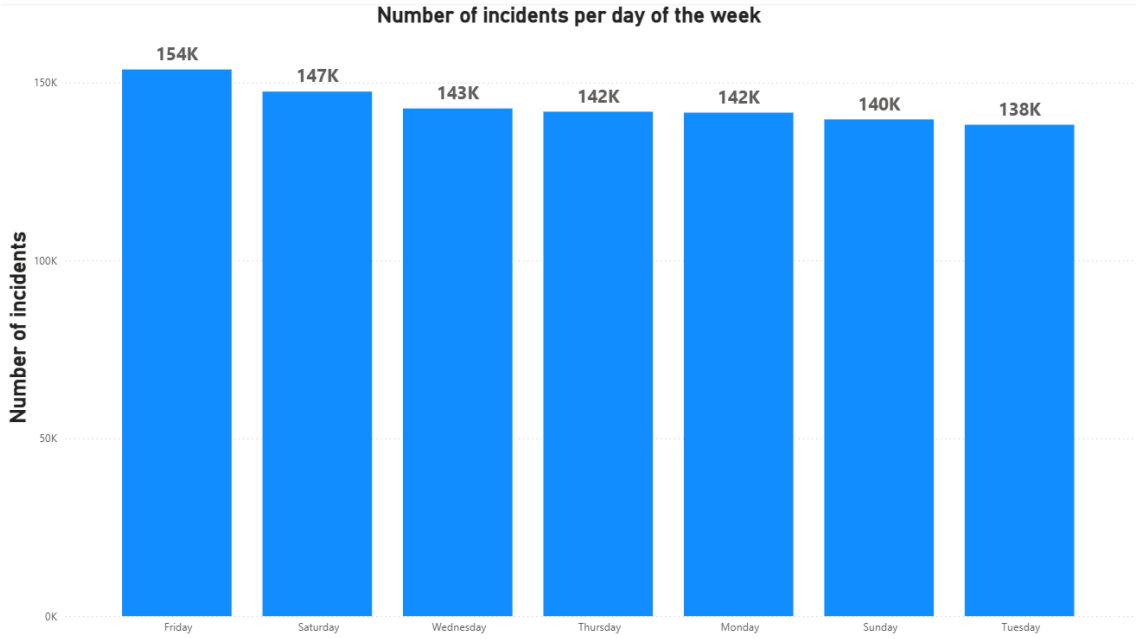
[Screenshot 86: “Average reporting delay”]

The twentieth visualization uses the dataset from `factCrime`. It portrays the

number of incidents per day of the week. The process of creating it involved the steps shown in Appendix 87.

The chart shows that Friday is the day in which most crimes occur. Our results in detail, sorted in descending order of incidents, are as follows:

1. Friday (153,689 incidents),
2. Saturday (147,468 incidents),
3. Wednesday (142,725 incidents),
4. Thursday (141,826 incidents),
5. Monday (141,550 incidents),
6. Sunday (139,654 incidents),
7. Tuesday (138,150 incidents).

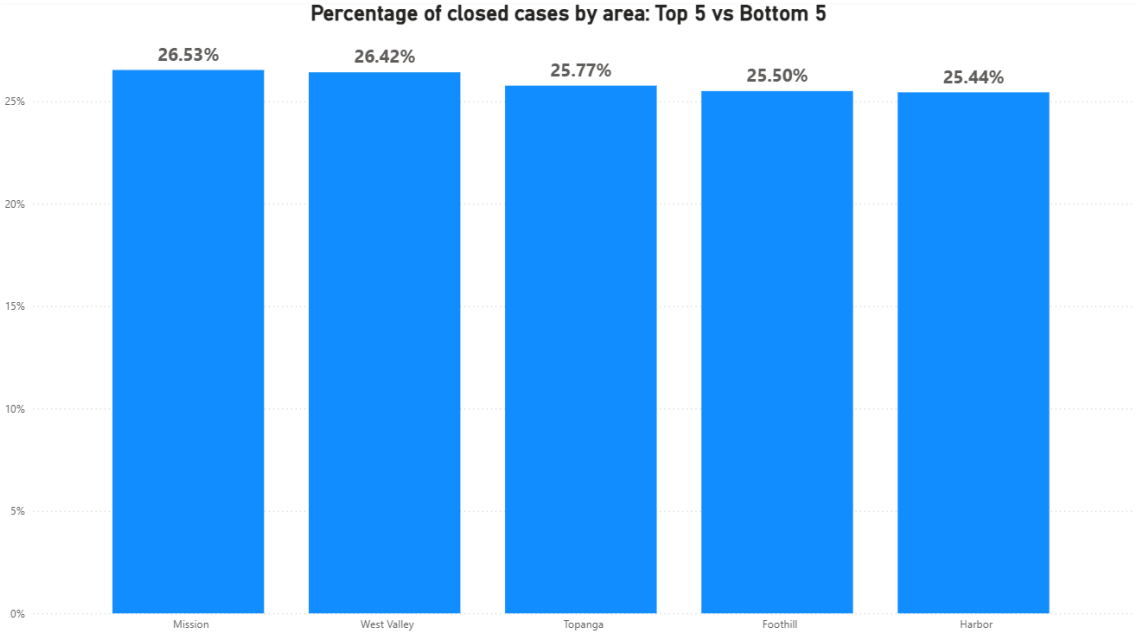


Screenshot 87: “Incidents by day of the week”

The twenty-first visualization uses the datasets from dimArea, and factCrime. It reveals the five areas with the highest and the five areas with the lowest percentage of resolved cases. The guide provided in Appendix 88 assisted us in its creation.

Our data shows that the area with the highest percentage of closed cases is Mission, with 26.53%. In detail, the five areas with the highest percentage, sorted by the percentage in descending order, are:

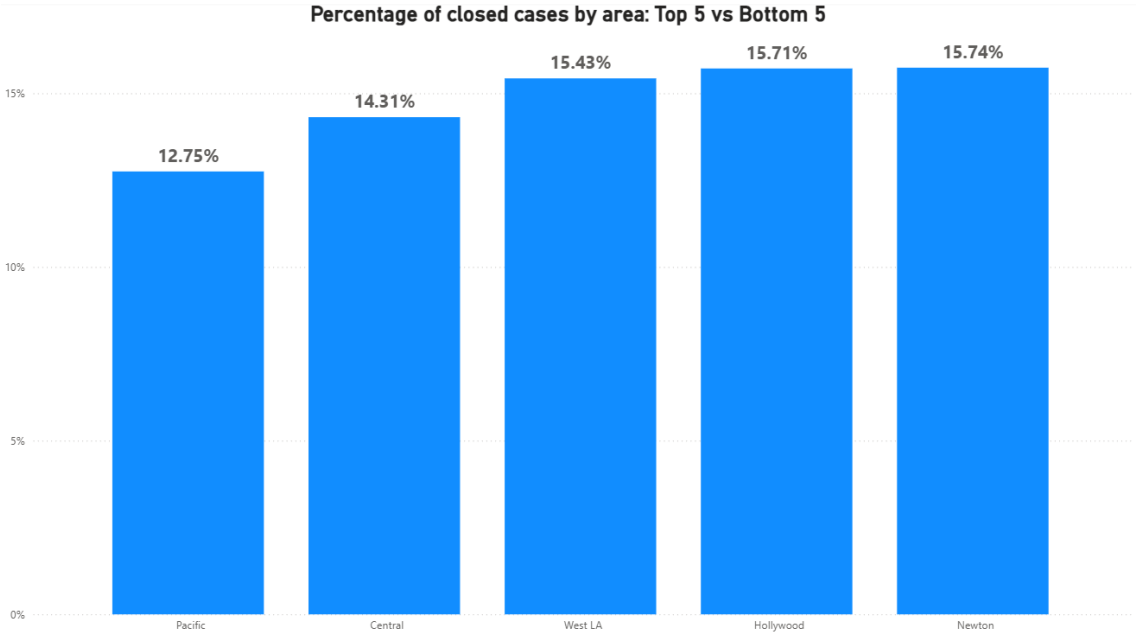
1. Mission (26.53%),
2. West Valley (26.42%),
3. Topanga (25.77%),
4. Foothill (25.5%),
5. Harbor (25.44%).



Screenshot 88.1: “Closed cases by area: Top 5”

On the contrary, the area with the lowest percentage of closed cases is Pacific, with 12.75%. In detail, the five areas with the lowest percentage, sorted by the percentage in ascending order, are:

1. Pacific (12.75%),
2. Central (14.31%),
3. West LA (15.43%),
4. Hollywood (15.71%),
5. Newton (15.74%).

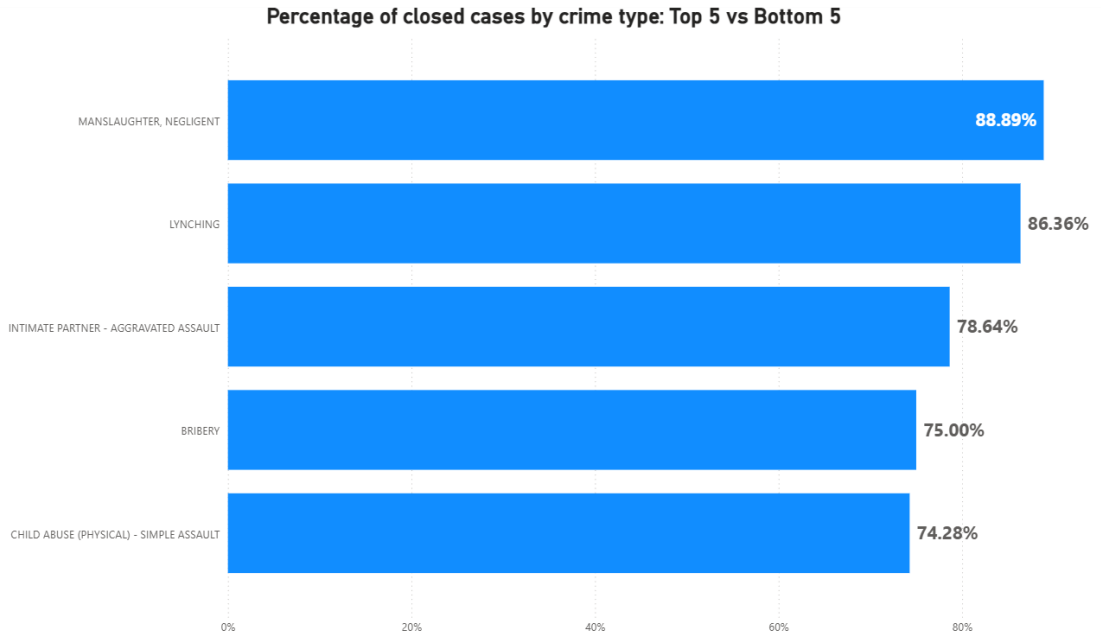


Screenshot 88.2: “Closed cases by area: Bottom 5”

The twenty-second visual, which is based on the datasets from `factCrime` and `dimCrime`, presents the five crime types with the highest and the five crime types with the lowest percentage of closed/resolved cases. We created it using the steps listed in Appendix 89.

Examining the bar chart we created, we observe that the crime type with the highest percentage of closed cases is MANSLAUGHTER, NEGLIGENT, with 88.89%. Based on the data, the five crime types with the highest percentage, ordered by the percentage in descending order, are:

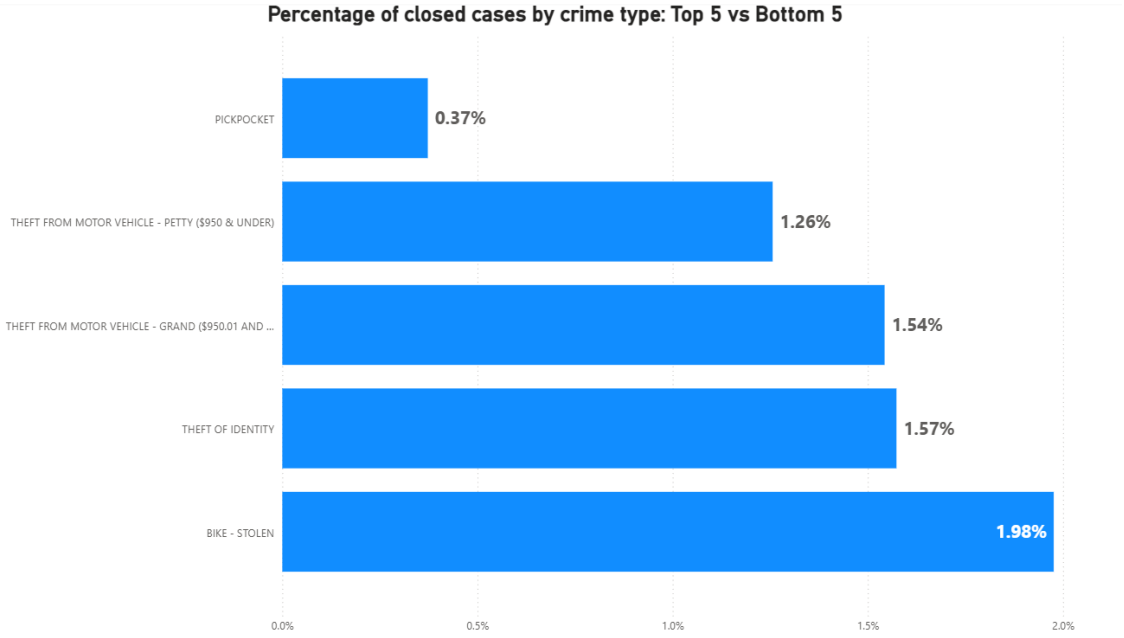
1. MANSLAUGHTER, NEGLIGENT (88.89%),
2. LYNCHING (86.36%),
3. INTIMATE PARTNER — AGGRAVATED ASSAULT (78.64%),
4. BRIBERY (75%),
5. CHILD ABUSE (PHYSICAL) — SIMPLE ASSAULT (74.28%).



Screenshot 89.1: "Closed cases by crime type: Top 5"

In contrast, the crime type with the lowest percentage of closed cases is PICKPOCKET, with 0.37%. Data-wise, the five crime types with the lowest percentage, listed in ascending order, are:

1. PICKPOCKET (0.37%),
2. THEFT FROM MOTOR VEHICLE — PETTY ($950 & UNDER) (1.26%),
3. THEFT FROM MOTOR VEHICLE — GRAND ($950.01 AND OVER) (1.54%),
4. THEFT OF IDENTITY (1.57%),
5. BIKE - STOLEN (1.98%).



Screenshot 89.2: "Closed cases by crime type: Bottom 5"

The twenty-third visualization, which uses `factCrime`'s dataset, presents to us the percentage of closed cases. To design it, the steps located at Appendix 90 were executed.

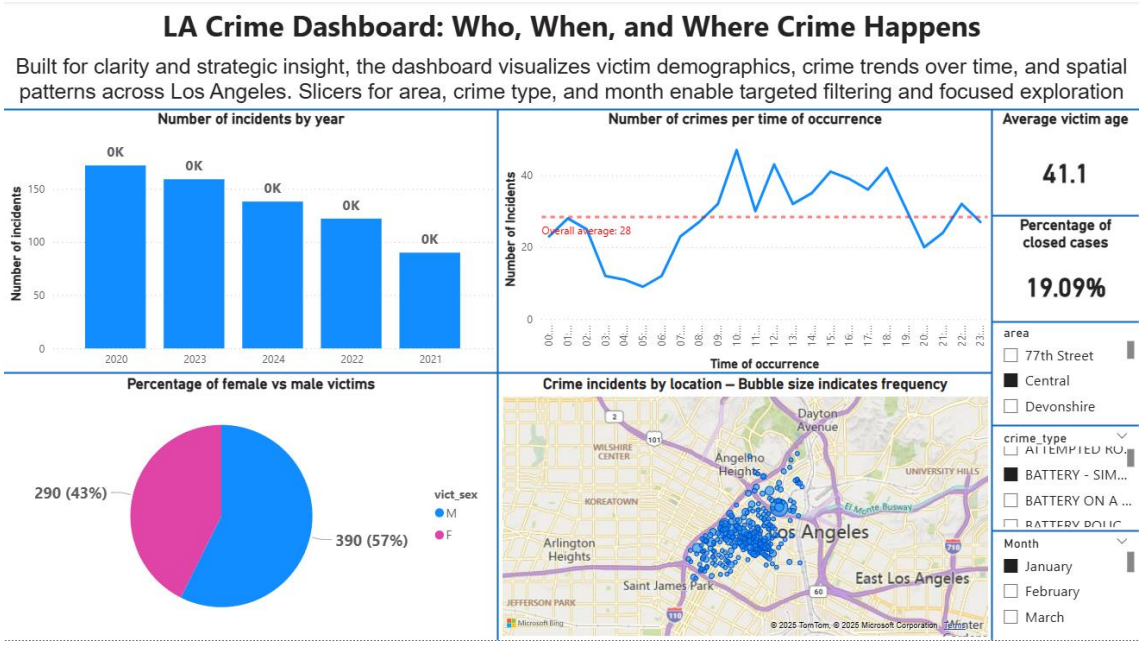
According to our chart, the percentage of closed cases is 20.09%.

Screenshot 90: "Closed cases"

4.2. Dashboard and Interactive Analysis:

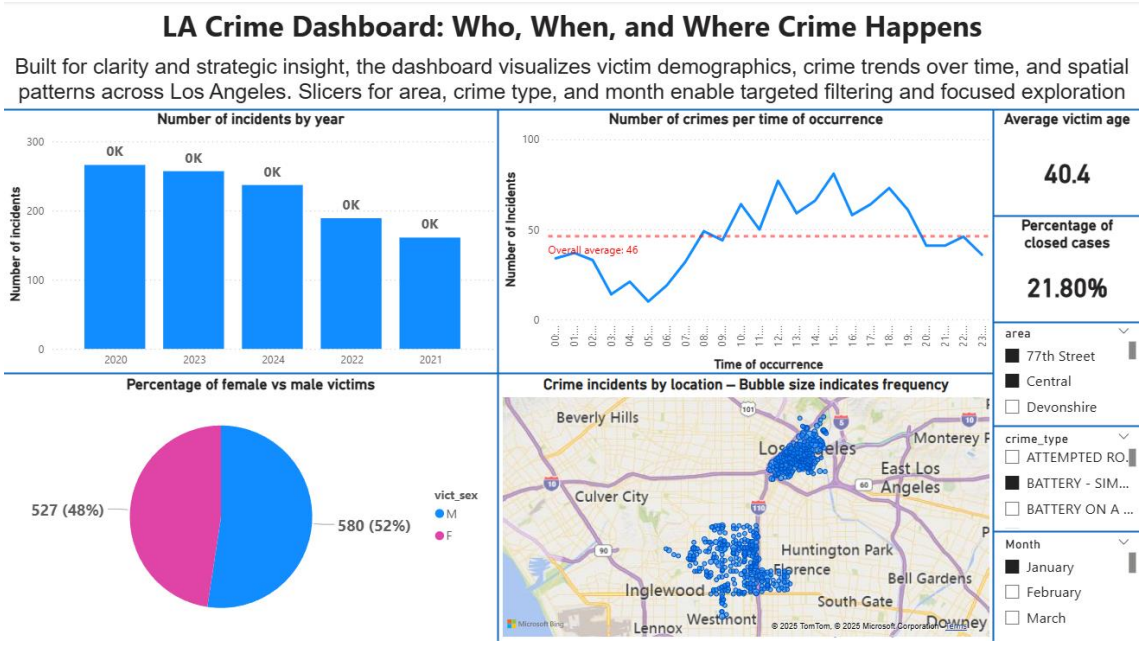
After all visualizations had been developed, we proceeded with the creation of an interactive dashboard. A dashboard is an interactive visual interface that integrates multiple visualizations and filters, allowing users to explore data dynamically and monitor key indicators in real time. This dashboard incorporates six of the previously created visualizations (the first, eighth, ninth, fifteenth, sixteenth, and twenty-third) and three slicer filters (areas, crime types, and months). These elements are combined to facilitate deeper exploration of the data and reveal key insights. The creation process is illustrated in Appendix 91.

After selecting Central as the area, BATTERY — SIMPLE ASSAULT as the crime type, January as the month, we can observe that the rest of our visuals are automatically filtered accordingly. For example, in this case, the number of incidents peaks during the year 2020 (172 incidents) and at 10 AM (47 incidents), while the percentage of closed cases reaches 19.09%. Moreover, the percentage of male victims is 57% (390 individuals) and the percentage of female victims is 43% (290 individuals), with the average victim age being 41.1 years. Finally, the map now displays only the locations corresponding to the Central area.



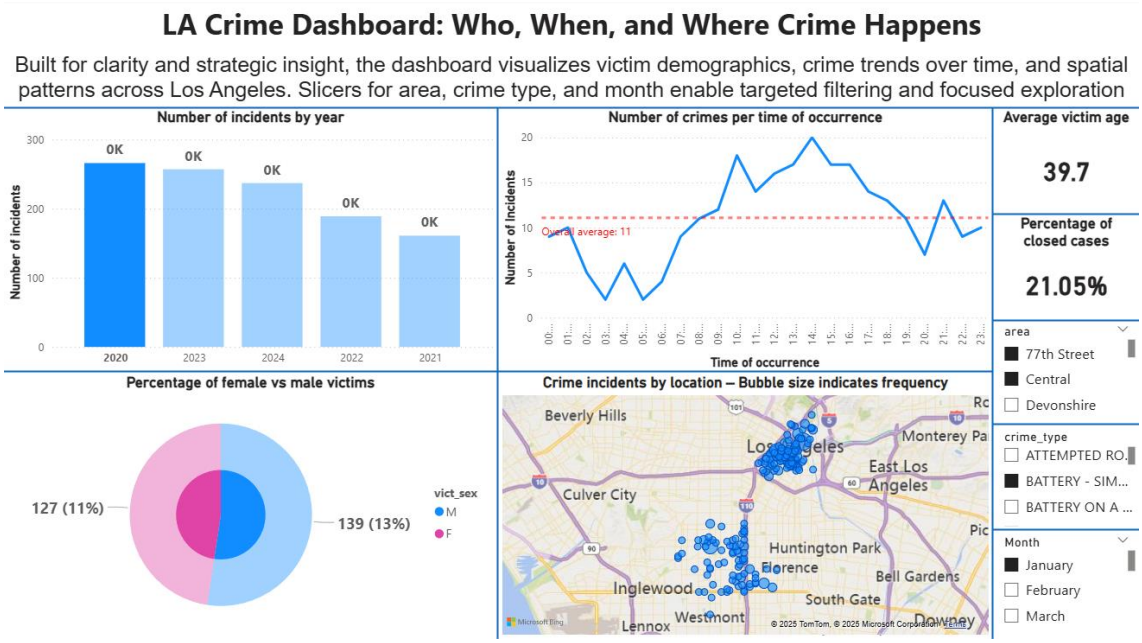
Screenshot 91.1: "Single selection filters applied"

Note that each slicer can accept multiple selections at once — for instance, two areas or two crime types — allowing the visuals to update and display combined results for the selected categories. After selecting 77th Street along with Central as the area, while keeping the crime type and month slicers set to the same options as before, we can see the number of incidents peaks during the year 2020 (266 incidents) and at 3 PM (81 incidents), while the percentage of closed cases is 21.8%. Moreover, the percentage of male victims is 52% (580 individuals) and the percentage of female victims is 48% (527 individuals), with an average victim age of 40.4 years, while the map now visualizes the incidents occurring across both selected areas.



Screenshot 91.2: "Multiple selections per slicer"

If we select a category from any of the four main charts — for instance, the column for the year 2020 in the eighth visualization – we observe that the rest of the charts are filtered accordingly to display results specific to 2020. In this case, the number of incidents peaks at 2 PM (20 incidents), while the percentage of closed cases is 21.05%. Moreover, there were 139 male victims and 127 female victims, with the average victim age being 39.7 years. Additionally, multiple categories of one or more visuals can be selected simultaneously — for instance, combining a year, a gender, and a time of occurrence — to display results filtered by all these criteria at once.



Screenshot 91.3: “Cross-filtering between visuals”

5. Conclusions and Suggestions

5.1. Conclusions:

As we reach the conclusion of our project, it is essential to summarize the key findings of our research.

- **General crime insights:**
  1. The crime types with the most incidents were stolen vehicle, simple battery, vehicle burglary, identity theft, and felony vandalism.
  2. Serious crimes accounted for 6 out of 10 total crimes while less serious accounted for the remaining 4 out of 10.
  3. The crime types with the highest percentage of closed cases were negligent manslaughter, lynching, intimate partner-related simple assault, bribery, and simple physical child abuse. Conversely, the five crime types with the lowest percentage of closed cases were pickpocket, petty theft from motor vehicle, grand theft from motor vehicle, identity theft, and bicycle theft.
  4. The percentage of closed cases reached 20.09%.
- **Victim-related insights:**
  1. Men were victimized more often than women. Specifically, 53% of recorded victims were males while the rest were females.
  2. Women were predominantly victimized in crime categories such as identity theft, intimate partner-related simple assault, simple battery, vehicle burglary, and petty theft (under $950). Men, on the other hand, were more frequently victimized in offenses including simple battery, aggravated assault with a deadly weapon, vehicle burglary, burglary, and felony vandalism.
  3. The most victimized descents were Hispanic (Latin/Mexican), White, Black, Other, and Other Asian.
  4. The mean age of victims was 39.4 years.
  5. The five crime types with the highest average age of victims were attempted theft by a dishonest employee, drunk roll attempt, blocking door at an induction center, lynching, and grand theft/auto repair. Conversely, the five crime types with the lowest average age were

child neglect, crime against child, child abandonment, simple physical child abuse, and aggravated physical child abuse.

- **Location-related insights:**
  5. The five areas with the highest number of incidents were Central, 77th Street, Pacific, Southwest, and Hollywood.
  6. The five areas with the highest average reporting delay were Devonshire, Topanga, West Valley, Van Nuys, and Foothill. Conversely, the five areas with the lowest average reporting delay were Central, Newton, Hollenbeck, Southeast, and Rampart.
  7. Among the twelve location combinations with more than 1,000 recorded incidents, the coordinates lat: 34.0561, lon: -118.2375 exhibit the highest concentration, with 2,397 incidents. This suggests a significant clustering of criminal activity in that specific area.
  8. The five areas with the highest percentage of closed cases were Mission, West Valley, Topanga, Foothill, and Harbor. Conversely, the five areas with the lowest percentage of closed cases were Pacific, Central, West LA, Hollywood, and Newton.
- **Time-related insights:**
  1. Afternoon was the most dangerous period of day in terms of reported crime incidents, followed by evening, morning, and night.
  2. January witnessed the highest number of incidents during the 2020–2025 period. It was followed by March, February, October, July, August, April, May, June, September, November, and December.
  3. The year with the highest number of recorded incidents was 2022, followed by 2023, 2021, 2020, 2024, and 2025.
  4. The crime types with the highest average reporting delay were crime against child, sex offender registrant out of compliance, unlawful sex, attempted theft by dishonest employee, and child-related lewd/lascivious acts. In contrast, the five crime types with the highest average reporting delay were train wrecking, attempted purse snatching, school disruption, minor theft at auto repair shop, and lynching.
  5. On average, incidents in our dataset were reported 12.2 days after they occurred.
  6. The day of the week in which the most crimes happened was Friday.

It is followed by Saturday, Wednesday, Thursday, Monday, Sunday, and Tuesday.

7. The number of recorded incidents steadily increased the period 2020–2022, remained stable throughout 2023, abruptly decreased during 2024 and remained relatively stable with minor fluctuations during the first five months of 2025.
8. 12 PM was the time of day when most crimes occurred. Incident numbers showed a slight increase after noon, followed by a sharp rise after 1 PM.

- **Weapon-related insights:**

1. Armed incidents accounted for 32.56% of the total.
2. The premise types with the highest number of armed incidents were streets, single family dwellings, multi-unit dwellings, sidewalks, and parking lots.

This study has yielded meaningful insights into the patterns and dynamics of daily crime in Los Angeles. However, it is not without its limitations and weaknesses:

- **Inaccuracies in the demographic data of the victims:** In the raw dataset, multiple incidents had invalid victim age values (negative numbers, zero age, unrealistic values above 90), as well as missing or invalid values for the victim's sex (e.g., "H", "–", blanks) and descent (e.g., "–", blanks). These issues occurred for various reasons that were not registered by the authorities and remain unknown to us. To reach the conclusions presented earlier regarding the age, sex and descent of the victims, we had to:
    1. Format the raw CSV file of the dataset by assigning a specific placeholder value to the records with incorrect or unknown age, sex and descent.
    2. Create visual-level filters in selected Power BI visualizations, following the CSV formatting step, to exclude all records containing the placeholder value.
- **Date and time formatting incompatibility:** Initially, `Date Rptd`, `DATE_OCC` and `TIME_OCC` had an incorrect format. To import a version of the data related to these columns into our database – intended for

subsequent query creation – we had to:

1. Remove the time from `Date Rptd` and `DATE_OCC`, since it was consistently set to 12:00:00 AM across all records, and convert them from date type to text strings formatted in ISO standard (yyyy–MM–dd).
2. Convert time values from a four-digit format (e.g., “1715”) to standard 24-hour format with a colon (e.g., “17:15”).

- **Data unreliability for 2024 and 2025:** The incident figures for the aforementioned years likely deviate from their actual counts, as each year from 2020 to 2023 recorded at least 200,000 incidents, whereas 2024 and 2025 show only 128,000 and fewer than 1,000 incidents, respectively. This discrepancy is most likely due to delays in incident uploads by the police department.

5.2. Suggestions:

Based on the findings and limitations identified in this study, the following suggestions are proposed to enhance future analyses and support crime prevention efforts.

- **Increased policing:** The LAPD, with the support from the State of Los Angeles, should consider increasing the number of police officers in response to the following patterns:
  1. Days of the week with the highest incident rates, such as Friday.
  2. High-risk hours of the day, particularly around 12 PM and afternoon hours in general.
  3. Months with elevated crime levels, for example, January.
  4. High-incident areas such as Central, low-resolution-rate areas such as Pacific, and specific crime hotspots – for instance, the location at coordinates `lat`: 34.0561, `lon`: -118.2375.

  This would allow for more targeted resource allocation and faster response times in high-risk zones.
- **Quicker reporting of incidents:** The police department should prioritize speeding up incident reporting in areas with high average reporting delays – such as Devonshire – and especially in crime types that consistently show delayed reporting, like crimes against children.

This can be achieved by:

1. Accelerating reporting-related procedures, e.g., reducing bureaucracy and implementing modern reporting mechanisms.
2. Hiring additional police personnel to assist with the reporting process.
3. Ensuring that incidents perceived as low-severity are still reported and documented.
4. Reducing hesitation, fear, or lack of trust in authorities on the victim's part by offering witness protection programs and other supportive incentives.

This would improve the timeliness and completeness of incident data, allowing for more accurate analyses and better-targeted prevention strategies.

- **Timely upload of incidents for future analyses:** Authorities should ensure that incidents from the previous year are uploaded within the first few months of the current year at the latest. For example, in our dataset, the year 2024 and the first five months of 2025 appear to be missing a significant number of incidents, compared to the years 2020–2023. Since the current year is 2025, all incidents of 2024 should have been already uploaded by the end of the first semester of 2025. This would allow data scientists to conduct similar analyses to ours, but with more recent and complete data.
- **Facilitation of future analyses:** Los Angeles Open Data Portal should enrich the dataset with columns and measures similar to those we created in our project, such as `day_of_week`, `period_of_day`, `hour_of_day`, `delay`, `avg_delay`, `avg_vict_age`, `weapon_percent` and `closed_percent`. This way, data scientists' work would be expedited, as they would not need to recreate these calculated columns every time they initiate an analysis.
- **Avoid omitting victim-related data:** Police should ensure that all the victim-related data is accurately recorded, including the victim's age, sex, and descent. This information is crucial for the speed and accuracy of future analyses, as it allows data scientists to reach significant conclusions without having to exclude values — such as age zero — or assign placeholder values — such as "X" for sex or

descent — to records with missing data.

- **Inclusion of perpetrator-related data:** It is recommended that future datasets also include perpetrator-related information (e.g., age, sex, descent), similar to the data currently provided for victims. Such additions would enhance the analytical potential of the dataset and support more informed decision-making.

Overall, the insights and recommendations presented in this project highlight the potential of data-driven approaches in understanding and addressing urban crime. By improving the quality, timeliness, and analytical accessibility of data, authorities and researchers can make better-informed decisions that contribute to public safety and community well-being.

6. References

1. **Centers for Disease Control and Prevention.** (2025, June 5). *Life expectancy*. National Center for Health Statistics. https://www.cdc.gov/nchs/fastats/life-expectancy.htm

2. **Fisk, N. R., Ng, M. K. M, Shabrina, Z.** (2025). *Advancing Spatiotemporal Prediction using Artificial Intelligence: Extending the Framework of Geographically and Temporally Weighted Neural Network (GTWNN) for Differing Geographical and Temporal Contexts.* arXiv. https://arxiv.org/abs/2503.22751

3. **hemil26.** (2025, March). *Los Angeles crime data*. Kaggle. https://www.kaggle.com/datasets/hemil26/crime-in-los-angeles

4. **Jenga, K., Catal, C., & Kar, G.** (2023). *Machine learning in crime prediction.* Journal of Ambient Intelligence and Humanized Computing. https://link.springer.com/article/10.1007/s12652-023-04530-y

5. **LAPD OpenData.** (2025, March). *Crime data from 2020 to present*. Los Angeles Open Data Portal. https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8/about_data

6. **Mandalapu, V., Elluri, L., Vyas, P., & Roy, N.** (2023). *Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions.* arXiv. https://arxiv.org/abs/2303.16310

7. **Microsoft.** (2024, July 15). *Microsoft SQL Server 2022 Express*. https://www.microsoft.com/en-us/download/details.aspx?id=104781

8. **Microsoft.** (2025, April 8). *Install SQL Server Management Studio*. https://learn.microsoft.com/en-us/ssms/install/install

9. **Microsoft.** (2025, July 23). *Microsoft Power BI Desktop*. https://www.microsoft.com/en-us/download/details.aspx?id=58494

10. **Wolpert, S.** (2015, October 7). Predictive policing substantially reduces crime in Los Angeles during months-long test. UCLA. https://newsroom.ucla.edu/releases/predictive-policing-substantially-reduces-crime-in-los-angeles-during-months-long-test

7. Appendix

1.

I) Initiate Excel on a blank workbook.

II) Click the button to get data from text/csv file.

III) Select the crime_in_la_raw.csv,

IV) Choose “65001: Unicode (UTF-8)” as file origin, “Comma” as delimiter, “Do not detect data types” as data type detection and pressing “Transform”.

2.

I) On the Home tab, above the “Transform” section, choose “Use First Row as Headers” to transform column names from Column1, Column2 etc to `DR_NO`, `Date Rptd` etc.

II) Delete the “Changed Type” step that is added automatically and is located below the Promoted Headers step in the Query Settings Tab.

III) Click on `Date Rptd` column, head to the “Transform” tab and above the “Text Column” section, click “Extract” and then “First Characters”.

IV) Type 10 as the number of starting characters to keep and press “OK”.

V) Click on the symbol “ABC” that is placed on the left of the title of our `Date Rptd` column and select “Using locale…”.

VI) Choose “Date” as the data type, “English (United States)” as the locale and press “OK”.

VII) Open the Add Column tab and above the “General” section, choose “Custom Column”.

VIII) Type “`Date Rptd New`” as new column name and in the Custom column formula box, use the expression:

```
Date.ToText([Date Rptd], "yyyy-MM-dd")
```

and press “OK”.

IX) Click on the symbol “ABC 123” that is placed on the left of the title of our `Date Rptd New` column and select “Text” as the Data Type.

X) Delete `Date Rptd` column, rename `Date Rptd` New column by double clicking on its name and then typing “`Date Rptd`”. After that, press enter.

XI) Bring the new `Date Rptd` column to the left of the `DATE OCC` column by

dragging it exactly where the old `Date Rptd` column was.

3.

```
Date.ToText([DATE OCC], "yyyy-MM-dd")
```

4.

I) With Power Query open, head to the “Add Column” tab and above the “General” section, click “Custom Column”.
II) On the window that pops up, give a name to our new `TIME OCC` custom column e.g., “`TIME OCC New`” and fill the “Custom column formula” field with the following M code:

```
Text.PadStart(
Text.From(
Number.IntegerDivide(Number.FromText([TIME OCC]),100)),2,"0")
&
":"
&
Text.PadStart(
Text.From(
Number.Mod(Number.FromText([TIME OCC]), 100)), 2, "0")
```

and click “OK”.
III) Click on the symbol “ABC 123” that is placed on the left of the title of our `TIME OCC New` column and select “Text” as the Data Type.
IV) Delete the original `TIME OCC` column by selecting it by right-clicking and choosing “Remove”.
V) Double click on the `TIME OCC New` column header and rename it to “`TIME OCC`”.
VI) Bring the new `TIME OCC` column to the right of the `DATE OCC` column by dragging it exactly where the old `TIME OCC` column was.

5.

I) Click on the symbol “ABC” that is found on the left of the title of our `Vict`

Age's column and select "Whole Number" as the Data Type.

II) Head to the "Add Column" tab and above the "General" section, click "Conditional Column".

III) On the opening window, give a name to our new Vict Age conditional column e.g., "Vict Age New", create the three clauses depicted below and press "OK":

- If Vict Age < 0 Then 0,
- Else If Vict Age > 90 Then 0,
- Else Vict Age.

IV) Click on the symbol "ABC 123" that is placed on the left of the title of our Vict Age New column and select "Whole Number" as the Data Type.

V) Delete the original Vict Age column by selecting it by right-clicking and choosing "Remove".

VI) Double click on the Vict Age New column header and rename it to "Vict Age".

VII) Bring the new Vict Age column to the right of the Mocodes column by dragging it exactly where the old Vict Age column was.

6.

I) Go to the "Add Column" tab and above the "General" section, select "Conditional Column".

II) On the window that pops up, think of a name for our new column, for example, "Vict Sex New", create the three clauses depicted below and press "OK":

- If Vict Sex equals M Then Vict Sex,
- Else If Vict Sex equals F Then Vict Sex,
- Else X.

III) Click on the symbol "ABC 123" that is placed on the left of the title of our Vict Sex New column and select "Text" as the Data Type.

IV) Delete the original Vict Sex column by selecting it by right-clicking and choosing "Remove".

V) Double click on the Vict Sex New column header and rename it to "Vict Sex".

VI) Bring the new Vict Sex column to the right of the Vict Age column by

dragging it exactly where the old `Vict Sex` column was.

7.

I) Go to the “Add Column” tab and above the “General” section, select “Conditional Column”.

II) On the opening window, type a name for our new column, for example, “`Vict Descent New`”, create the three clauses depicted below and press “OK”:

- If `Vict Descent` equals **-** Then X,
- Else If `Vict Descent` equals (leave this field blank) Then X,
- Else `Vict Descent`.

III) Click on the symbol “ABC 123” that is placed on the left of the title of our `Vict Descent New` column and select “Text” as the Data Type.

IV) Delete the original `Vict Descent` column by selecting it by right-clicking and choosing “Remove”.

V) Double click on the `Vict Descent New` column header and rename it to “`Vict Descent`”.

vi) Bring the new `Vict Descent` column to the right of the `Vict Sex` column by dragging it exactly where the old `Vict Descent` column was.

8.

Click on the ABC icon that exists on the left of each column title and select “Whole Number”.

9.

I) Trim all text columns by holding down “CTRL” and left-clicking each column and then, right-clicking on one of them, going to “transform” and pressing “Trim”.

II) Clean all text columns by following the last step’s process but instead of pressing “Trim”, click on “Clean”.

III) Locate “Close & Load” on the top-left corner, click on the arrow below it and then choose “Close & Load to…”.

IV) Select “Table” on the question of “How do you want to view this data in your workbook?” and “New worksheet” on the question of “Where do you want to put the data?”. After that, press “OK”.

V) Go to File, Options and Advanced.

VI) Untick “Use system separators” and set “.” as the Decimal Separator and “,” as the Thousands separator.
VII) Go back to “crime_in_la_raw” worksheet, press “CTRL + A” to select all data and “CTRL + C” to copy them.
VIII) Click on “Sheet1”, right-click on cell A1, go to “Paste Special” and choose “Values”. Then, rename “Sheet1” to “crime_in_la_cleaned”.
IX) Go to File and Save As. After that, select “CSV UTF-8” as file type, give the file a name, for example, crime_in_la_cleaned.csv and save the file in the desired file path.
X) Optionally, if you want to save the current Excel session, go to File, Save As, choose the file type “xlsx”, give the file a name e.g., crime_in_la_cleaned.xlsx and save anywhere in your computer.
XI) Open the CSV with Notepad++ to check the values of date columns, latitude and longitude. Don’t open the CSV with Excel because the depiction of these columns will change due to your system’s locale settings!

10.

I) Double click on the downloaded “SQL2022-SSEI-Expr.exe” file.
II) On the “Select an installation type” window, choose “Basic” as an installation type to install the SQL Server Database Engine feature with default configuration.
III) After having read the “License Terms and Privacy Statement” that appeared in the “Microsoft SQL Server License Terms” window, click “Accept”.
IV) When the “Specify SQL Server install location” appears, choose the location of the installation e.g., C:\Program Files\Microsoft SQL Server and click “Install”.

11.

I) Double click on the downloaded “SSMS-Setup-ENU.exe” file.
II) When the window with the title “Installing — SQL Server Management Studio 21” appears, specify SSMS install location, for example, C:\Program Files (x86)\Microsoft SQL Server Management Studio 20 and click “Install”.

12.

I) Opening SSMS.

II) In the “Connect to Server” pop-up window, on Login tab, fill in the Server name field, for example, DESKTOP-27F53GQ\SQLEXPRESS (if you are not able to recall the name of your computer, with the assist of **cmd.exe** and by typing the command “echo %COMPUTERNAME%”, your computer’s name can be retrieved in no time). Choose “Windows Authentication” as means of authentication and “Optional” as encryption method. After that, click “Connect”.

13.

I) Right-clicking on “Databases” in the Object Explorer window and selecting “New Database”.

II) In the New Database dialog, enter your desired Database Name (in our case `Crime_DB`) and click “OK” after keeping the default settings.

14.

I) Right-click on our database and choose New Query.

II) On the blank query that appears, type the code below and click “Execute”.

```
CREATE TABLE dbo.crime_import(
dr_no INT NOT NULL,
date_rptd DATE NOT NULL,
date_occ DATE NOT NULL,
time_occ TIME NOT NULL,
area INT NOT NULL,
area_name VARCHAR(50) NOT NULL,
rpt_dist_no INT NOT NULL,
part_1_2 INT NOT NULL,
crm_cd INT NOT NULL,
crm_cd_desc VARCHAR(200) NOT NULL,
mocodes VARCHAR(200) NULL,
vict_age INT NOT NULL,
vict_sex CHAR(1) NOT NULL,
vict_descent CHAR(1) NOT NULL,
premis_cd INT NULL,
premis_desc VARCHAR(200) NOT NULL,
weapon_used_cd INT NULL,
weapon_desc VARCHAR(200) NULL,
status CHAR(2) NOT NULL,
status_desc VARCHAR(50) NOT NULL,
crm_cd_1 INT NULL,
crm_cd_2 INT NULL,
crm_cd_3 INT NULL,
crm_cd_4 INT NULL,
location VARCHAR(100) NOT NULL,
```

```
cross_street VARCHAR(100) NULL,
lat FLOAT NOT NULL,
lon FLOAT NOT NULL);
```

15.

I) Delete the already written query or start a new query.

II) Execute after using the chunk of code below:

```
BULK INSERT dbo.crime_import
FROM 'D:\Other Programs\Master\Project\crime_in_la_cleaned.csv'
WITH(
ROWTERMINATOR = '\n',
FIELDTERMINATOR = ';',
FIRSTROW = 2,
CODEPAGE = 65001,
TABLOCK);
```

16.

```
select * from dbo.crime_import;
```

Results:

dr_no	date_rptd	date_occ	…
211507896	2021-04-11	2020-11-07	…
201516622	2020-10-21	2020-10-18	…
240913563	2024-12-10	2020-10-30	…
210704711	2020-12-24	2020-12-24	…
201418201	2020-10-03	2020-09-29	…
…	…	…	…

17.

I) Delete bulk insert's query or start a new query.

II) Press "Execute" after writing the code below:

```
CREATE TABLE dbo.dimArea (
area_id INT IDENTITY (1,1) PRIMARY KEY,
area INT NOT NULL,
area_name VARCHAR(50) NOT NULL,
rpt_dist_no INT NOT NULL,
location VARCHAR(100) NOT NULL,
cross_street VARCHAR(100) NULL,
lat FLOAT NOT NULL,
lon FLOAT NOT NULL);
```

18.

```
CREATE TABLE dbo.dimCrime (
crm_id INT IDENTITY (1,1) PRIMARY KEY,
```

```
part_1_2 INT NOT NULL,
crm_cd INT NOT NULL,
crm_cd_desc VARCHAR(200) NOT NULL,
mocodes VARCHAR(200) NULL,
crm_cd_1 INT NULL,
crm_cd_2 INT NULL,
crm_cd_3 INT NULL,
crm_cd_4 INT NULL,
status CHAR(2) NOT NULL,
status_desc VARCHAR(50) NOT NULL);
```

19.

```
CREATE TABLE dbo.dimPremise (
premis_id INT IDENTITY (1,1) PRIMARY KEY,
premis_cd INT NULL,
premis_desc VARCHAR(200) NOT NULL);
```

20.

```
CREATE TABLE dbo.dimVictim (
vict_id INT IDENTITY (1,1) PRIMARY KEY,
vict_age INT NOT NULL,
vict_sex CHAR(1) NOT NULL,
vict_descent CHAR(1) NOT NULL);
```

21.

```
CREATE TABLE dbo.dimWeapon (
weapon_id INT IDENTITY (1,1) PRIMARY KEY,
weapon_used_cd INT NULL,
weapon_desc VARCHAR(200) NULL);
```

22.

```
CREATE TABLE dbo.factCrime (
dr_no INT PRIMARY KEY,
date_rptd DATE NOT NULL,
date_occ DATE NOT NULL,
time_occ TIME NOT NULL,
area_id INT NOT NULL FOREIGN KEY REFERENCES
dbo.dimArea(area_id),
crm_id INT NOT NULL FOREIGN KEY REFERENCES dbo.dimCrime(crm_id),
premis_id INT NOT NULL FOREIGN KEY REFERENCES
dbo.dimPremise(premis_id),
vict_id INT NOT NULL FOREIGN KEY REFERENCES
dbo.dimVictim(vict_id),
weapon_id INT NOT NULL FOREIGN KEY REFERENCES
dbo.dimWeapon(weapon_id));
```

23.

```
INSERT INTO dbo.dimArea (area, area_name, rpt_dist_no, location,
cross_street, lat, lon)
SELECT DISTINCT area, area_name, rpt_dist_no, location,
cross_street, lat, lon
```

```
FROM dbo.crime_import;
```

24.

```
SELECT * FROM dbo.dimArea;
```

Results:

area_id	area	area_name	rpt_dist_no	…
1	1	Central	101	…
2	1	Central	101	…
3	1	Central	101	…
4	1	Central	101	…
5	1	Central	101	…
…	…	…	…	…

25.

```
INSERT INTO dbo.dimCrime (part_1_2, crm_cd, crm_cd_desc,
mocodes, crm_cd_1, crm_cd_2, crm_cd_3, crm_cd_4,status,
status_desc)
SELECT DISTINCT part_1_2, crm_cd, crm_cd_desc, mocodes,
crm_cd_1, crm_cd_2, crm_cd_3, crm_cd_4, status, status_desc
FROM dbo.crime_import;
```

26.

```
SELECT * FROM dbo.dimCrime;
```

Results:

crm_id	part_1_2	crm_cd	crm_cd_desc	…
1	1	110	CRIMINAL HOMICIDE	…
2	1	110	CRIMINAL HOMICIDE	…
3	1	110	CRIMINAL HOMICIDE	…
4	1	110	CRIMINAL HOMICIDE	…
5	1	110	CRIMINAL HOMICIDE	…
…	…	…	…	…

27.

```
INSERT INTO dbo.dimPremise (premis_cd, premis_desc)
SELECT DISTINCT premis_cd, premis_desc
FROM dbo.crime_import;
```

28.

```
SELECT * FROM dbo.dimPremise;
```

Results:

premis_id	premis_cd	premis_desc
1	248	…
2	510	…
3	835	…
4	748	…
5	745	…
…	…	…

29.

```
INSERT INTO dbo.dimVictim (vict_age, vict_sex, vict_descent)
SELECT DISTINCT vict_age, vict_sex, vict_descent
FROM dbo.crime_import;
```

30.

```
SELECT * FROM dbo.dimVictim;
```

Results:

vict_id	vict_age	vict_sex	vict_descent
1	24	M	A
2	62	M	A
3	81	M	A
4	25	M	J
5	44	M	J
…	…	…	…

31.

```
INSERT INTO dbo.dimWeapon (weapon_used_cd, weapon_desc)
SELECT DISTINCT weapon_used_cd, weapon_desc
FROM dbo.crime_import;
```

32.

```
SELECT * FROM dbo.dimWeapon;
```

Results:

weapon_id	weapon_used_cd	weapon_desc
1	509	ROPE/LIGATURE
2	506	FIRE
3	212	BOTTLE
4	123	M1-1 SEMIAUTOMATIC ASSAULT RIFLE
5	512	MACE/PEPPER SPRAY
…	…	…

33.

```
BEGIN TRANSACTION;

INSERT INTO dbo.factCrime (dr_no, date_rptd, date_occ, time_occ,
area_id, crm_id, premis_id, vict_id, weapon_id)
SELECT ci.dr_no, ci.date_rptd, ci.date_occ, ci.time_occ,
da.area_id, dc.crm_id, dp.premis_id, dv.vict_id, dw.weapon_id
FROM dbo.crime_import ci
JOIN dbo.dimArea da ON
ci.area = da.area AND
ci.area_name = da.area_name AND
ci.rpt_dist_no = da.rpt_dist_no AND
ci.location = da.location AND
ISNULL(ci.cross_street, '') = ISNULL(da.cross_street, '') AND
ci.lat = da.lat AND
ci.lon = da.lon
JOIN dbo.dimCrime dc ON
ci.part_1_2 = dc.part_1_2 AND
ci.crm_cd = dc.crm_cd AND
ci.crm_cd_desc = dc.crm_cd_desc AND
ISNULL(ci.mocodes, '') = ISNULL(dc.mocodes, '') AND
ISNULL(ci.crm_cd_1, 0) = ISNULL(dc.crm_cd_1, 0) AND
ISNULL(ci.crm_cd_2, 0) = ISNULL(dc.crm_cd_2, 0) AND
ISNULL(ci.crm_cd_3, 0) = ISNULL(dc.crm_cd_3, 0) AND
ISNULL(ci.crm_cd_4, 0) = ISNULL(dc.crm_cd_4, 0) AND
ci.status = dc.status AND
ci.status_desc = dc.status_desc
JOIN dbo.dimPremise dp ON
ISNULL(ci.premis_cd, 0) = ISNULL(dp.premis_cd, 0) AND
ci.premis_desc = dp.premis_desc
JOIN dimVictim dv ON
ci.vict_age = dv.vict_age AND
ci.vict_sex = dv.vict_sex AND
ci.vict_descent = dv.vict_descent
JOIN dimWeapon dw ON
ISNULL(ci.weapon_used_cd, 0) = ISNULL(dw.weapon_used_cd, 0) AND
ISNULL(ci.weapon_desc, '') = ISNULL(dw.weapon_desc, '')
WHERE NOT EXISTS (
SELECT 1 FROM dbo.factCrime fc WHERE fc.dr_no = ci.dr_no);

COMMIT;
```

34.

```
FROM dbo.crime_import ci
JOIN dbo.dimArea da ON
ci.area = da.area AND
ci.area_name = da.area_name AND
ci.rpt_dist_no = da.rpt_dist_no AND
ci.part_1_2 = da.part_1_2 AND
ci.location = da.location AND
ISNULL(ci.cross_street, '') = ISNULL(da.cross_street, '') AND
```

```
ci.lat = da.lat AND
ci.lon = da.lon
```

35.

```
SELECT * FROM dbo.factCrime;
```

Results:

dr_no	date_rptd	date_occ	time_occ	…
817	2020-09-20	2020-09-19	17:00:00.0000000	…
2113	2021-06-21	2021-04-05	18:35:00.0000000	…
2203	2020-12-31	2022-12-19	19:30:00.0000000	…
2315	2023-07-23	2023-07-22	18:00:00.0000000	…
2401	2024-01-10	2023-12-21	19:30:00.0000000	…
…	…	…	…	…

36.

I) Right-click on “Database Diagrams” and choose “New Database Diagram”.

II) A pop-up window with the question “This database does not have one or more of the support objects required to use database diagramming. Do you wish to create them?” will appear. Click “Yes”.

III) A new window will reveal its presence with the name “Add Table”, containing all of our tables’ names. Select them all, except for the crime_import table, by holding down “CTRL” and left-clicking and then click the button “Add”. In the background, you will notice a new SSMS window with our diagram created. Since we are done here, click “Close” to close the “Add Table” window.

IV) Lastly, right-click on the new SSMS window and select “Save Diagram_0”. Enter a desired name for our diagram, for example, “Crime_ERD” and click “OK”.

37.

```
SELECT area_id,
area_name as area,
lat,
lon
FROM dbo.dimArea;
```

Results:

area_id	area	lat	lon
1	Central	34.0663	-118.2442
2	Central	34.0649	-118.2465
3	Central	34.0646	-118.2474
4	Central	34.0693	-118.2488
5	Central	34.0703	-118.2484
…	…	…	…

38.

```
SELECT crm_id,
part_1_2,
crm_cd_desc as crime_type,
status_desc as status
FROM dbo.dimCrime;
```

Results:

crm_id	part_1_2	crime_type	status
1	1	CRIMINAL HOMICIDE	Adult Arrest
2	1	CRIMINAL HOMICIDE	Invest Cont
3	1	CRIMINAL HOMICIDE	Juv Arrest
4	1	CRIMINAL HOMICIDE	Invest Cont
5	1	CRIMINAL HOMICIDE	Adult Arrest
…	…	…	…

39.

```
SELECT premis_id,
premis_desc
FROM dbo.dimPremise;
```

Results:

premis_id	premis_desc
1	CELL PHONE STORE
2	NURSING/CONVALESCENT/RETIREMENT HOME
3	7TH AND METRO CENTER (NOT LINE SPECIFIED)
4	CULTURAL SIGNIFICANCE/MONUMENT
5	WATER FACILITY
…	…

40.

```
SELECT vict_id,
vict_age,
vict_sex,
vict_descent
FROM dbo.dimVictim;
```

Results:

vict_id	vict_age	vict_sex	vict_descent
1	24	M	A
2	62	M	A
3	81	M	A
4	25	M	J
5	44	M	J
…	…	…	…

41.

```
SELECT weapon_id
FROM dbo.dimWeapon;
```

Results:

weapon_id
1
2
3
4
5
…

42.

```
SELECT dr_no,
date_occ,
time_occ,
DATEDIFF(DAY, date_occ, date_rptd) AS delay,
area_id,
crm_id,
premis_id,
vict_id,
weapon_id
FROM dbo.factCrime;
```

Results:

dr_no	date_occ	time_occ	delay	area_id	…
817	2020-09-19	17:00:00.0000000	1	153161	…
2113	2021-04-05	18:35:00.0000000	77	117315	…
2203	2022-12-19	19:30:00.0000000	12	21363	…
2315	2023-07-22	18:00:00.0000000	1	137036	…
2401	2023-12-21	19:30:00.0000000	20	120859	…
…	…	…	…		…

43.

```
SELECT crm_cd_desc AS crime_type, COUNT(*) AS incident_count
FROM dbo.dimCrime dc
JOIN dbo.factCrime fc ON dc.crm_id = fc.crm_id
```

```
GROUP BY crm_cd_desc
ORDER BY incident_count DESC;
```

Results:

crime_type	incident_count
VEHICLE - STOLEN	115226
BATTERY - SIMPLE ASSAULT	74839
BURGLARY FROM VEHICLE	63517
THEFT OF IDENTITY	62539
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)	61092
…	…

44.

```
SELECT area_name as area, COUNT(*) AS incident_count
FROM dbo.dimArea da
JOIN dbo.factCrime fc ON da.area_id = fc.area_id
GROUP BY area_name
ORDER BY incident_count DESC;
```

Results:

area	incident_count
Central	69670
77th Street	61758
Pacific	59514
Southwest	57514
Hollywood	52429
…	…

45.

```
SELECT vict_descent, COUNT(*) as vict_count
FROM dbo.dimVictim dv
JOIN dbo.factCrime fc ON dv.vict_id = fc.vict_id
WHERE vict_descent NOT LIKE 'X'
GROUP BY vict_descent
ORDER BY vict_count DESC;
```

Results:

vict_descent	vict_count
H	296403
W	201441
B	135817
O	78005
A	21340
…	…

46.

```
SELECT part_1_2, COUNT(*) AS incident_count
```

```
FROM dbo.dimCrime dc
JOIN dbo.factCrime fc ON dc.crm_id = fc.crm_id
GROUP BY part_1_2
ORDER BY incident_count DESC;
```

Results:

part_1_2	incident_count
1	602715
2	402347

47.

```
SELECT crm_cd_desc AS crime_type, vict_sex, COUNT(*) AS
vict_count
FROM dbo.dimVictim dv
JOIN dbo.factCrime fc ON dv.vict_id = fc.vict_id
JOIN dbo.dimCrime dc ON dc.crm_id = fc.crm_id
WHERE vict_sex NOT LIKE 'X'
GROUP BY vict_sex, crm_cd_desc
ORDER BY vict_count DESC;
```

Results:

crime_type	vict_sex	vict_count
BATTERY - SIMPLE ASSAULT	M	39222
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	M	38173
THEFT OF IDENTITY	F	35804
INTIMATE PARTNER - SIMPLE ASSAULT	F	35508
BURGLARY FROM VEHICLE	M	35400
…	…	…

48.

```
SELECT premis_desc, COUNT(*) AS weapon_incident_count
FROM dbo.dimPremise dp
JOIN dbo.factCrime fc ON dp.premis_id = fc.premis_id
JOIN dbo.dimWeapon dw ON dw.weapon_id = fc.weapon_id
WHERE weapon_desc IS NOT NULL
GROUP BY premis_desc
ORDER BY weapon_incident_count DESC;
```

Results:

premis_desc	weapon_incident_count
STREET	69382
SINGLE FAMILY DWELLING	60159
MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)	54183
SIDEWALK	31847
PARKING LOT	18228
...	...

49.

```
SELECT crm_cd_desc AS crime_type, AVG(vict_age) as avg_vict_age
FROM dbo.dimVictim dv
JOIN dbo.factCrime fc ON dv.vict_id = fc.vict_id
JOIN dbo.dimCrime dc ON dc.crm_id = fc.crm_id
WHERE vict_age > 0
GROUP BY crm_cd_desc
ORDER BY avg_vict_age;
```

Results:

crime_type	average_vict_age
CHILD NEGLECT (SEE 300 W.I.C.)	7
CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER)	10
CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT	11
CHILD ABANDONMENT	11
CHILD ABUSE (PHYSICAL) - AGGRAVATED ASSAULT	11
...	...

50.

```
SELECT time_occ, COUNT(*) AS incident_count
FROM dbo.factCrime
GROUP BY time_occ
ORDER BY incident_count DESC;
```

Results:

time_occ	incident_count
12:00:00.0000000	35205
18:00:00.0000000	26579
17:00:00.0000000	25195
20:00:00.0000000	24780
19:00:00.0000000	23083
...	...

51.

```
SELECT DATEPART(year, date_occ) as year, DATEPART(month,
date_occ) as month, COUNT(*) AS incident_count
FROM dbo.factCrime
GROUP BY DATEPART(year, date_occ), DATEPART(month, date_occ)
ORDER BY year, month;
```

Results:

year	month	incident_count
2020	1	18576
2020	2	17284
2020	3	16188
2020	4	15706
2020	5	17230
...	...	...

52.

```
SELECT
(SELECT COUNT(*)
FROM dbo.factCrime
WHERE weapon_id <> 10) * 1.0 /
(SELECT COUNT(*)
FROM dbo.factCrime) as weapon_percent;
```

Results:

weapon_percent
0.325597823815

53.

```
SELECT lat, lon, COUNT(*) AS incident_count
FROM dbo.dimArea da
JOIN dbo.factCrime fc ON da.area_id = fc.area_id
WHERE lat <> 0 AND lon <> 0
GROUP BY lat, lon
ORDER BY incident_count desc;
```

Results:

lat	lon	incident_count
34.0561	-118.2375	2397
34.0483	-118.2631	1826
34.244	-118.5583	1521
34.0428	-118.4582	1469
34.0736	-118.3563	1363
...	...	...

54.

```
SELECT crm_cd_desc as crime_type, AVG(DATEDIFF(DAY, date_occ,
date_rptd)) AS avg_delay
FROM dbo.factCrime fc
JOIN dbo.dimCrime dc ON fc.crm_id = dc.crm_id
GROUP BY crm_cd_desc
ORDER BY avg_delay desc;
```

Results:

crm_cd_desc	average_delay
CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER)	157
SEX OFFENDER REGISTRANT OUT OF COMPLIANCE	124
SEX,UNLAWFUL(INC MUTUAL CONSENT, PENETRATION W/ FRGN OBJ	120
LEWD/LASCIVIOUS ACTS WITH CHILD	105
DISHONEST EMPLOYEE ATTEMPTED THEFT	105
…	…

55.

```
SELECT area_name as area, AVG(DATEDIFF(DAY, date_occ,
date_rptd)) AS avg_delay
FROM dbo.factCrime fc
JOIN dbo.dimArea da ON fc.area_id = da.area_id
GROUP BY area_name
ORDER BY avg_delay desc;
```

Results:

area	average_delay
Devonshire	16
Topanga	15
Mission	14
Van Nuys	14
Foothill	14
…	…

56.

```
SELECT date_occ, COUNT(*) as incident_count
FROM dbo.factCrime fc
GROUP BY date_occ
ORDER BY incident_count desc;
```

Results:

date_occ	incident_count
1/1/2020	1164
1/1/2023	1159
2/12/2022	1132
1/2/2023	1093
1/10/2022	1079
…	…

57.

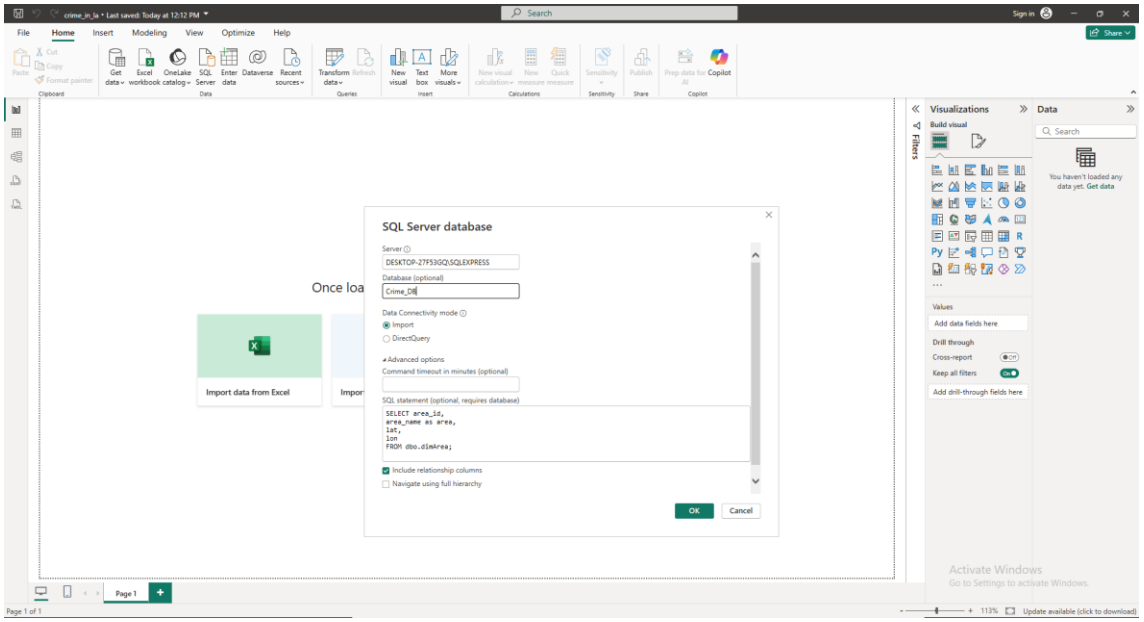
I) Double click on the downloaded “PBIDesktopSetup_x64.exe” file.

II) Select the language that we want Power BI to be installed in, for example, “English” and click on “Next”.

III) Click “Next” on the window with the message “The Setup Wizard will install Microsoft Power BI Desktop (x64) on your computer”.
IV) On the window with the title “Microsoft Software License Terms”, after having read the terms, check the box that says “I accept the terms in the License Agreement” and click on “Next” once more.
V) Specify the installation location, for example, C:\Program Files\Microsoft Power BI Desktop\ and click “Next” again.
VI) When the “Ready to install Microsoft Power BI Desktop (x64)” pops up, click “Install” to initiate the actual installation process and then wait until it is done (this could take a couple of minutes so patience is important).
VII) When installation is completed, a window with the title “Completed the Microsoft Power BI Desktop” will show its presence. Click on “Finish”.

58.

I) On the home tab, under the “Select a data source or start with a blank report” menu, choose the box with the title “SQL Server”.
II) A pop-up window will come up with the title “SQL Server database”. On the Server field, fill in the server’s name, “DESKTOP-27F53GQ\SQLEXPRESS” in our case. On the database field, type the name of our database, “`Crime_DB`” in our case. Leave “Data Connectivity mode” to “Import” and then click on “Advanced options” to reveal some extra and important fields, like the one of the SQL statement. When new fields appear, copy the code of the first SQL query we created earlier located at Appendix 37 — and paste it into the text box of the “SQL statement (optional, requires database)” field. Then, make sure that “Include relationship columns” checkbox is checked and press “OK”.



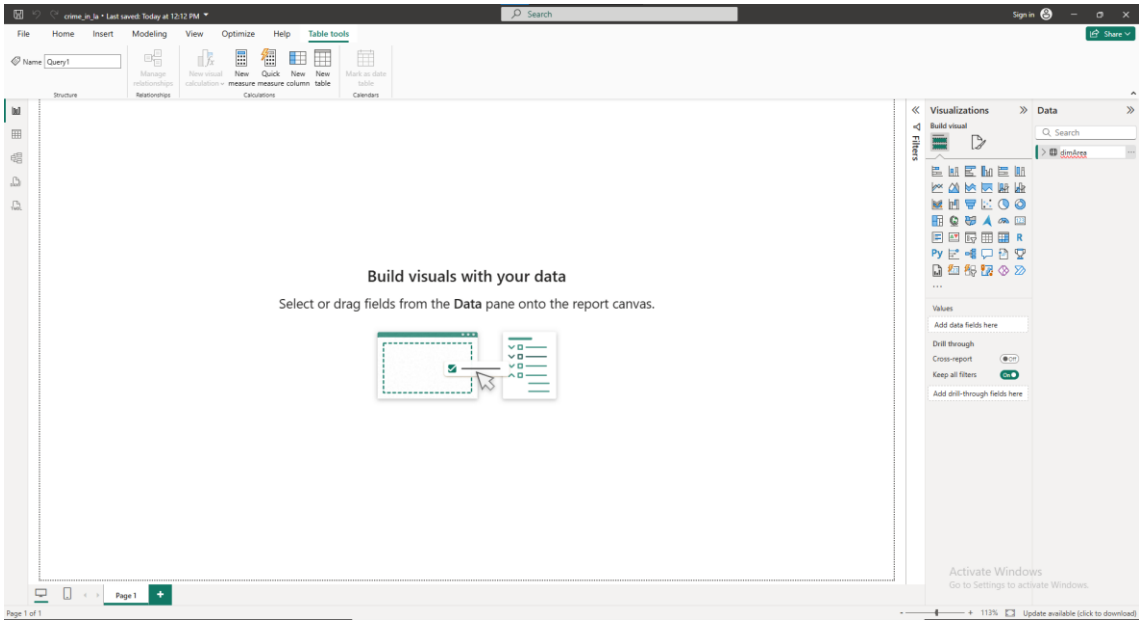
Screenshot 92: "SQL Server connection settings and query input in Power BI"

III) A new window will appear with the same title as the last one but with different options. In the "Windows" tab, locate the field "Use your Windows credentials to access this database" and click on "Use my current credentials" bullet (we won't be choosing the "Use alternate credentials" bullet because we did not set username/password during the creation of our server. Instead, we will be connecting through Windows Authentication). After that, make sure that the server is chosen and not the combination of server-database e.g., "desktop-27f53gq\sqlexpress;" and not "desktop-27f53gq\sqlexpress;Crime_DB", and press "Connect" (choosing the server and not the combination of server-database will prevent Power BI from reasking for authentication if we decide to create another database in the future, in the same SQL Server instance and attempt to load it into Power BI).

IV) In the "Encryption Support" window with the message "We were unable to connect to the data source using an encrypted connection. To access this data source using an encrypted connection, press "OK".

V) A window displaying the four columns of our first query — `area`, `area_id`, `lat`, and `lon` — will show up. Click "Load" to import the data into a dedicated table specifically designed for the dataset returned by the aforementioned query.

VI) After the table appears in the data pane as Query1, double-click it, rename it to “dimArea” and press “Enter” to apply the change.



Screenshot 93: “Table renamed”

VII) Finally, go to File and Save As. After that, select “Power BI file (*.pbix)” as file type, give the file a name, for example, “crime_in_la.pbix” and save the file in the desired file path. This step is critical to secure that the connection between the database and Power BI is saved and can later be used for the rest of the queries.

59.

I) On the Home tab, above the “Data” section, select “SQL Server”.

II) When the same “SQL Server database” window appears, fill in the name of the server and the name of the database once again. Then, copy second SQL query’s code — found in Appendix 38 — and paste it into “SQL statement”’s input box and click “OK”.

III) When the window displaying the four columns of our second query — `crime_type`, `crm_id`, `status`, and `part_1_2` — shows up, press “Load” to import the data into Power BI.

IV) After the table is imported, rename it to “dimCrime”.

60.

On the Home tab, click “Model view” on the left-hand side to display the schema.

61.

I) Right-click on `factCrime` that is located in the “Data” pane, on the right side of the home page and select the option “New column”.

II) In the input box that appears at the top of the page, delete any existing text, type the code below and press “Enter”:

```
period_of_day = SWITCH(TRUE(),
HOUR(factCrime[time_occ]) >= 0 && HOUR(factCrime[time_occ]) < 6,
"Night (00:00-05:59)",
HOUR(factCrime[time_occ]) >= 6 && HOUR(factCrime[time_occ]) <
12, "Morning (06:00-11:59)",
HOUR(factCrime[time_occ]) >= 12 && HOUR(factCrime[time_occ]) <
18, "Afternoon (12:00-17:59)",
HOUR(factCrime[time_occ]) >= 18 && HOUR(factCrime[time_occ]) <=
23, "Evening (18:00-23:59)")
```

62.

I) Right-click on `factCrime` and select “New column”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
day_of_week = FORMAT([date_occ], "dddd")
```

63.

I) Right-click on `factCrime` and select “New column”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
hour_of_day = FORMAT(HOUR(factCrime[time_occ]), "00") & ":00"
```

64.

I) Right-click on `factCrime` and select “New measure”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
avg_delay = AVERAGE(factCrime[delay])
```

65.

I) Right-click on `factCrime` and select “New measure”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
weapon_percent =
DIVIDE(
CALCULATE(COUNTROWS(factCrime), factCrime[weapon_id] <> 10),
COUNTROWS(factCrime))
```

66.

I) Right-click on `factCrime` and select “New measure”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
avg_vict_age =
AVERAGEX(
FILTER(factCrime, RELATED(dimVictim[vict_age]) > 0),
RELATED(dimVictim[vict_age]))
```

67.

I) Right-click on `factCrime` and select “New measure”.

II) In the appearing input box, delete any existing text, type the code depicted below and press “Enter”:

```
closed_percent =
DIVIDE(
CALCULATE(
COUNTROWS(factCrime),
dimCrime[Status] IN {"Adult Arrest", "Adult Other", "Juv
Arrest", "Juv Other"}),
CALCULATE(
COUNTROWS(factCrime),
NOT dimCrime[Status] IN {"UNK"}), 0)
```

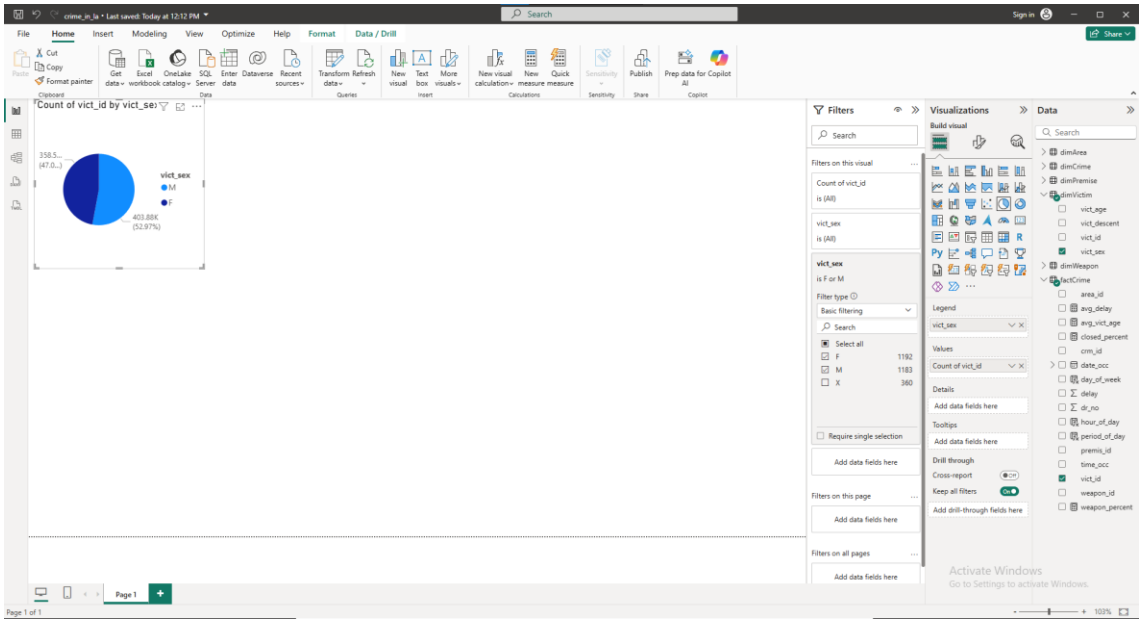
III) In the “Measure tools” tab, above the “Formatting” section, change measure’s format from “General” to “Percentage”.

68.

I) In the “Visualizations” pane, click on the “Pie chart”.

II) In the “Data” pane, expand `dimVictim` and `factCrime` to see all their fields. Then, drag `vict_sex` to “Legend” and any `factCrime`’s column (e.g. `vict_id`) to “Values”, both of which exist in “Visualizations”. Ensure that the aggregation type of the selected `factCrime` column is set to “Count” (we could have chosen any other `factCrime` column and have the exact same result without issues).

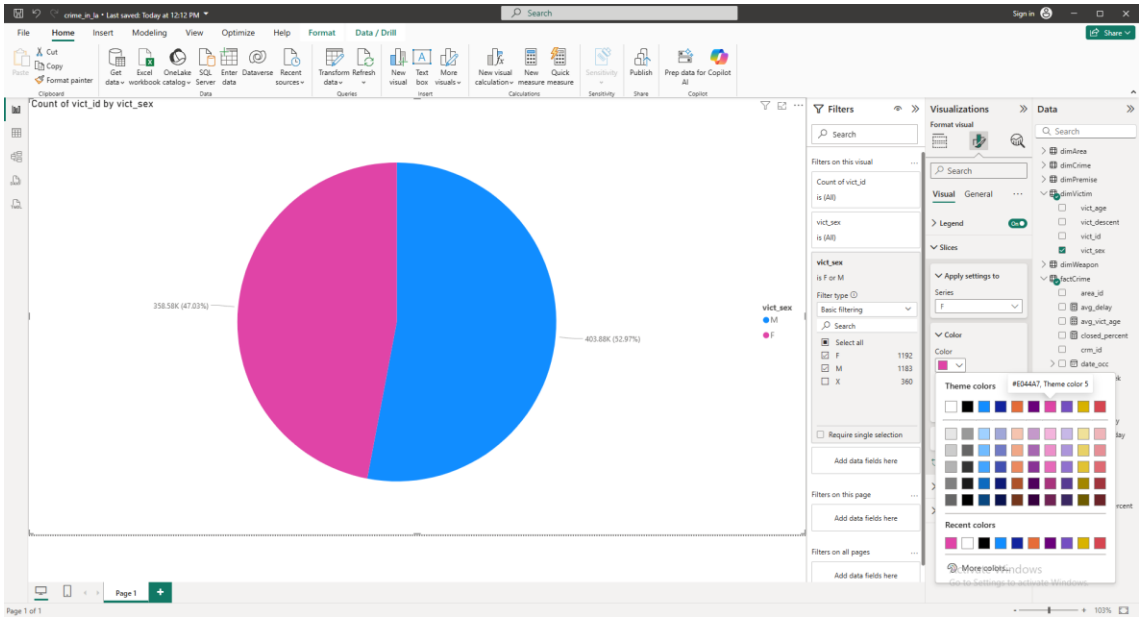
III) Drag `vict_sex` to “Add data fields here” box, which exists under the “Filters on this visual” menu of the “Filters” pane. Select “F” (females) and “M” (males) but not “X” (we want to exclude unknown sex from our chart because including it would distort the comparison between the two known categories).



Screenshot 94: “Sex filter applied”

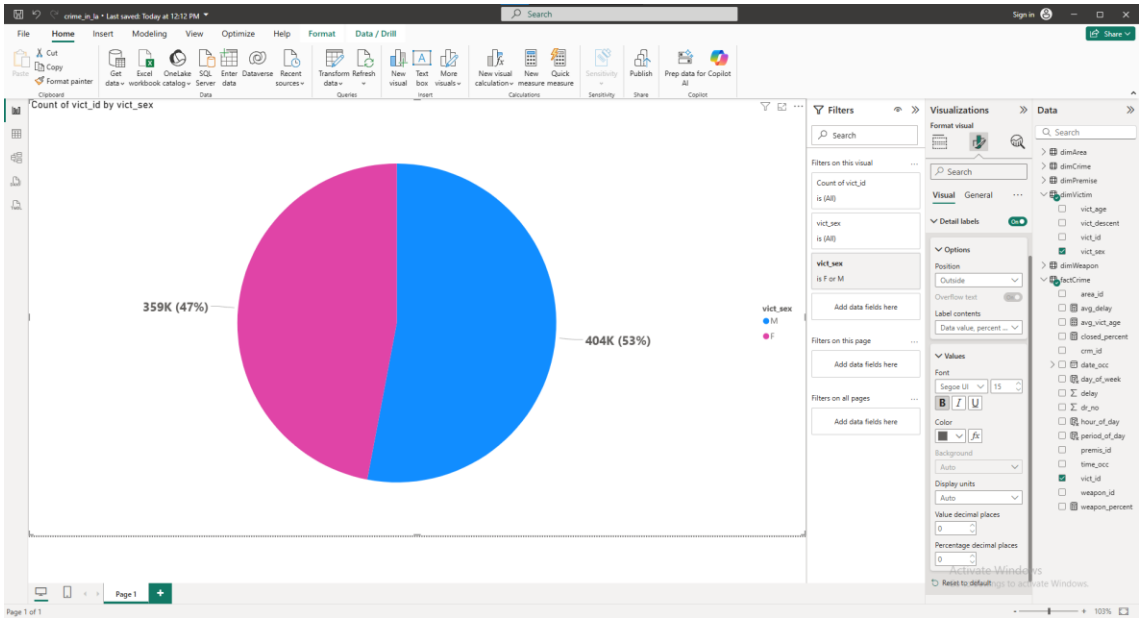
IV) Adjust chart’s height and width by dragging its bottom right corner diagonally to the right in order to fill the whole screen.

V) For better distinguishability, further differentiate the color used for females (“F”) from that of males (“M”) by going to “Format your visual” that exists under “Visualizations” pane, expanding “Slices” under “Visual” menu, expanding “Apply settings to” and choosing “F” as “Series”, expanding “Color” and choosing “#E044A7, Theme color 5” as “Color”.



Screenshot 95: “Female color altered”

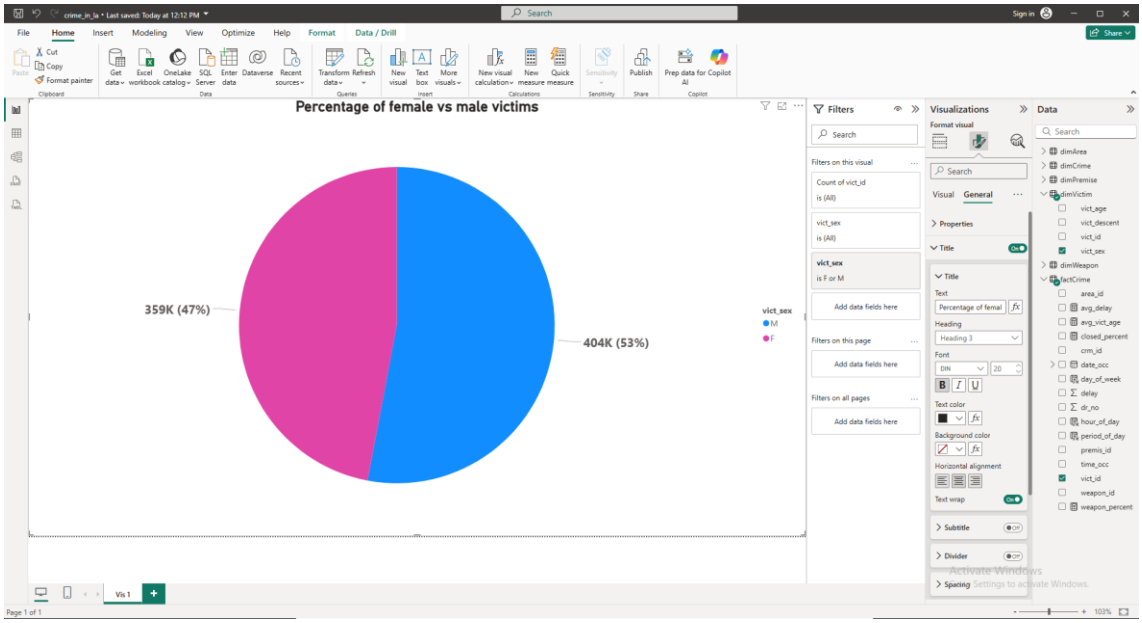
VI) Increase the font size of the chart’s values by expanding “Detail labels”, then “Values” and setting the font number from 9 to 15. Make chart’s values bold by clicking on the letter “B”, which stands for “Bold” and exists right under “Font” menu’s dropdown list. Next, reduce the decimal places of percentages by setting “Percentage decimal places” from “Auto” to 0. Moreover, reduce the decimal places of values by swapping “Percentage decimal places” from “Auto” to 0.



Screenshot 96: “Chart’s values formatted”

VII) Change the chart's title by clicking on "General" that also exists under "Visualizations", expanding "Title", deleting the pre-existing title text and assigning a new one e.g., "Percentage of female vs male victims". Make the title larger by setting font number from 14 to 20 and turn it to bold by clicking on the "B" icon. Then, move the new title to the top-middle of the canvas by clicking on "Center" that can be found under "Horizontal alignment" of the "Title" menu.

VIII) Rename the page by double-clicking on it, deleting its name and typing the new name e.g., "Vis 1".



Screenshot 97: "Chart's title and page name changed"

69.

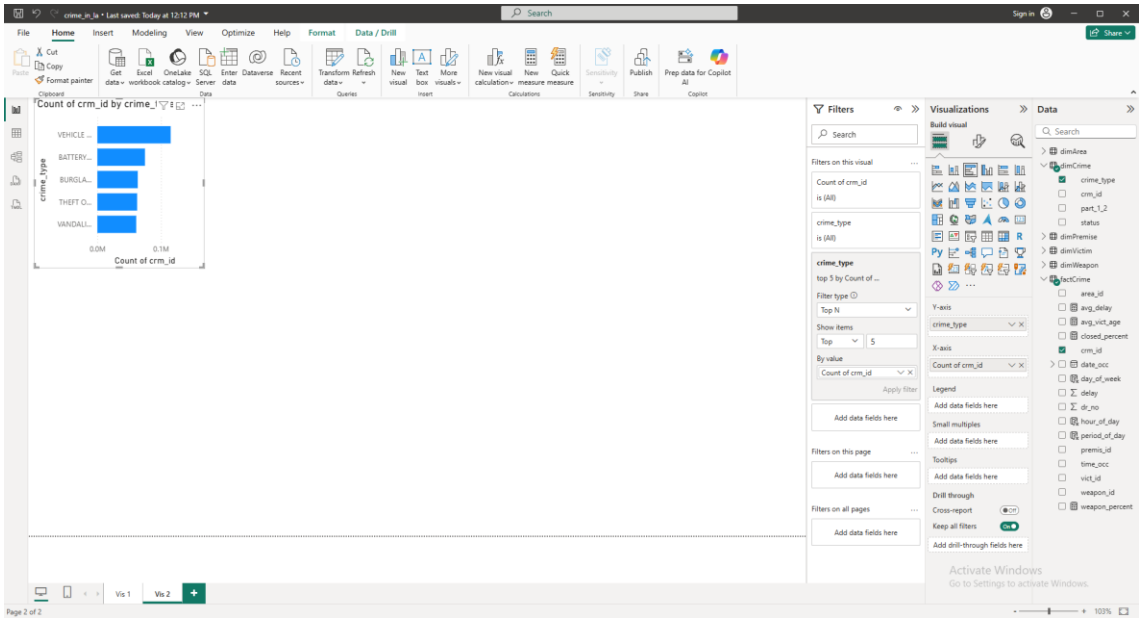
I) Create a new page with the name "Vis 2".

II) In the "Visualizations" pane, select the "Clustered bar chart".

III) Drag `factCrime`'s `crm_id` to "X-axis" and `dimCrime`'s `crime_type` to "Y-axis".

IV) Drag `crime_type` to "Filters on this visual". Set the "Filter type" of `crime_type` to "Top N". This will trigger the appearance of two new fields with the names "Show items" and "By value". "Show items" has a dropdown list with the category "Top" already selected. On the right side of the dropdown list, write the number 5 into the blank input box. "By value" has a box with the name "Add data fields here". Drag `factCrime`'s `crm_id` on top of it and click the button "Apply filter" that exists exactly below it. This filter will only allow the five crime

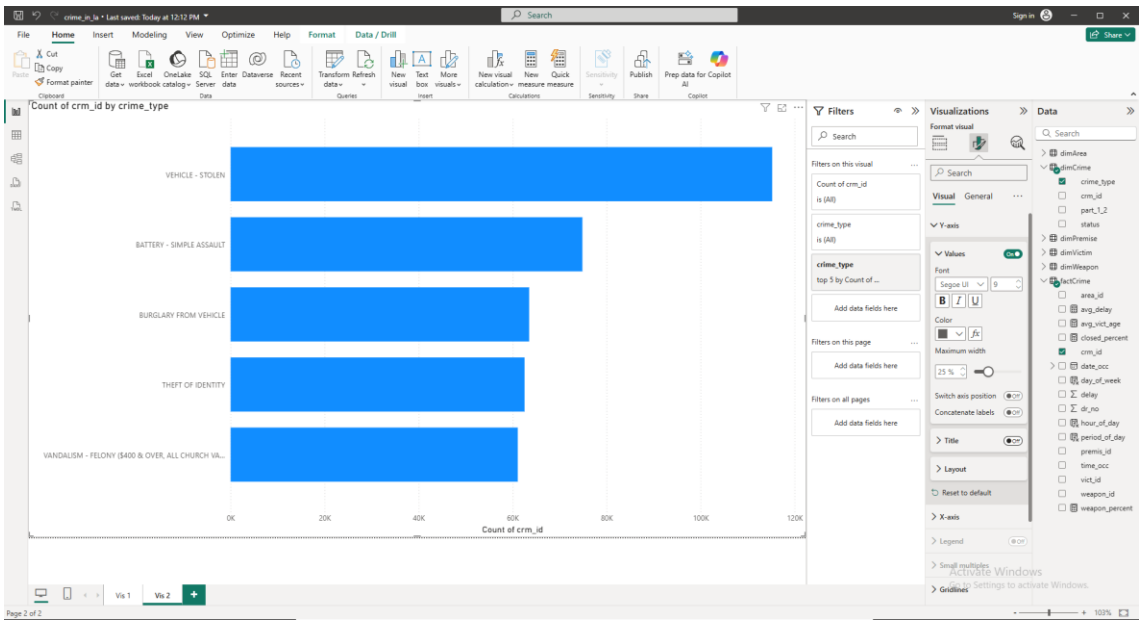
types with the highest number of victims to be shown in our chart.



Screenshot 98: “Top 5 filter applied”

V) Adjust chart’s height and width to fit the entire screen.

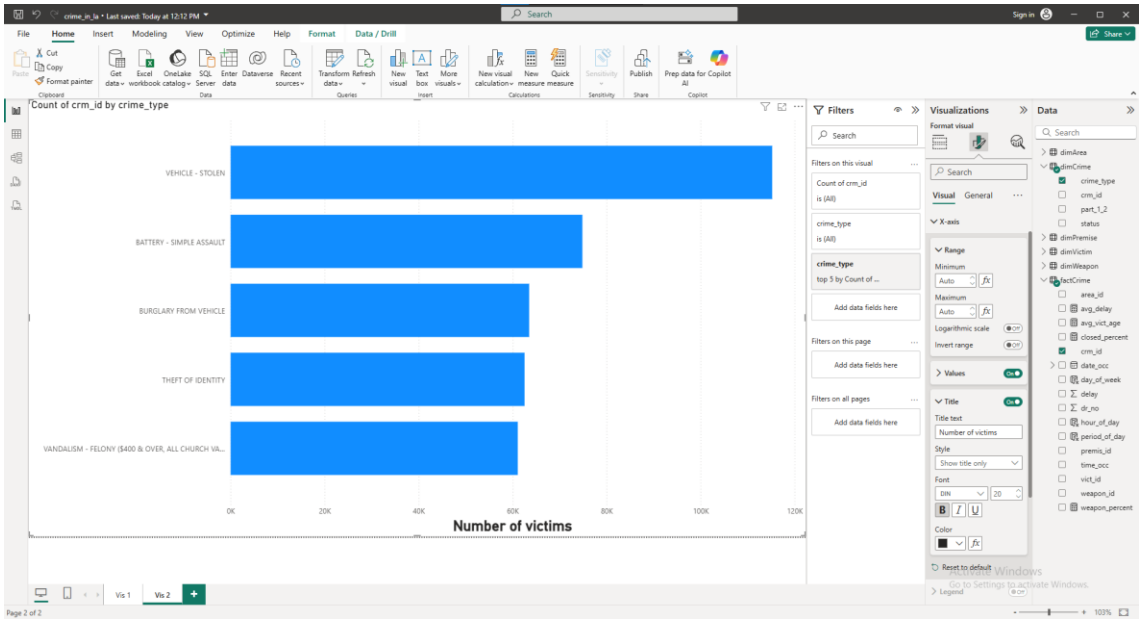
VI) Remove the redundant Y-axis title to make our chart more appealing by heading to “Format your visual”, expanding “Y-axis” that exists under “Visual” menu and turning “Title” from “On” to “Off”.



Screenshot 99: “Y-axis title removed”

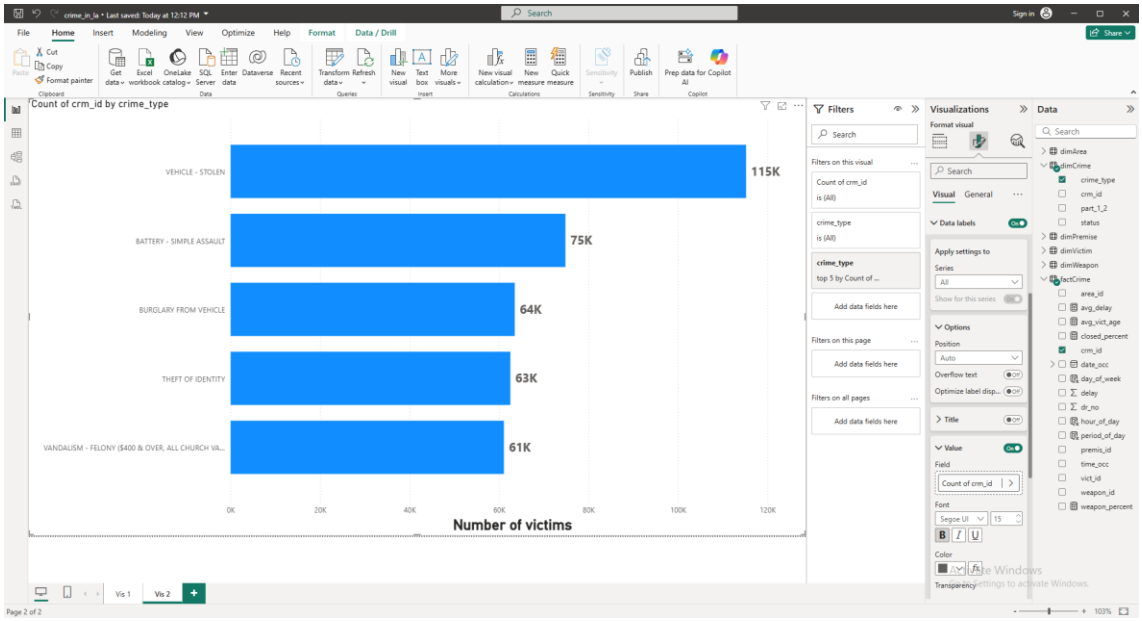
VII) Update the X-axis title by expanding “X-axis”, expanding “Title” and

changing “Title text” from “Auto” to “Number of victims”. Make it bold and larger with a font size of 20.



Screenshot 100: “X-axis title changed”

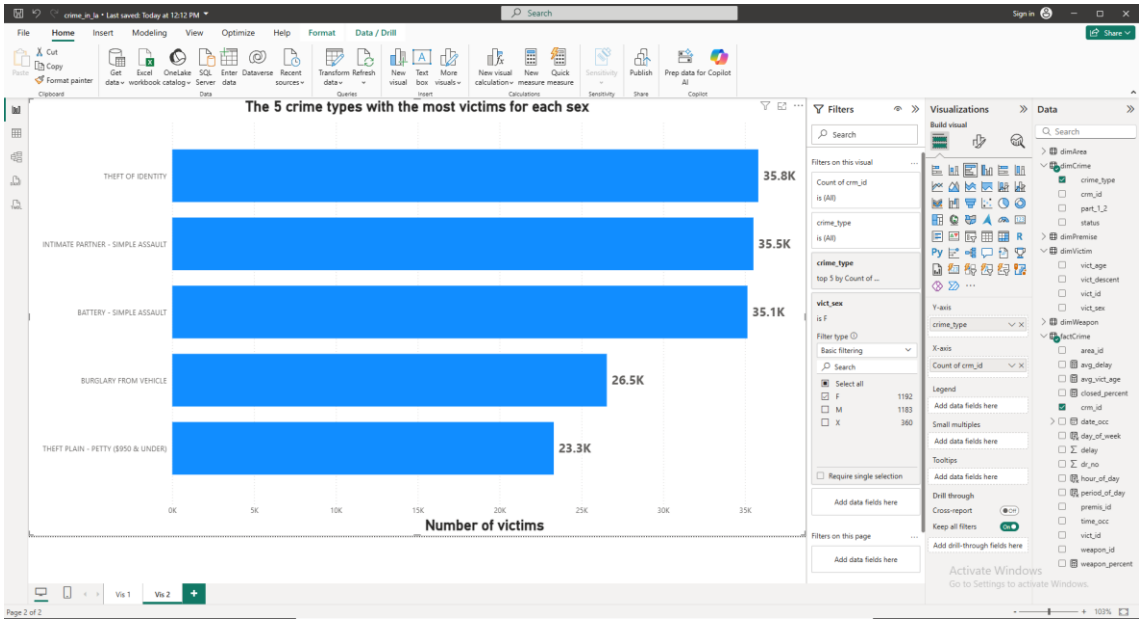
VIII) Switch “Data labels” from “Off” to “On” to display the number of victims directly on chart’s bars to provide ourselves with immediate information about all the data points without needing to hover over them. Next, make data labels’ values larger by giving them a font number of 15 and turn them into bold.



Screenshot 101: “Data labels activated and formatted”

IX) Change the chart's title to "The 5 crime types with the most victims for each sex" move the new title to the top-middle of the canvas. Moreover, increase its font size to 20 and make it bold.

X) Finally, drag `vict_sex` to "Filters on this visual" and switch between "F" and "M" to project the top ten crime types for each sex.



Screenshot 102: "Victim sex filter added"

70.

I) Create a new page.

II) Choose "Donut chart".

III) Drag `factCrime`'s `crm_id` to "Values" and `part_1_2` to "Legend".

IV) Adjust chart's height and width accordingly.

V) Replace the color used for Part 2 (#12239E) with #AEB83D.

VI) Make chart's values distinguishable by setting their font number from 9 to 15 and by turning them into bold. Then, reduce the decimal places of values and percentages.

VII) Change the chart's title to "Part 1 vs Part 2 crimes". Make the title larger by setting font number to 20, make it bold and move it to the top-middle of the canvas.

71.

I) Create a new page.

II) Choose “Clustered bar chart”.

III) Drag `factCrime`’s `crm_id` to “X-axis”, `crime_type` to “Y-axis”.

IV) Drag `crime_type` to “Filters on this visual”. Set its “Filter type” to “Top N” with the number 5. Then, drag `factCrime`’s `crm_id` to “By value” and apply the filter.

V) Adjust chart’s height and width accordingly.

VI) Remove the Y-axis title.

VII) Change the X-axis title from “Auto” to “Number of incidents”. Make it bold with a font size of 20.

VIII) Switch “Data labels” from “Off” to “On” and assign to data labels’ values a font number of 15 and turn them into bold.

IX) Lastly, change the chart’s title to “The 5 crime types with the most incidents” and move the new title to the top-middle of the canvas. Moreover, increment its font size to 20 and make it bold.

72.

I) Create a new page.

II) Choose “Clustered bar chart”.

III) Drag `factCrime`’s `crm_id` to “X-axis”, `period_of_day` to “Y-axis”.

IV) Adjust chart’s height and width accordingly.

V) Remove the Y-axis title.

VI) Rename the X-axis title from “Auto” to “Number of incidents”. Make it bold with a font size of 20.

VII) Switch “Data labels” from “Off” to “On” and assign to data labels’ values a font number of 15 and turn them into bold.

VIII) Lastly, change chart’s title to “In which day period do most crimes take place?” and move the new title to the top-middle of the canvas. Moreover, increment its font size to 20 and make it bold.

73.

I) Create a new page.

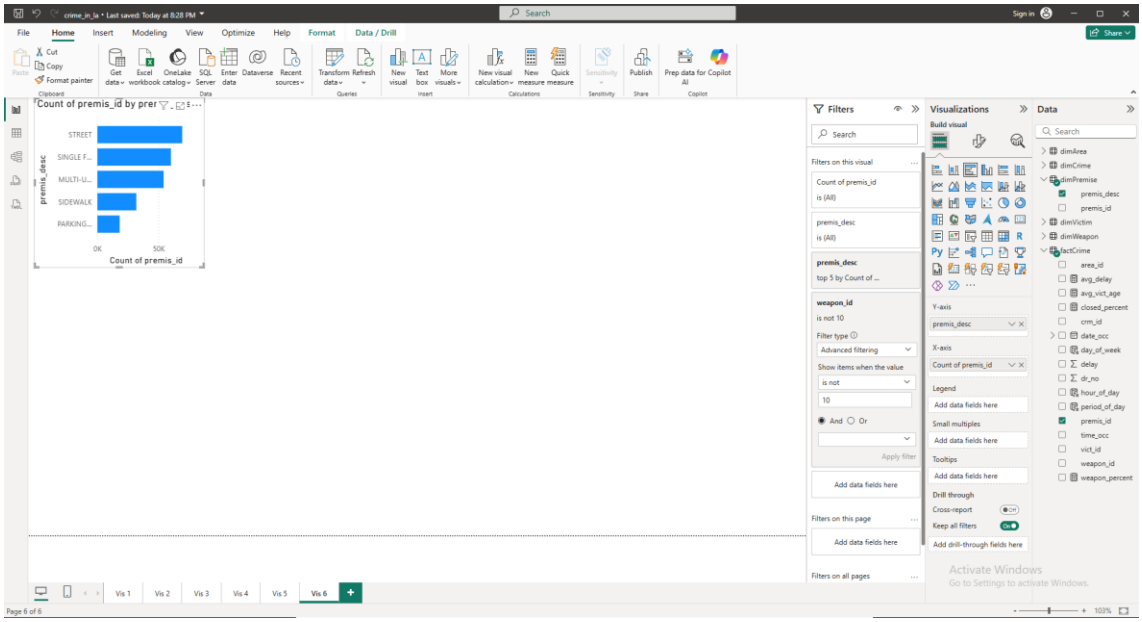
II) Choose “Clustered bar chart”.

III) Drag `factCrime`’s `premis_id` to “X-axis”, `premis_desc` to “Y-axis”.

IV) Drag `premis_desc` to “Filters on this visual”. Set its “Filter type” to “Top N”

with the number 5. Then, drag `factCrime`'s `premis_id` to "By value" and apply the filter.

V) Drag `weapon_id` to "Filters on this visual" and set its "Filter type" to "Advanced filtering". Under the "Show items when the value" section, choose "is not", enter the value 10 and apply the filter.



Screenshot 103: "Advanced filtering applied"

VI) Adjust chart's height and width accordingly.

VII) Remove the Y-axis title.

VIII) Change the X-axis title from "Auto" to "Number of armed incidents". Make it bold with a font size of 20.

IX) Switch "Data labels" from "Off" to "On" and give data labels' values a font size 15 and turn them into bold.

X) Lastly, change the chart's title to "The 5 premise types with the most armed incidents" and move the new title to the top-middle of the canvas. Increase its font size to 20 and make it bold.

74.

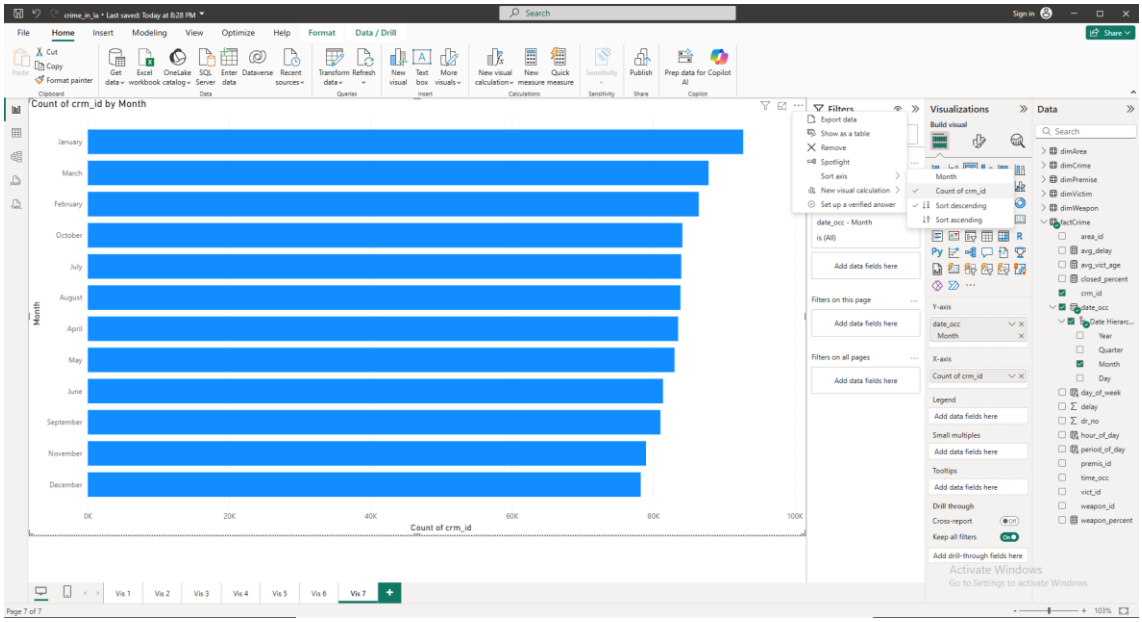
I) Create a new page.

II) Select "Clustered bar chart".

III) In the data pane, expand `date_occ`, "Date Hierarchy", drag `Month` to "Y-axis", `crm_id` to "X-axis".

IV) Adjust chart's height and width accordingly.

V) To sort the data by the incident count in descending order, click the three-dot menu in the top-right corner of the chart, expand "Sort axis" and select "Count of crm_id".



Screenshot 104: "Data sorted by the incident count"

VI) Remove the Y-axis title.

VII) Update the X-axis title to "Number of incidents". Make it bold with a font size of 20.

VIII) Switch "Data labels" from "Off" to "On", set 15 as the font size of data labels' values and turn them into bold.

IX) Lastly, change the chart's title to "Number of incidents by month" and move the new title to the top-middle of the canvas. Use a 20-point font and make it bold.

75.

I) Create a new page.

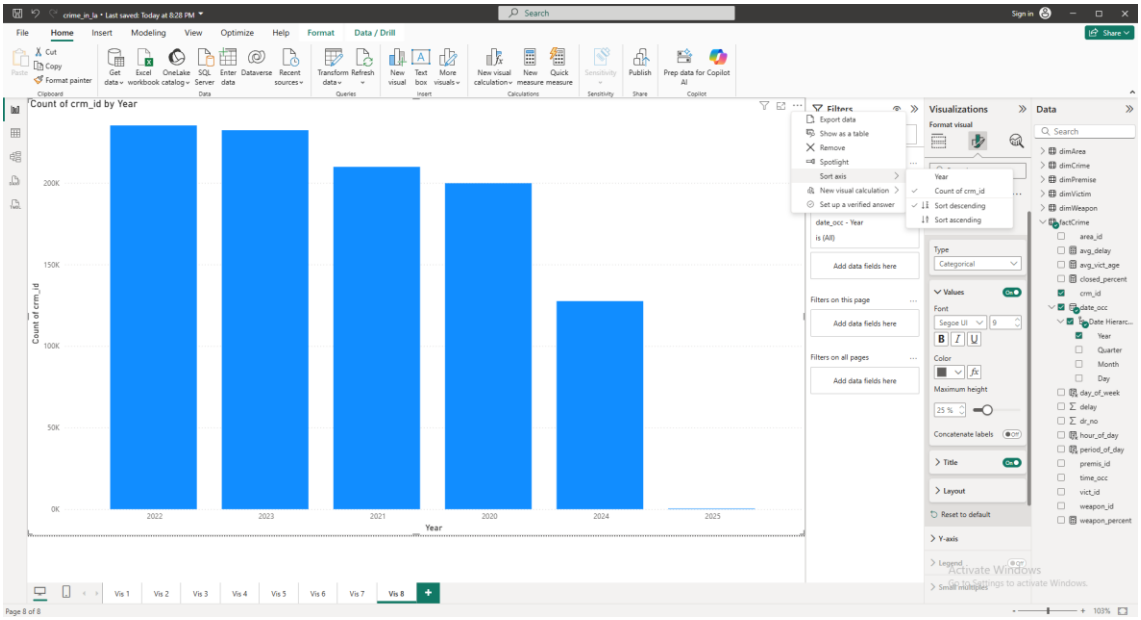
II) Select "Clustered column chart".

III) Drag `Year` to "X-axis", `crm_id` to "Y-axis".

IV) Adjust chart's height and width accordingly.

V) In order to acquire the ability to sort the chart from the year with the highest number of incidents to the year with the lowest number of incidents, proceed to

changing X-axis's type by heading to "Format your visual", expanding "X-axis" and switching the type from "Continuous" to "Categorical" (since the field of month is numerical, Power BI considers it already sorted and doesn't display any sorting by another field option). Then, click the three-dot menu in the top-right corner of the chart, expand "Sort axis", and select both "Count of crm_id" and "Sort descending" options.



Screenshot 105: "Year's type changed to categorical"

VI) Remove the X-axis title.

VII) Update the Y-axis title to "Number of incidents". Make it bold with a font size of 20.

VIII) Switch "Data labels" from "Off" to "On", set 15 as the font size of data labels' values and turn them into bold.

IX) Lastly, change the chart's title to "Number of incidents by year" and move the new title to the top-middle of the canvas. Use a 20-point font and make it bold.

76.

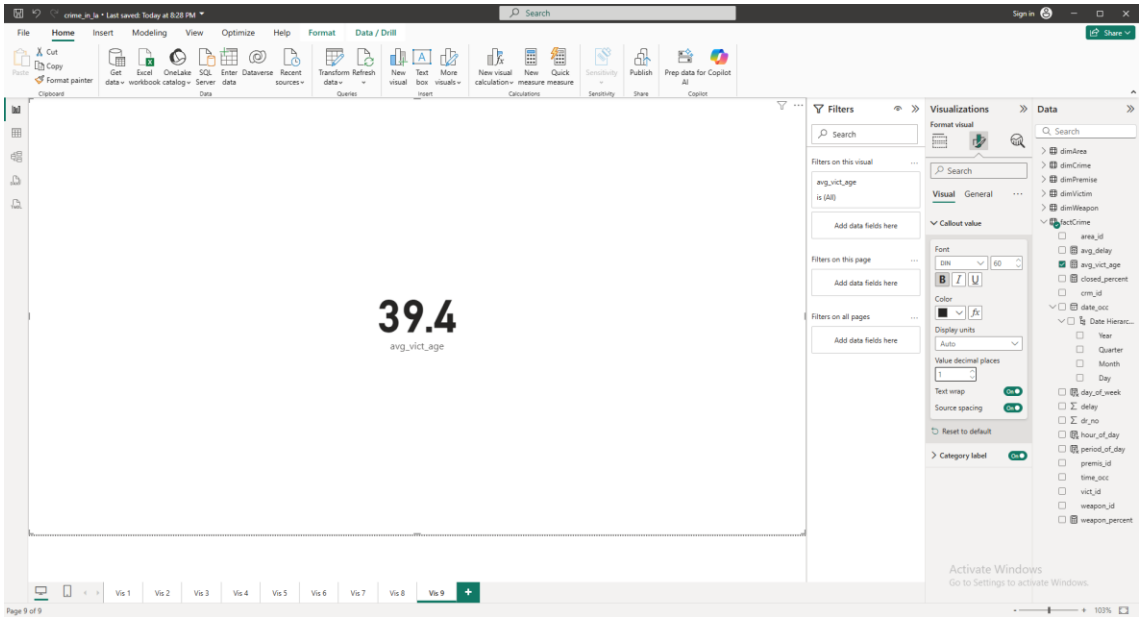
I) Create a new page.

II) Select the "Card" chart.

III) Drag `avg_vict_age` to "Fields", which exists in the "Visualizations" pane.

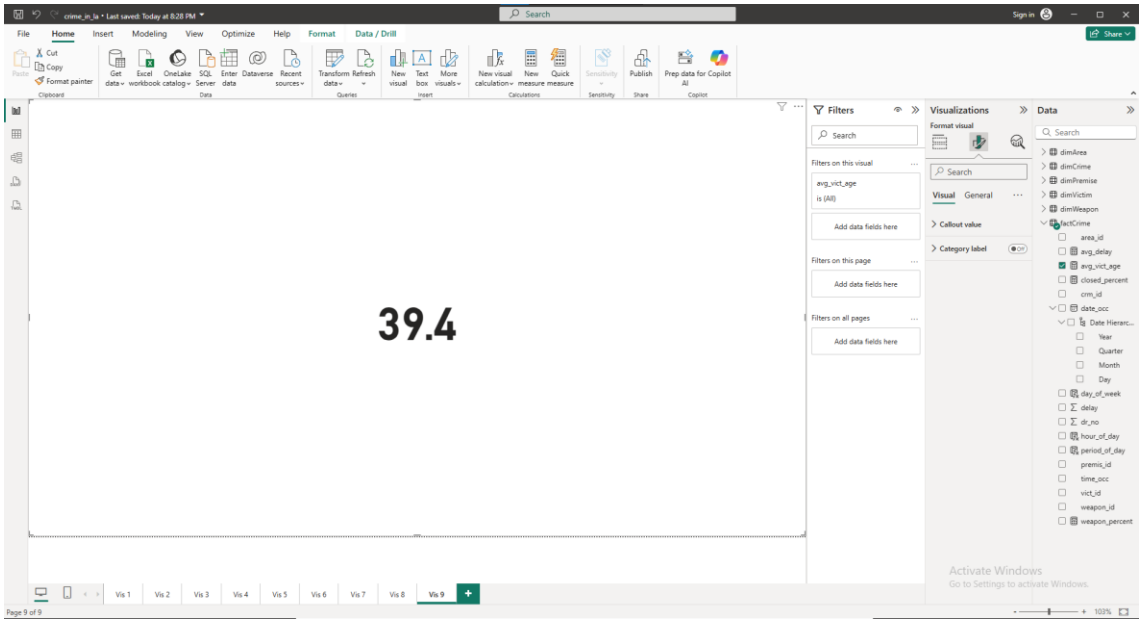
IV) Adjust chart's height and width accordingly.

V) Make the chart's value distinct by clicking on "Format your visual", expanding "Callout value", setting a font size of 60 and making it bold. Furthermore, make it look smoother by reducing its decimal places from "Auto" (equals by default to 2) to 1.



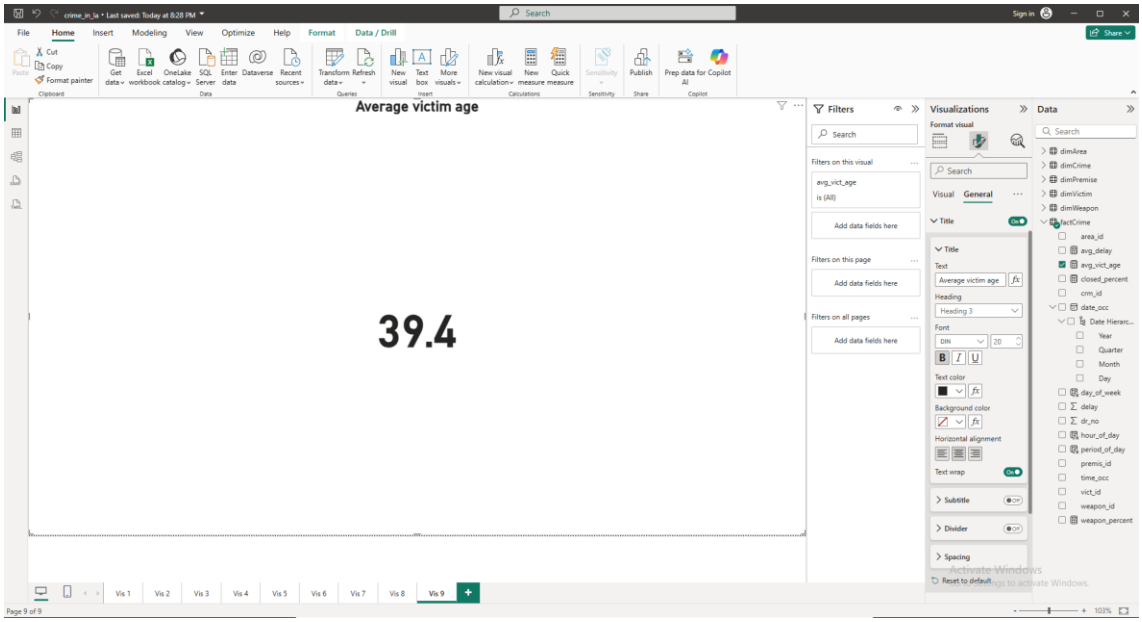
Screenshot 106: "Chart's value formatted"

VI) Turn off "Category label" because a title will be added to the chart in the next step.



Screenshot 107: "Category label deactivated"

VII) To add a title, click on the “General” tab, which exists under “Format your visual” menu, turn on “Title”, expand “Title” and write the title’s text into the input box e.g., “Average victim age”. As we did with the rest of our charts’ titles, make it bold with a font size of 20 and move it to the center of the page.



Screenshot 108: “Title added”

77.

I) Create a new page.

II) Select the “Card” visual.

III) Drag `weapon_percent` to “Fields”.

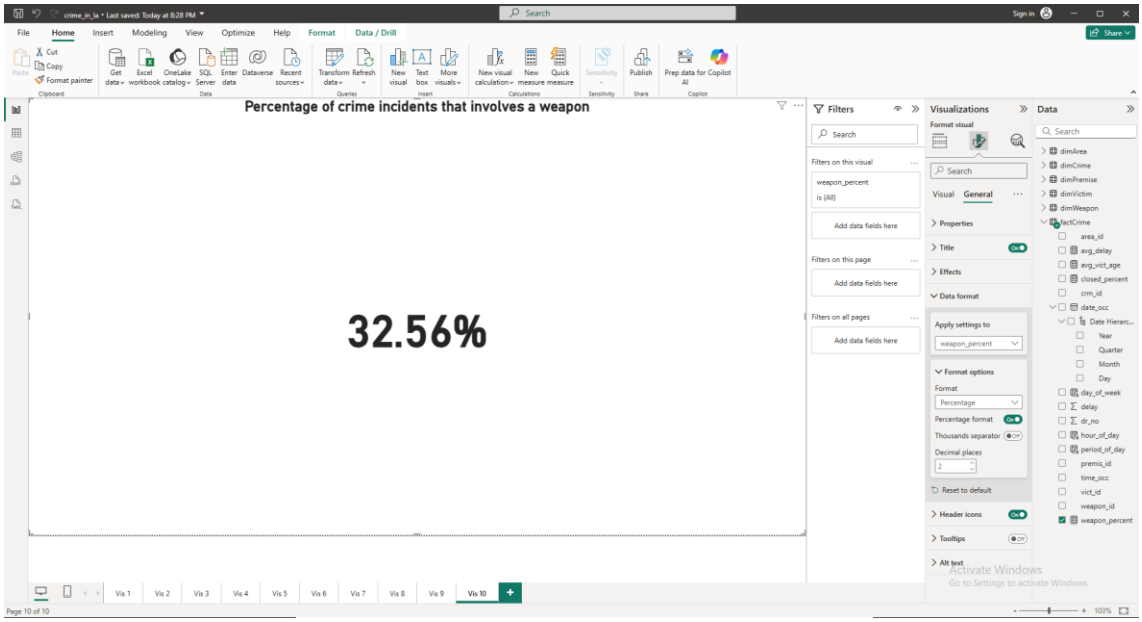
IV) Adjust chart’s height and width accordingly.

V) Make the chart’s value bold with a font size of 60.

VI) Turn off “Category label”.

VII) Add the title “Percentage of crime incidents that involves a weapon”. Make the title bold with a font size of 20 and move it to the top-middle of the page.

VIII) Lastly, change the data format of `weapon_percent` by expanding “Data format” that exists under the “General” tab of the “Format your visual” menu, expanding “Format options” and switch the format from blank (decimal number) to “Percentage”. This will convert 0.33 into 32.56%.



Screenshot 109: “Data format from decimal to percentage”

78.

I) Create a new page.

II) Select “Clustered column chart”.

III) Drag `vict_descent` to “X-axis”, `factCrime`’s `vict_id` to “Y-axis”.

IV) Drag `vict_descent` to “Filters on this visual”. Set its “Filter type” to “Top N” with the number 5. Then, drag `factCrime`’s `vict_id` to “By value” and apply the filter.

V) Adjust chart’s height and width accordingly.

VI) Rename X-axis title to “Victim descent” and make it bold with a font size of 20.

VII) Update the Y-axis title to “Number of incidents”. Make it bold with a font size of 20.

VIII) Switch “Data labels” from “Off” to “On”, set 15 as the font size of data labels’ values and turn them into bold.

IX) Lastly, change the chart’s title to “The 5 most victimized descents” and move the new title to the top-middle of the canvas. Use a 20-point font and make it bold.

79.

I) Create a new page.

II) Select “Clustered bar chart”.

III) Drag `area` to “Y-axis”, `factCrime`’s `area_id` to “X-axis”.

IV) Drag `area` to “Filters on this visual”. Set its “Filter type” to “Top N” with the number 5. Then, drag `factCrime`’s `area_id` to “By value” and apply the filter.

V) Adjust chart’s height and width accordingly.

VI) Remove the Y-axis title.

VII) Rename the X-axis title from “Auto” to “Number of incidents”. Make it bold with a font size of 20.

VIII) Enable “Data labels” and set their font to bold with a size of 15.

IX) Lastly, update the chart’s title to “The 5 areas with the most crimes” and move the new title to the top-middle of the canvas. Also, apply bold styling with font size 20.

80.

I) Create a new page.

II) Choose the “Clustered bar chart”.

III) Drag `crime_type` to “Y-axis” and `avg_vict_age` to “X-axis”.

IV) Adjust chart’s height and width accordingly.

V) Remove the Y-axis title.

VI) Rename the X-axis title to “Average age”. Make it bold and larger with a font size of 20.

VII) Enable “Data labels”, make their values larger by giving them a font size of 15 points and turn them into bold.

VIII) Change the chart’s title to “The 5 crime types with the highest/lowest average victim age” and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

IX) To view the five crime types with the highest average age, drag `crime_type` to “Filters on this visual”, set its filter type to “Top N” with the number 5, drag `avg_vict_age` to “By value” and apply the filter.

X) To view the five crime types with the lowest average age, select “Bottom” instead of “Top” under the “Show items” menu and apply the filter. To order the results in ascending order, click on the three dots that exist at the top-right corner of the chart, expand the “Sort axis” option and change the default setting from “Sort descending” to “Sort ascending”.

81.

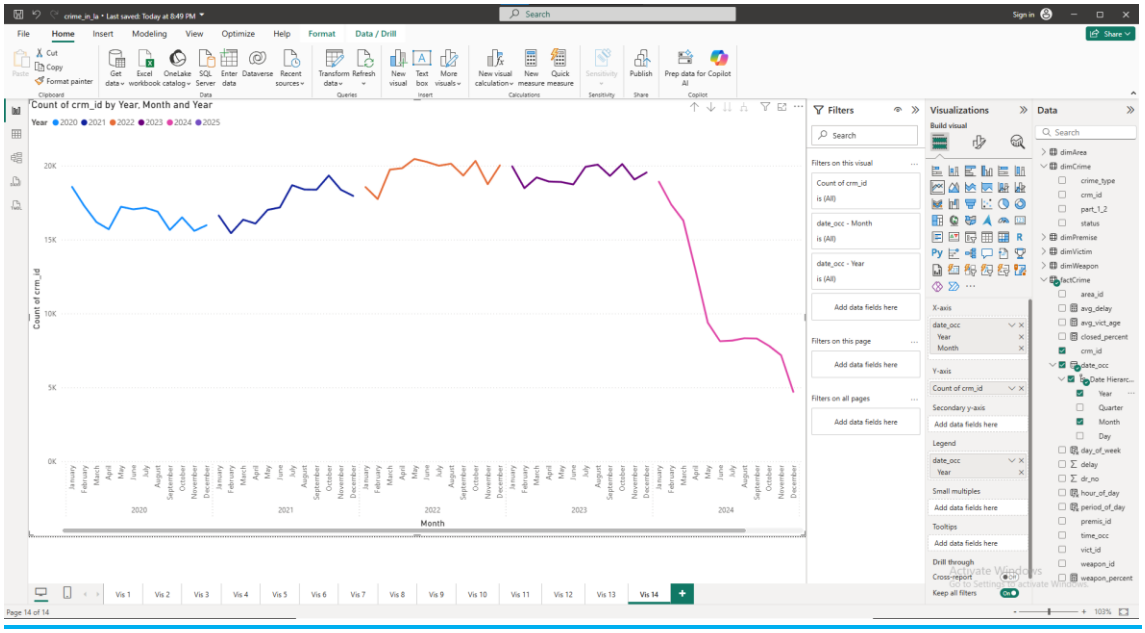
I) Create a new page.

II) Choose the “Line chart”.

III) Drag `year` and `month` to “X-axis”, `crm_id` to “Y-axis”.

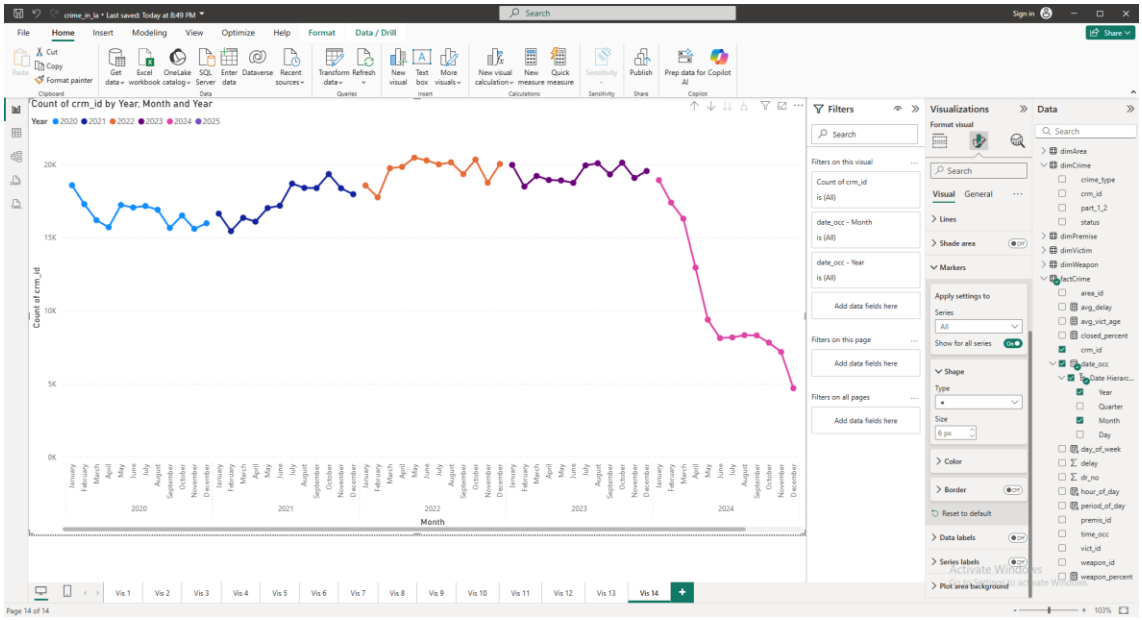
IV) Adjust chart’s height and width accordingly.

V) Drag `year` to “Legend” to differentiate each year’s data series with a unique color.



Screenshot 110: “Each year with its own color”

VI) Enable markers for months by expanding “Markers” that is located at the “Visual” tab of the “Format your visual” menu and turning on the “Show for all series” option. Set their size to 6 px to make them more visible.

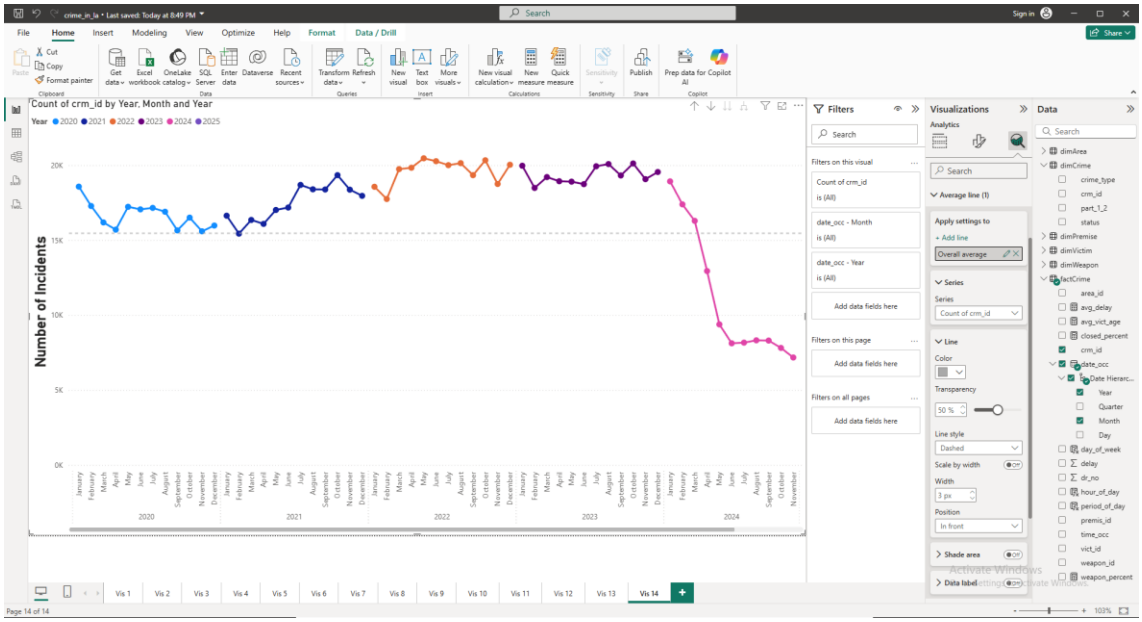


Screenshot 111: “Markers for months enabled”

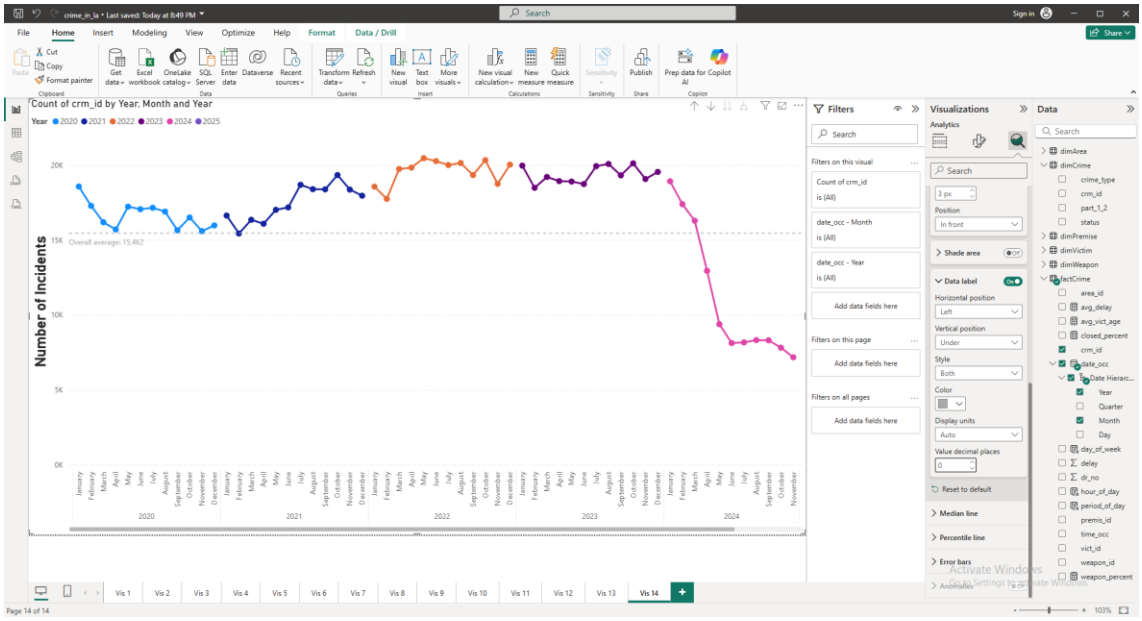
VII) Update the Y-axis title to “Number of Incidents”, set the font to bold and use size 20.

VIII) Remove the X-axis title.

IX) To add an average line, expand “Average line” section found under the “Add further analyses to your visual” menu and click the button “+ Add line”. When the line is added, rename it to “Overall average” by clicking on the pencil icon next to it. Then, expand the “Line” section and choose the color “#A9A9A9”. To show the average line’s value, enable the “Data label” section. Additionally, expand “Data label” to update its color to “#A9A9A9”. In the same section, change its style from “Data value” to “Both” to ensure that the line’s title is added on the left side of its value, choose “Under” as its “Vertical position” setting, and decrease value decimal places to 0.



Screenshot 112.1: “Gray average line added”



Screenshot 112.2: “Average line’s title and value added in gray”

X) Change the chart’s title to “Number of crimes per year-month combination” and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

82.

I) Create a new page.

II) Choose the “Line chart”.

III) Drag `hour_of_day` to “X-axis” and `crm_id` to “Y-axis”.

IV) Adjust chart’s height and width accordingly.

V) Change Y-axis title to “Number of Incidents”. Make it bold and larger with a font size of 20.

VI) Update the X-axis title to “Time of occurrence”, set the font to bold and use size 20.

VII) Add an average line with the name “Overall average” and the color “#FF0000”. Enable its data label and set its color to “#FF0000”. Also, set its style to “Both”, choose “Under” as its vertical position, and decrease its decimal places to 0.

VIII) Rename the chart title to “Number of crimes per time of occurrence” and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of “20”.

IX) To order the results by the hour of day in ascending order, click on the three dots at the top-right corner of the chart and select both “hour_of_day” and “Sort ascending” options.

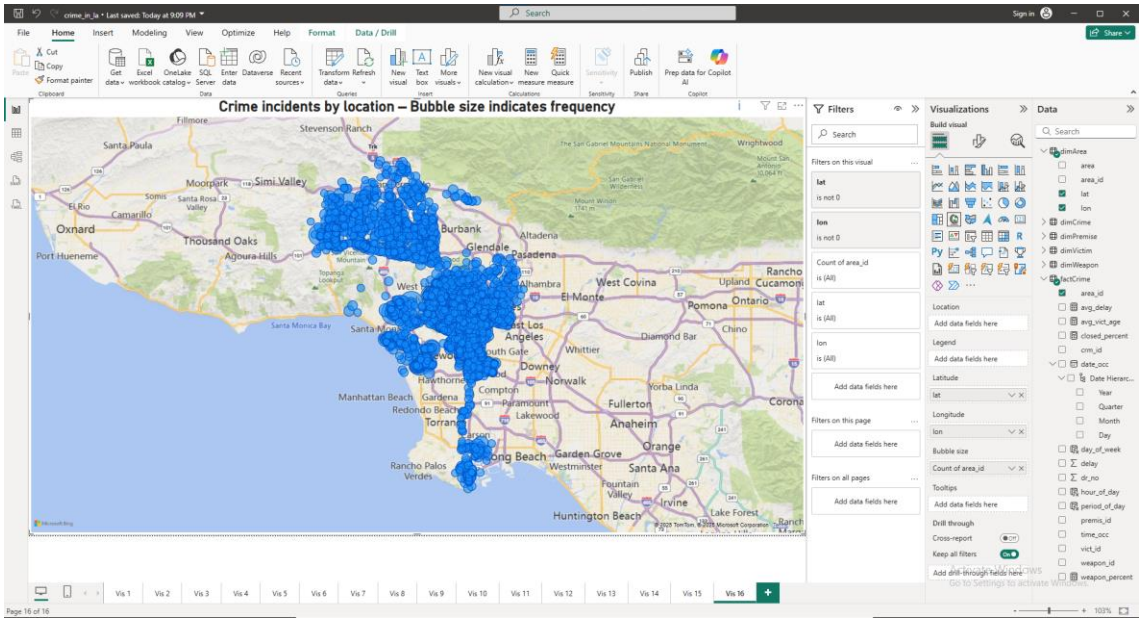
83.

I) Create a new page.

II) Click on the “Map” chart.

III) Drag `lat` to “Latitude”, `lon` to “Longitude”, and `factCrime`’s `area_id` to “Bubble size”.

IV) Drag `lat` to “Filters on this visual”, set its “Filter type” to “Advanced filtering”, Under the “Show items when the value” section, choose “is not”, enter the value 0 and apply the filter. Then repeat the exact same procedure for `lon`, excluding records where the value is 0. This step ensures that incidents with unknown coordinates (i.e., `lat` = 0, `lon` = 0) are not displayed on the map.

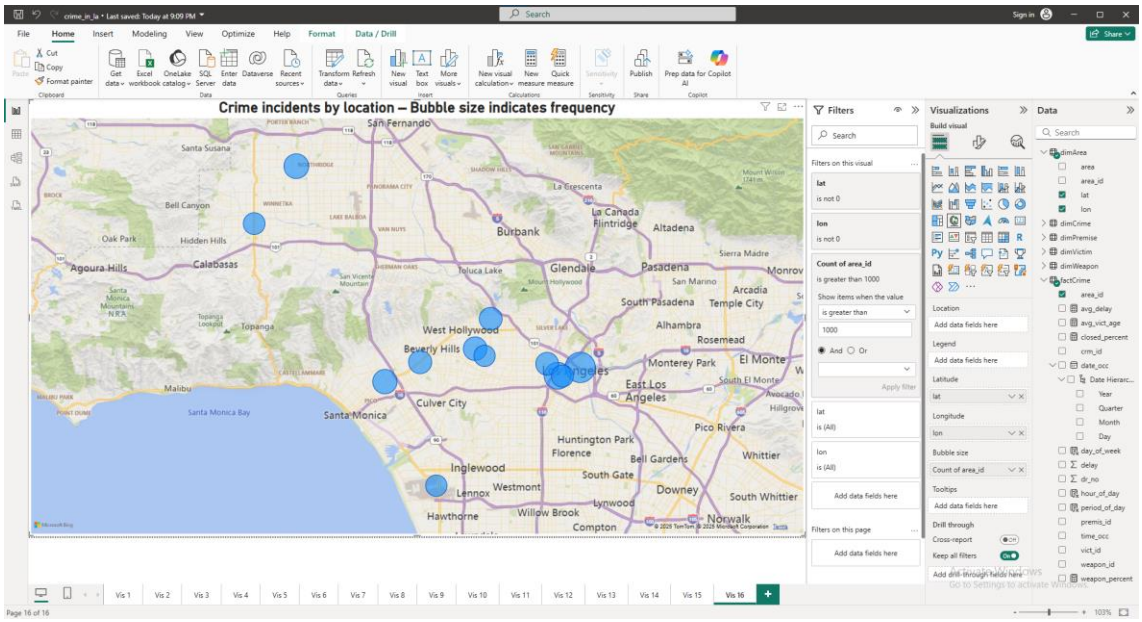


Screenshot 113.1: "Lat & lon filter added"

V) Adjust chart's height and width accordingly.

VI) Update the chart title to "Crime incidents by location — Bubble size indicates frequency" and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

VII) Finally, to display only the locations with more than 1,000 incidents, expand "Count of area_id is (All)", which can be found within the "Filters on this visual" menu, select "is greater than", enter the value 1000 in the "Show items when the value" field and apply the filter to update the visual accordingly.



Screenshot 113.2: "> 1000 incidents filter added"

84.

I) Create a new page.

II) Select “Clustered bar chart”.

III) Drag `crime_type` to “Y-axis”, `avg_delay` to “X-axis”.

IV) Adjust chart’s height and width accordingly.

V) Remove the Y-axis title.

VI) Change the X-axis title to “Average delay”. Make it bold and larger with a font size of 20.

VII) Enable “Data labels”, make their values larger by giving them a font size of 15 points and turn them into bold.

VIII) Change the chart’s title to “Average reporting delay (in days) by crime type: Top 5 vs Bottom 5” and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

IX) To view the five crime types with the highest average delay, drag `crime_type` to “Filters on this visual”, set its filter type to “Top N” with the number 5, drag `average_delay` to “By value” and apply the filter.

X) To view all the crime types with the lowest average delay, select “Bottom” instead of “Top” and apply the filter. To order the results in ascending order, click on the three dots at the top-right corner of the chart and choose “Sort ascending”.

85.

I) Create a new page.

II) Select “Clustered column chart”.

III) Drag `area` to “X-axis”, `avg_delay` to “Y-axis”.

IV) Adjust chart’s height and width accordingly.

V) Remove the X-axis title.

VI) Update the Y-axis title to “Average delay”. Make it bold and larger with a font size of 20.

VII) Enable “Data labels”, set their values’ font size to 15 and turn them into bold.

VIII) Change the chart’s title to “Average reporting delay (in days) by area: Top 5 vs Bottom 5” and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

IX) To view the five areas with the highest average delay, drag `area` to “Filters on this visual”, set its filter type to “Top N” with the number 5, drag `avg_delay` to “By value” and apply the filter.
X) To view the five areas with the lowest average delay, select “Bottom” instead of “Top” and apply the filter. To order the results in ascending order, click on the three dots at the top-right corner of the chart and choose “Sort ascending”.

86.

I) Create a new page.
II) Select the “Card” visual.
III) Drag `avg_delay` to “Fields”.
IV) Adjust chart’s height and width accordingly.
V) Make the chart’s value bold with a font size of 60 and reduce its decimal places from 2 to 1.
VI) Turn off “Category label”.
VII) Add the title “Average reporting delay (in days)”. Make the title bold with a font size of 20 and move it to the top-middle of the page.

87.

I) Create a new page.
II) Select “Clustered column chart”.
III) Drag `day_of_week` to “X-axis”, `crm_id` to “Y-axis”.
IV) Adjust chart’s height and width accordingly.
V) Remove the X-axis title.
VI) Rename the Y-axis title to “Number of incidents”. Set its font size to 20 and turn it into bold.
VII) Enable “Data labels”, make their values bold and set their font size to 15.
VIII) Update the chart’s title to “Number of incidents per day of the week”. Set its font size to 20, make it bold and move it at the top-middle of the chart.

88.

I) Create a new page.
II) Select “Clustered column chart”.
III) Drag `area` to “X-axis”, `closed_percent` to “Y-axis”.

IV) Adjust chart's height and width accordingly.

V) Remove the X-axis title.

VI) Remove the Y-axis title.

VII) Enable "Data labels", set their values' font size to 15 and turn them into bold.

VIII) Change the chart's title to "Percent of closed cases by area: Top 5 vs Bottom 5" and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

IX) To view the five areas with the highest percentage of closed cases, drag `area` to "Filters on this visual", set its filter type to "Top N" with the number 5, drag `closed_percent` to "By value" and apply the filter.

X) To view the five areas with the lowest percentage of closed cases, select "Bottom" instead of "Top" and apply the filter. To order the results in ascending order, click on the three dots at the top-right corner of the chart and choose "Sort ascending".

89.

I) Create a new page.

II) Select "Clustered bar chart".

III) Drag `crime_type` to "Y-axis", `closed_percent` to "X-axis".

IV) Adjust chart's height and width accordingly.

V) Remove the Y-axis title.

VI) Remove the X-axis title.

VII) Enable "Data labels", make their values larger by giving them a font size of 15 points and turn them into bold.

VIII) Change the chart's title to "Percentage of closed cases by crime type: Top 5 vs Bottom 5" and move the new title to the top-middle of the canvas. Furthermore, format it in bold with a font size of 20.

IX) To view the five crime types with the highest percentage of closed cases, drag `crime_type` to "Filters on this visual", set its filter type to "Top N" with the number 5, drag `closed_percent` to "By value" and apply the filter.

X) To view all the crime types with the lowest percentage of closed cases, select "Bottom" instead of "Top" and apply the filter. To order the results in ascending order, click on the three dots at the top-right corner of the chart and choose "Sort ascending".

90.

I) Create a new page.

II) Select the “Card” visual.

III) Drag `closed_percent` to “Fields”.

IV) Adjust chart’s height and width accordingly.

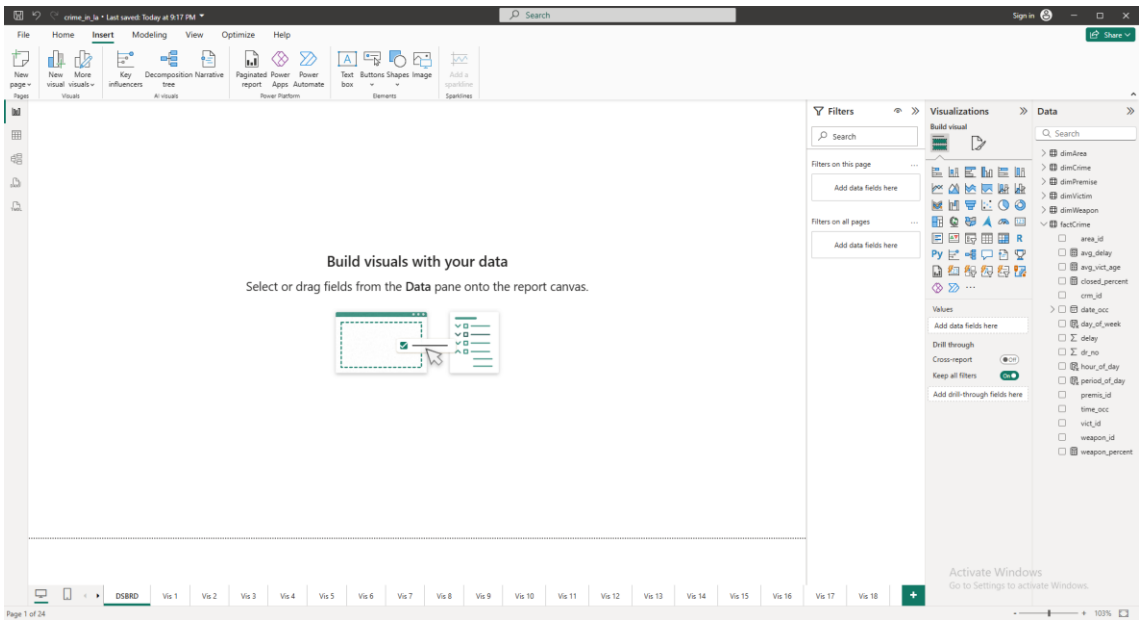
V) Make the chart’s value bold with a font size of 60.

VI) Turn off “Category label”.

VII) Add the title “Percentage of closed cases”. Make the title bold with a font size of 20 and move it to the top-middle of the page.

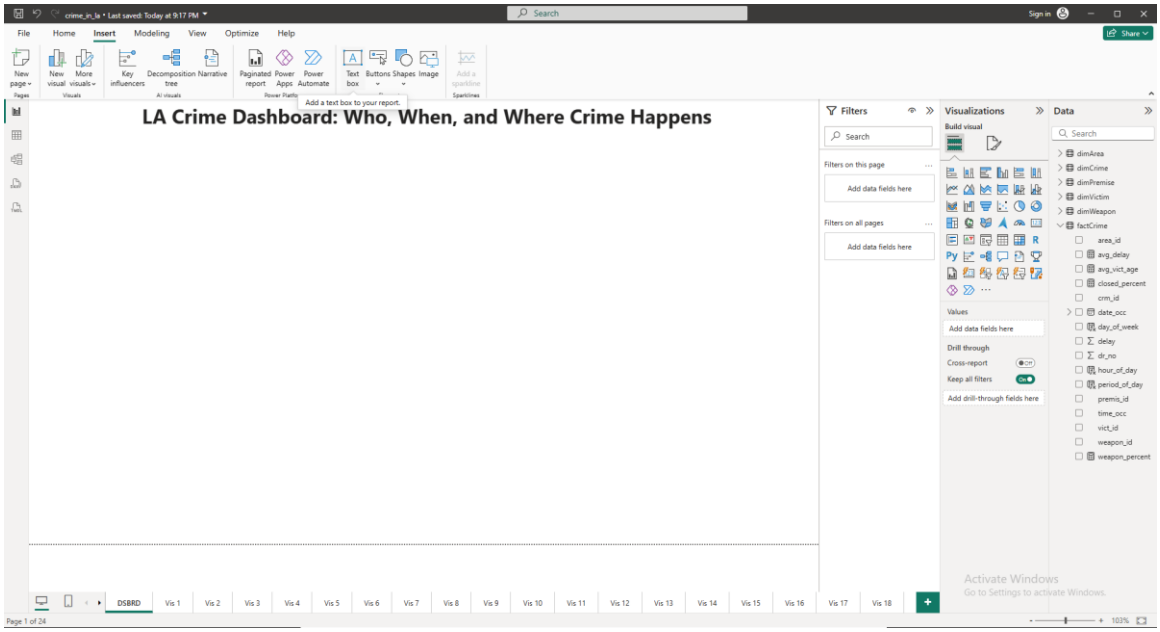
91.

I) Create a new page and move it to the front of the report by right-clicking on the page’s name, hovering over “Move to”, and selecting “Move to front”.



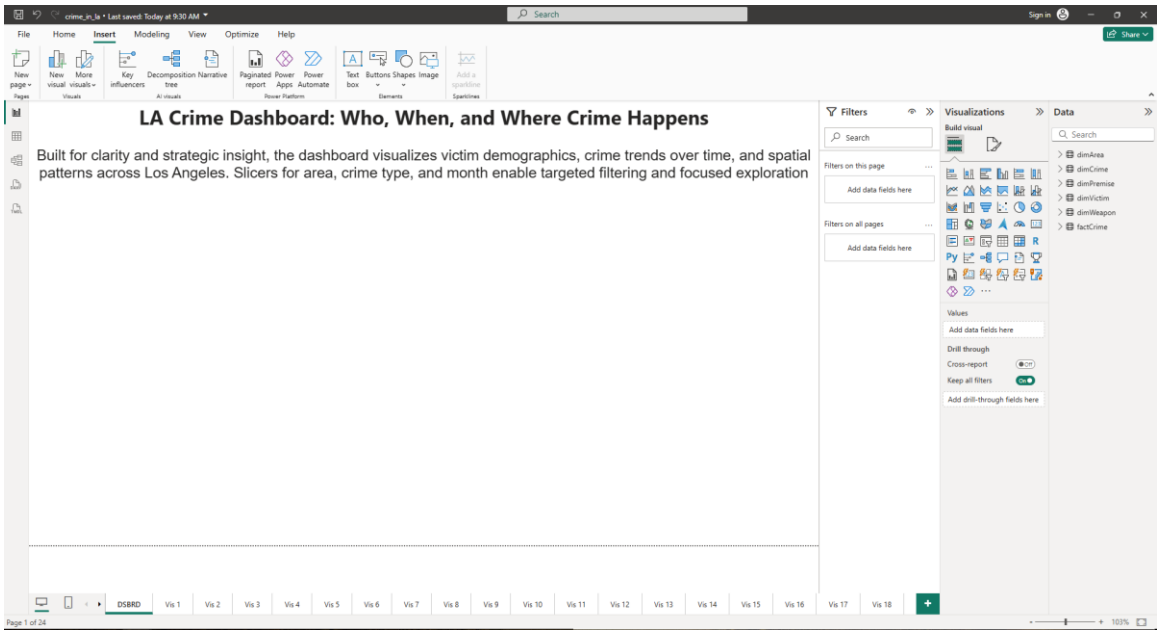
Screenshot 114: “Page moved to the front”

II) Give the dashboard a title by heading to the “Insert” tab, and above the “Elements” section, click “Text box”. Assign a title e.g.
“LA Crime Dashboard: Who, When, and Where Crime Happens”, make it bold with a font size of 24, and bring it to the center. Then, adjust the box’s height and width so it occupies the minimum amount of space. To apply the title, left-click anywhere on the page.



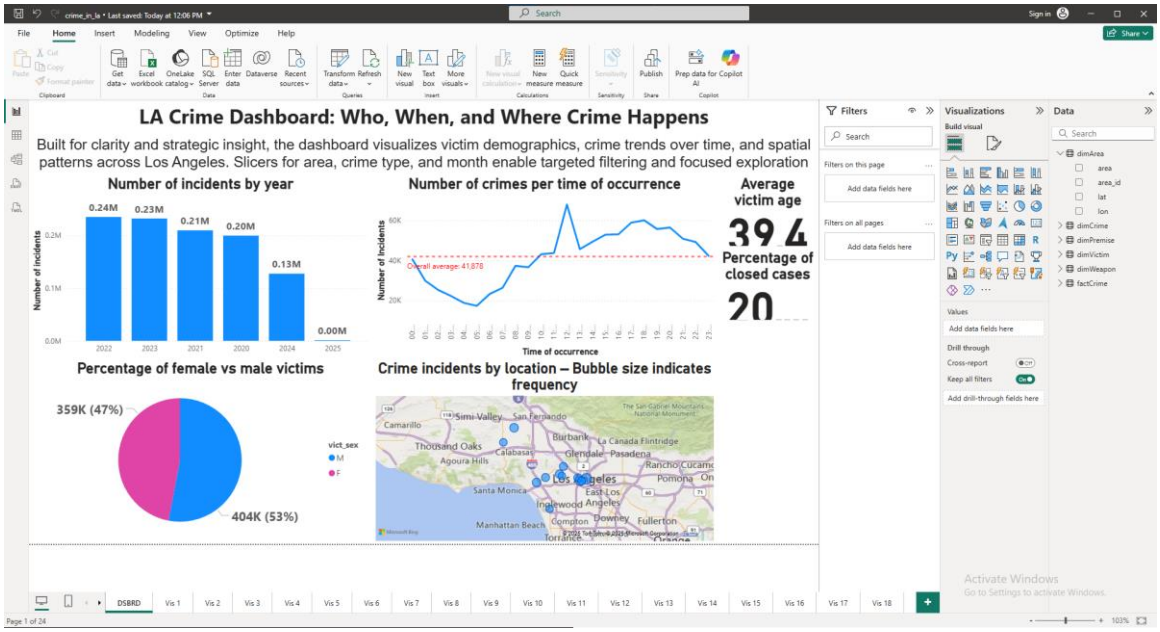
Screenshot 115: “Dashboard title added”

III) Give the dashboard a description by adding a second text box with a descriptive text e.g. “Built for clarity and strategic insight, the dashboard visualizes victim demographics, crime trends over time, and spatial patterns across Los Angeles. Slicers for area, crime type, and month enable targeted filtering and focused exploration”, make it bold with a font size of 18, and bring it to the center. Adjust the box’s height and width accordingly.



Screenshot 116: “Description added”

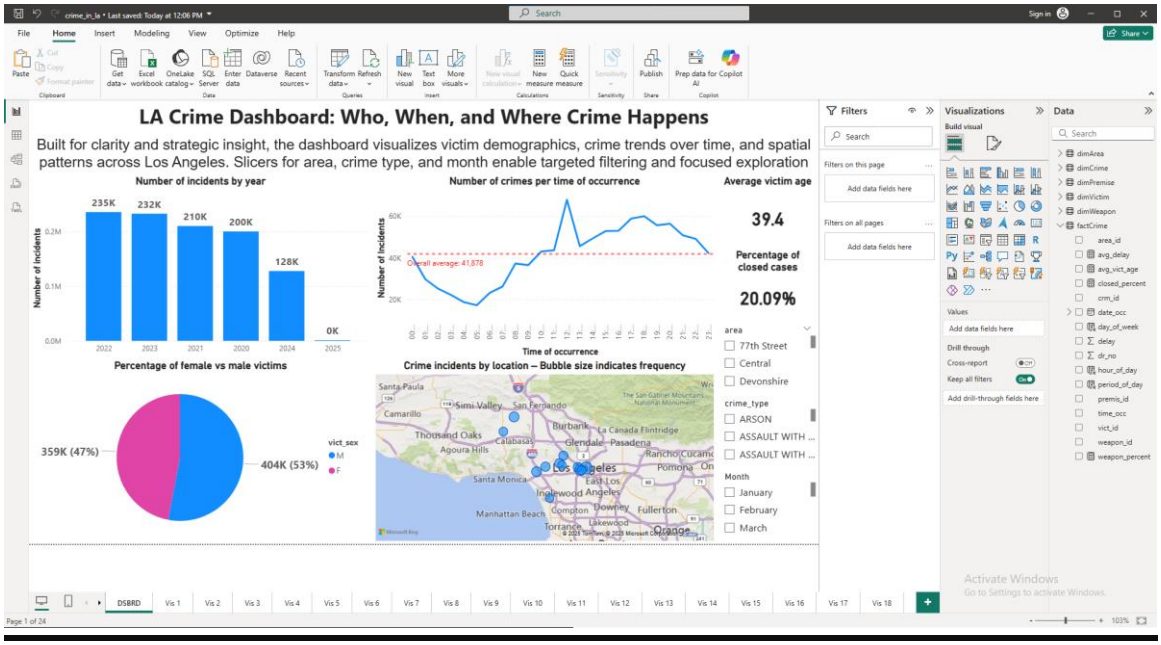
IV) Copy the first visual from its page — by left-clicking the visual, and pressing "CTRL + C" — and paste it — by clicking "CTRL + V" — to our newly created dashboard page. Then, head to "Format your visual", select the "General" tab, expand "Properties", expand "Size", and set "Height" as 298 and "Width" as 557. Then, drag the visual to a corner of the canvas to make space for other visuals. The same procedure has to be followed for the eighth, fifteenth, and sixteenth visual, except for the ninth and twenty-third visuals. Their height and width will be the same but smaller in size from the first four visuals — 119 and 165 points respectively.



Screenshot 117: "Visuals placed with height-width adjusted"

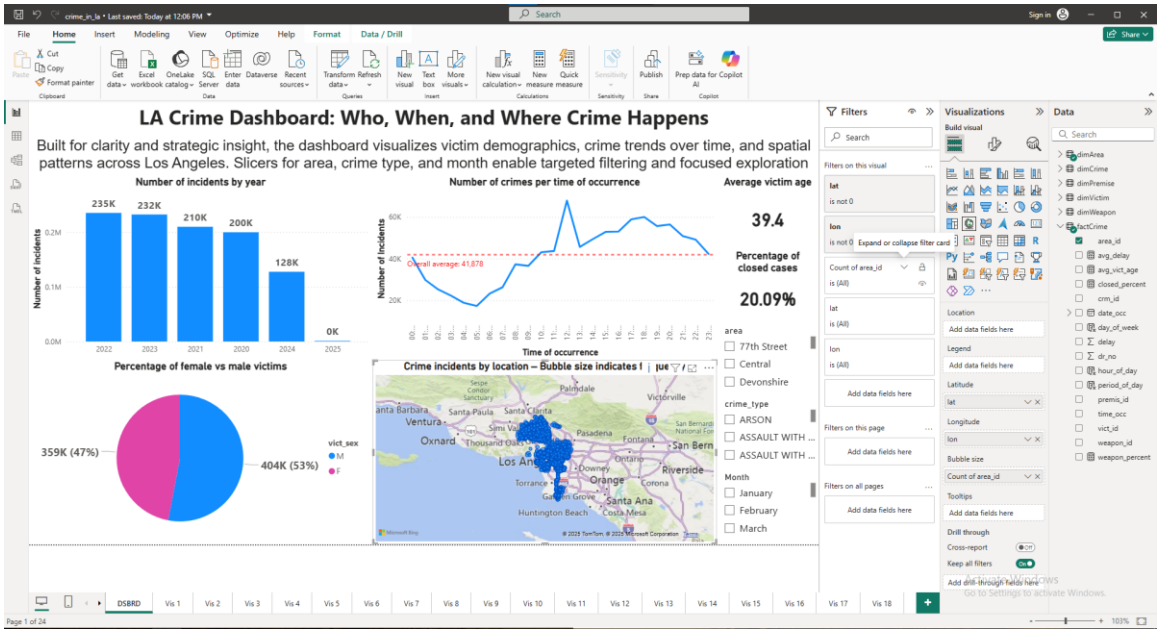
V) Decrease the title's font size of all visuals from 20 to 14 points, the callout value's size of the card visuals from 60 to 24, data labels values' size of the first and eighth visuals from 15 to 13. Change the display units of the eighth visual's data labels values from "Auto" to "Thousands" and set its decimal places to 0.

VI) Select the slicer chart and drag `area` to it. Set its height and width to 119 and 172 points respectively, and drag it to a corner. Then, create two more slicer charts — one for `crime_type` and the other for `month` — with the same height and width as the area slicer.



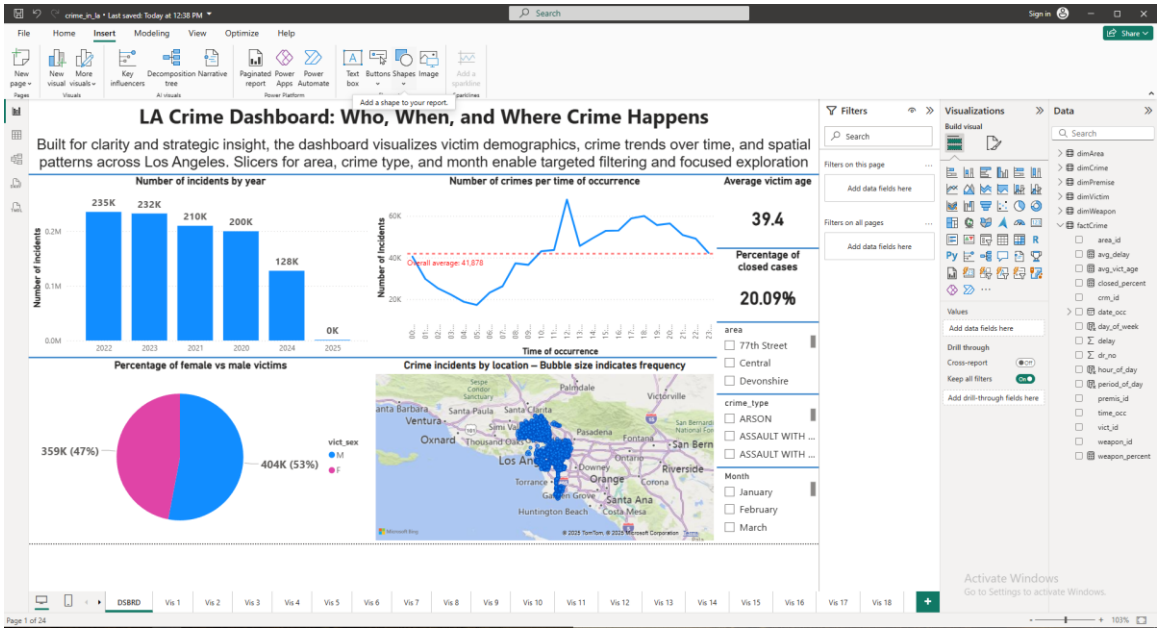
Screenshot 118: "Slicers added"

VII) Remove sixteenth visual's filter — showing >1000 incidents — to view all incident locations by hovering over the filter and clicking "Clear filter".



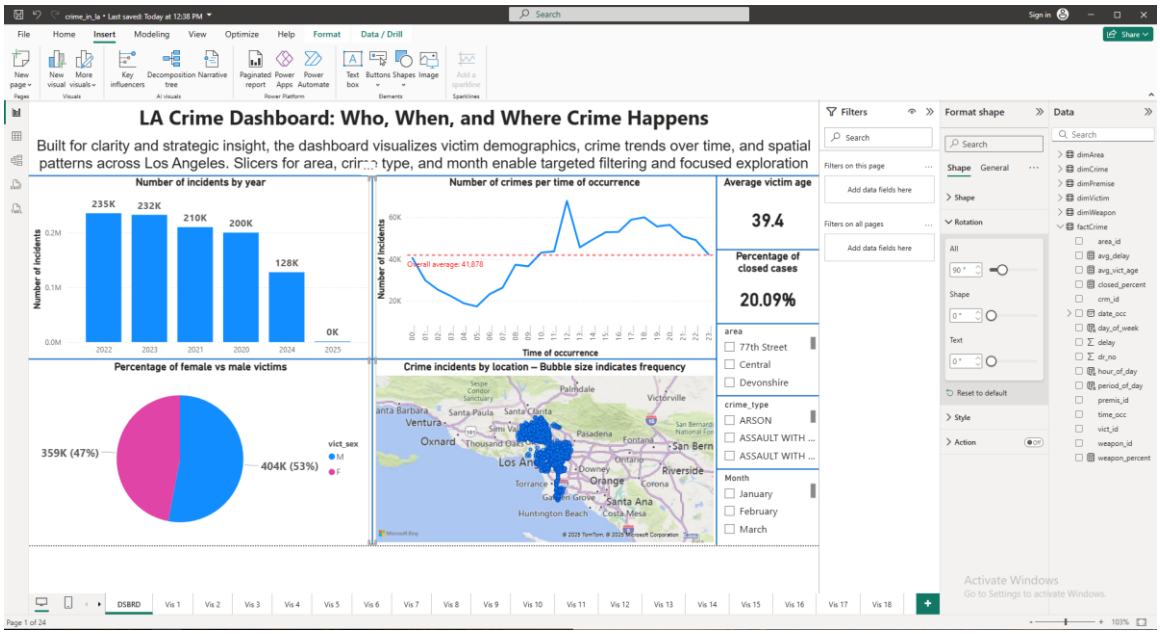
Screenshot 119: "Filter removed"

VIII) Add six horizontal lines to the dashboard by heading to the "Insert" tab, and above the "Elements" section, expand "Shapes" and select "Line". Adjust their height, width, and position to properly segregate the visuals.



Screenshot 120: “Horizontal lines added”

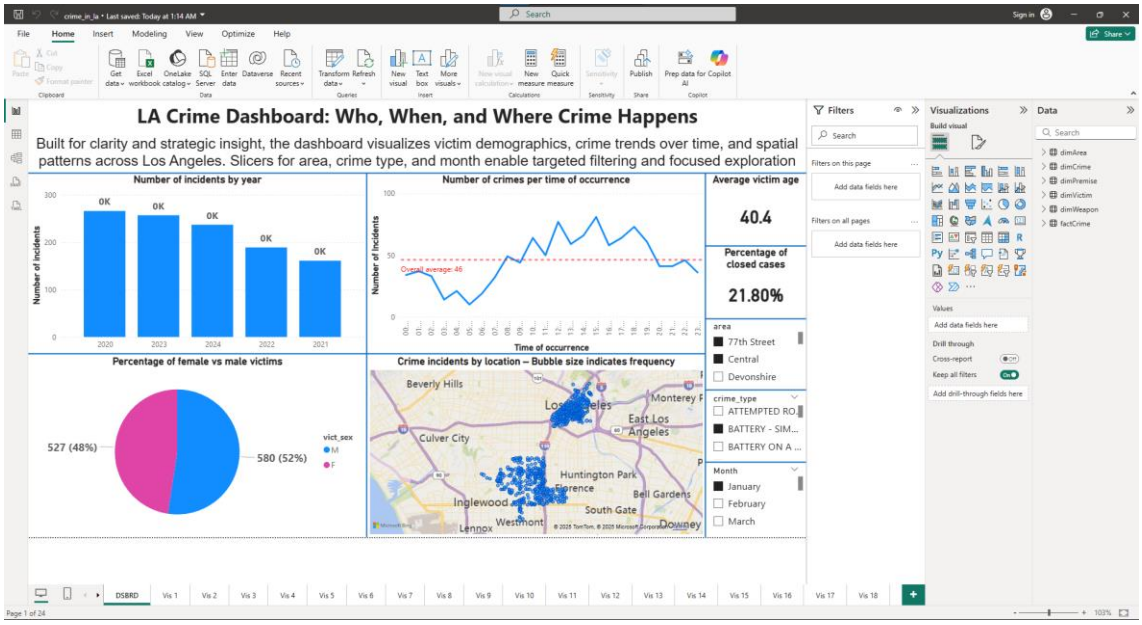
IX) Insert two more lines and convert them to vertical by going to “Format shape” menu, expanding “Rotation”, and setting “All” to 90 degrees for both of the lines. Again, adjust their height, width, and position to properly segregate the visuals.



Screenshot 121: “Vertical lines added”

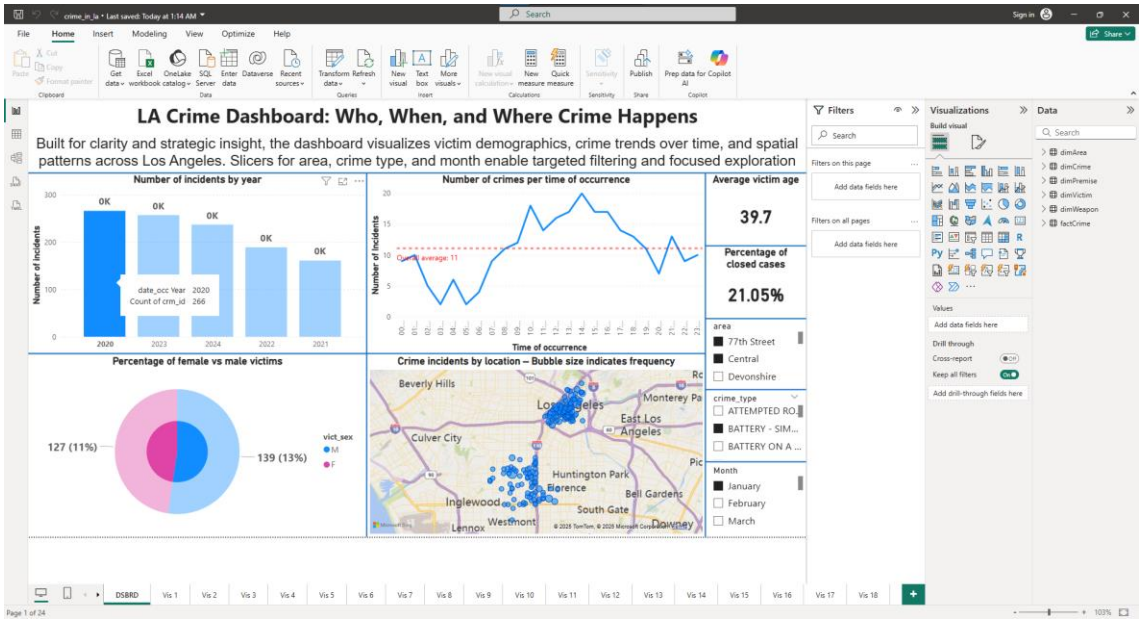
X) To use the categories of the added slicers to filter the visuals, click on the box next to a category to select it, e.g., “Central” as area. Selecting multiple

categories per slicer can be easily achieved by holding down “CTRL” and left-clicking”.



Screenshot 122: “Slicers’ categories selected”

XI) To cross-filter between visuals based on a visual’s category, click on a category within a visual e.g., the year 2020. To cross-filter using more than one category from one or more visuals, simply select all the desired categories by holding down “CTRL” and left-clicking.



Screenshot 123: “Year category selected”