CITS3401 PROJECT 1

BY: Nicodemus Ong (22607943)

Concept Hierarchies and Schema Designs

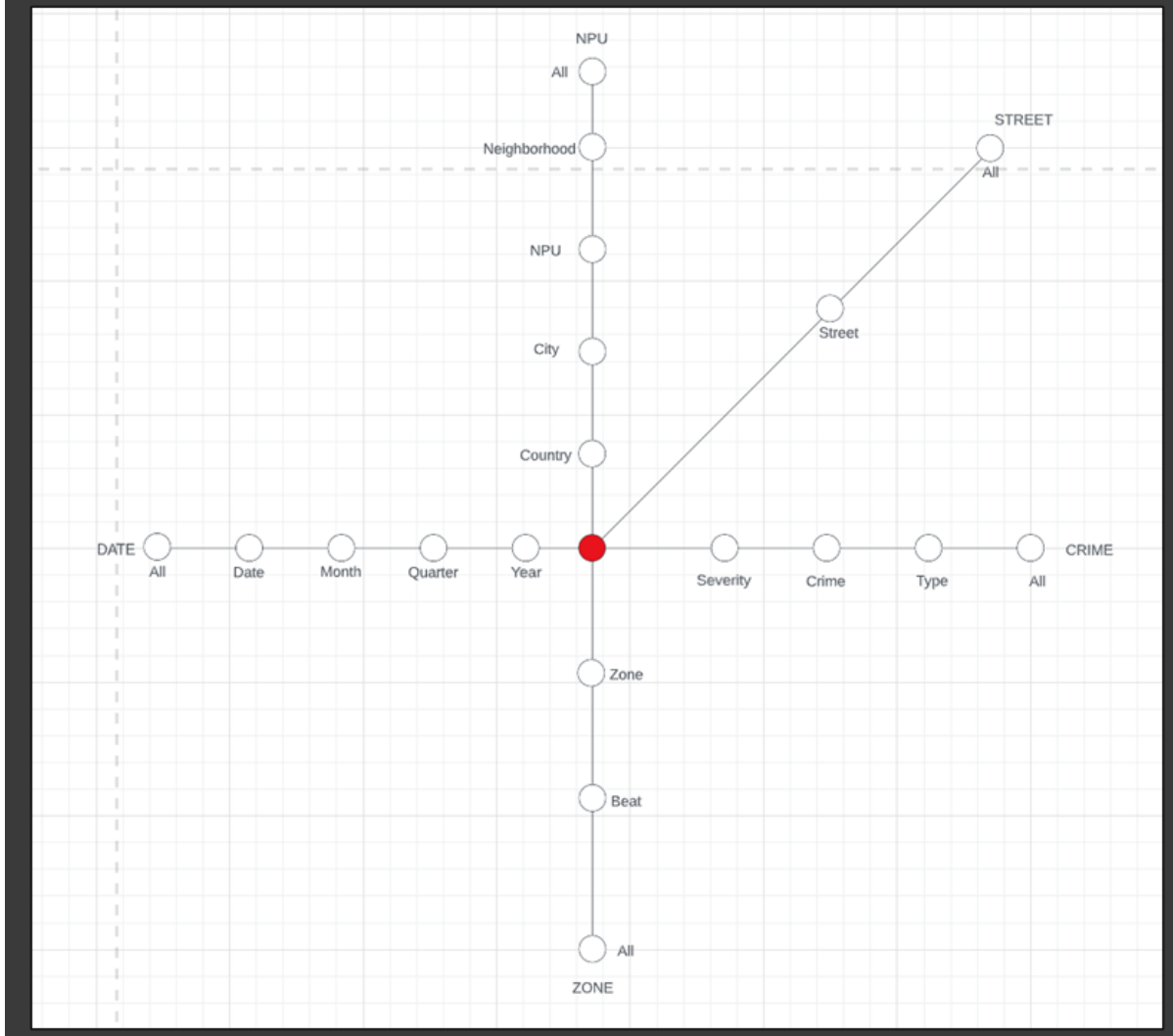
The concept Hierarchy is a critical tool in organizing complex data in a data warehouse. in this case, the hierarchy is based on the severity of crimes, which has been split into three groups: LOW, MEDIUM, HIGH. The first step to creating this hierarchy was ordering the 11 different crime types based on their severity. The most severe crime was ranked as number one, and the least severe crime was ranked as number 11.

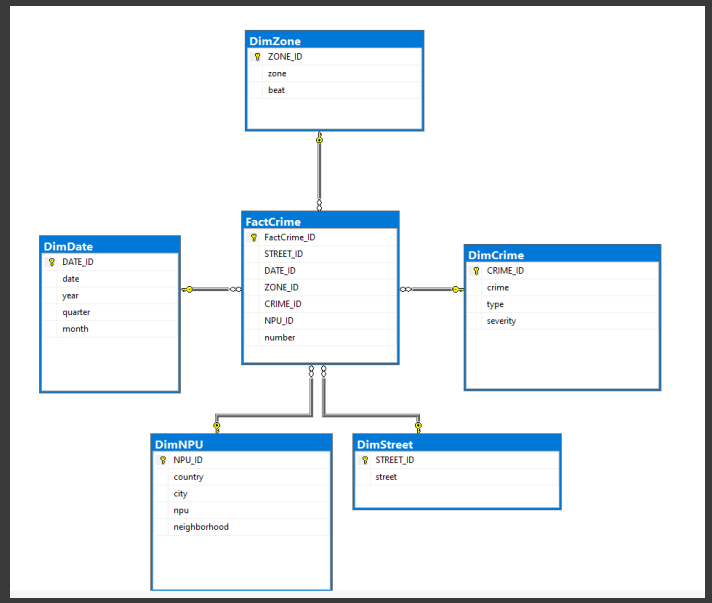
The order of the crimes was determined based on the potential harm and impact each crime could have on individuals and society. For instance, homicide, which involves taking a life, was ranked as the most severe crime, while larceny-nonvehicle, a non-violent crime, was ranked lower. once the crimes were ranked in the desired order, they were then split into three groups of 'HIGH', 'MEDIUM' AND 'LOW'. The top three crimes, which were ranked as the most severe, were placed in the 'high' group primarily because they cause the most harm to the victim, both physically and emotionally. the next four crimes were ranked as 'MEDIUM' because they involve force, threat, and intimidation of the perpetrator/s to the victim, although the harm may not be as severe as in the 'HIGH' group.

The last four crimes were ranked as 'LOW' because the victim was not involved in the crime at the time it occurred. These crimes typically involve the perpetrator stealing or taking something of the victim without their permission or knowledge. Since the victim is not at harm or unlikely to be in that situation, they are ranked under low.

Therefore, the final concept hierarchy of the data warehouse has three layers, with the root being 'ALL' data, which then branches out into the three different severity types and finally the 11 different crimes under their respective severity. This hierarchy provides a clear and organized way of categorizing and analyzing data related to crimes and their severity, making it easier to identify patterns, trends and insights.

StarNet and DataBase Design





In addition to the concept hierarchy of the data warehouse, each level of the hierarchy s further grouped and categorized to ensure efficient mapping to their respective dimension tables. The first category, 'ALL', includes all data within the data warehouse, regardless of crime type or severity level. The second category, 'crime severity', include the three different severity types: 'HIGH', 'MEDIUM' and 'LOW'. Finally, the third category, 'crime types', includes the 11 different types of crime within their respective severity level.

These categories are then mapped to their respective dimension tables in the StarNet model. This ensures that the data is organized in a manner that facilitates effective querying and analysis of the data. By categorizing the data in this way, it becomes much easier to retrieve specific information about the different types of crimes and their severity levels. This can be especially useful for law enforcement agencies and other organizations that require detailed information on crime statistics to inform their decision-making processes.

The diagram illustrates the design of the StarNet for this project's data warehouse as well as the data warehouse design with the fact and dimension tables

Creating Fact Crime Table

```
CREATE TABLE FactCrime

(

  FactCrime_ID INT IDENTITY(1,1) PRIMARY KEY,
  STREET_ID INT,

  DATE_ID INT,

  ZONE_ID INT,

  CRIME_ID INT,

  NPU_ID INT,

  number VARCHAR(50)

);

GO
```

	FactCrime_ID	STREET_ID	DATE_ID	ZONE_ID	CRIME_ID	NPU_ID	number
1	1	1	2	1	1	53	90021275
2	2	1	6	1	1	53	90060010
3	3	1	9	1	1	53	90090120
4	4	1	13	1	17	1	90132038
5	5	1	14	1	3	1	90141251
6	6	1	26	1	15	1	90261297
7	7	1	44	1	3	1	90440988
8	8	1	46	1	5	1	90460719
9	9	1	46	1	5	53	90461119
10	10	1	68	1	5	53	90681912
11	11	1	72	1	1	53	90721290
12	12	1	88	1	1	1	90881567
13	13	1	93	1	2	53	90932185
14	14	1	116	1	2	1	91160635
15	15	1	122	1	5	1	91220880
16	16	1	135	1	1	53	91352147
17	17	1	143	1	5	53	91431911
18	18	1	143	1	15	1	91430805
19	19	1	150	1	3	1	91500772

```
ALTER TABLE [dbo].[FactCrime] ADD CONSTRAINT
FK_CRIME_ID FOREIGN KEY (CRIME_ID) REFERENCES [dbo].[DimCrime] (CRIME_ID);

ALTER TABLE [dbo].[FactCrime] ADD CONSTRAINT
FK_ZONE_ID FOREIGN KEY (ZONE_ID) REFERENCES [dbo].[DimZone] (ZONE_ID);

ALTER TABLE [dbo].[FactCrime] ADD CONSTRAINT
FK_NPU_ID FOREIGN KEY (NPU_ID) REFERENCES [dbo].[DimNPU] (NPU_ID);

ALTER TABLE [dbo].[FactCrime] ADD CONSTRAINT
FK_STREET_ID FOREIGN KEY (STREET_ID) REFERENCES [dbo].[DimStreet] (STREET_ID);

ALTER TABLE [dbo].[FactCrime] ADD CONSTRAINT
FK_DATE_ID FOREIGN KEY (DATE_ID) REFERENCES [dbo].[DimDate] (DATE_ID);

GO
```

The fact table was generated after the cleaning and data prepped was finalized with the ETL code. columns such as long, lat and location were dropped because their information was too dirty and it was unable to be cleaned efficiently. An example of a dirty location is "OAKHILL AVE SW / LILIAN AVE SW" and "713 RALPH D ABERNATHY BLVD SW ; STEVENS GRAPHICS". Such data would take too much effort and computing power to be able to effectively and efficiently clean it hence the decision was to drop the column entirely. Similar situations were for the Postcode as well as long and lat.

The crime Fact Table is a table in this data warehouse that stores the quantitative information or facts of a business process, such as the number of crimes that occurred. The fact table has the following columns:

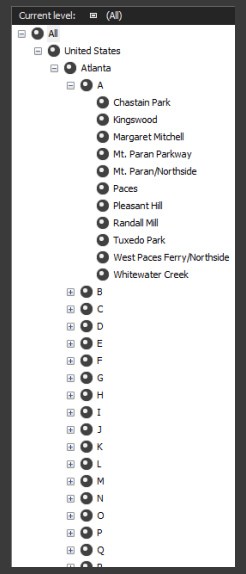
1. factCrime_ID: A unique identifier for each row in the fact table
2. STREET_ID: A foreign key that links to the street dimension table
3. DATE_ID: A foreign key that link to the date dimension table
4. ZONE_ID: A foreign key that links to the zone dimension table
5. CRIME_ID: A foreign key that links to the crime dimension table
6. NPU_ID: A foreign key that links to the NPU (Neighborhood Planning Unit) dimension table
7. number: The unique serial number of that specific crime that was committed

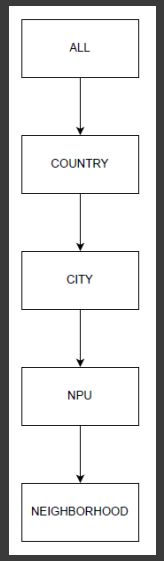
As for the constraints, foreign keys to their respective dimension tables are:

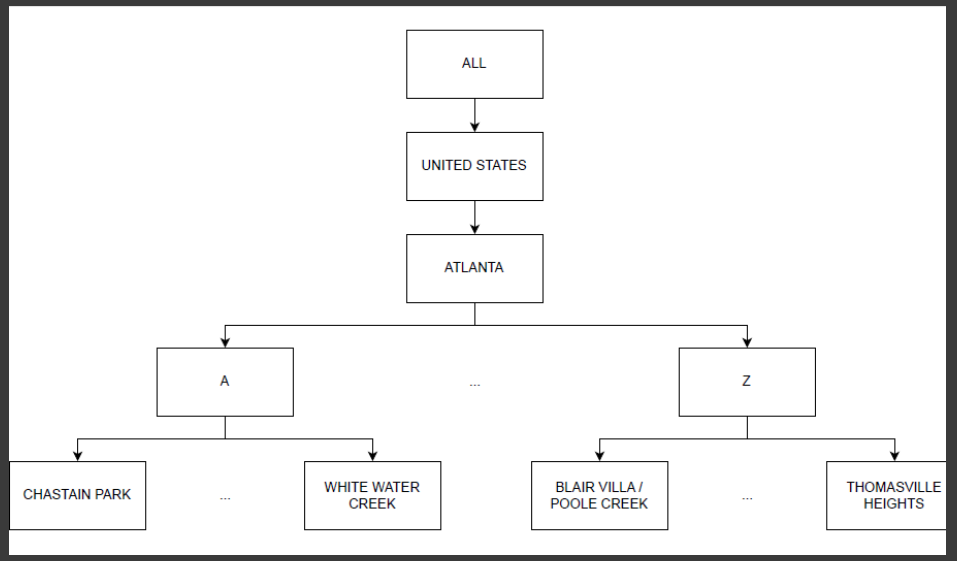
1. DimCrime: This table would contain information about each type of crime, such as severity, crime and crime type. The primary key in this table would be CRIME_ID, which is referenced in the FactCrime table as a foreign key.
2. DimZone: This table would contain information about each geographic zone in the dataset, such as the name, description, and any other relevant attributes. The primary key in this table would be ZONE_ID, which is referenced in the FactCrime table as a foreign key.
3. DimNPU: This table would contain information about each Neighborhood Planning Unit (NPU) in the dataset, such as the name, description, and any other relevant attributes. The primary key in this table would be NPU_ID, which is referenced in the FactCrime table as a foreign key.
4. DimStreet: This table would contain information about each street in the dataset, such as the name, location, and any other relevant attributes. The primary key in this table would be STREET_ID, which is referenced in the FactCrime table as a foreign key.
5. DimDate: This table would contain information about each date in the dataset, such as the year, quarter, month, date and any other relevant attributes. The primary key in this table would be DATE_ID, which is referenced in the FactCrime table as a foreign key.

These constraints ensure that the fact table references the correct dimensions and enables efficient querying and analysis of the data. For example, you can analyze the number of crimes that occurred on a specific street on a specific date, or the total number of crimes that occurred in a specific zone.

NPU Dim Concept & Schema Hierarchy & SQL Table





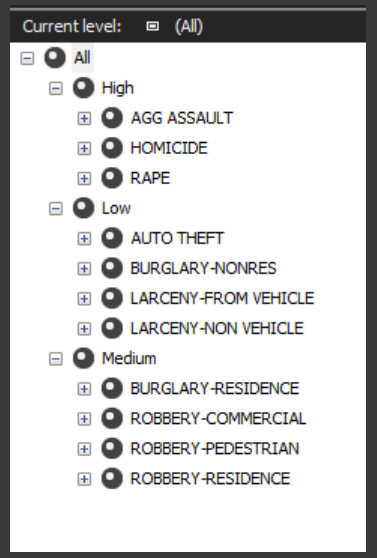


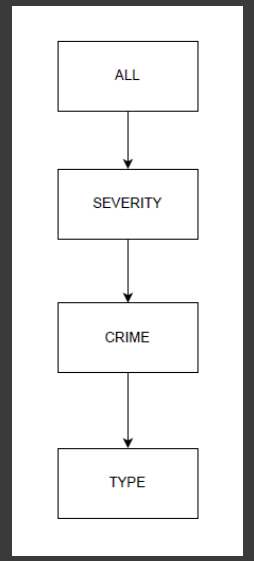
These are the concept and schema hierarchy for the NPU DIM table. In this schema hierarchy, the highest level is "all", which would include all the crimes reported in the dataset regardless of location. The next level down is "country", which would represent all the crimes reported within a specific country, for this case it is just the united states. The level below that is "city", which would represent all the crimes reported within a specific city, for this case its "Atlanta". The next level is "NPU", which stands for Neighborhood Planning Unit, which is a geographic designation used in some cities to divide the city into smaller areas for planning and resource allocation purposes. The NPU level would represent all the crimes reported within a specific NPU in a city. The lowest level of the hierarchy is "neighborhood", which would represent all the crimes reported within a specific neighborhood or community within an NPU.
Using this schema hierarchy, crime data can be analyzed at different geographic levels, from the broadest level of all reported crimes down to specific neighborhoods within an NPU. This allows for a more detailed understanding of crime patterns and trends at different levels of granularity

	NPU_ID	country	city	npu	neighborhood
1	1	United States	Atlanta	V	Adair Park
2	2	United States	Atlanta	R	Adams Park
3	3	United States	Atlanta	H	Adamsville
4	4	United States	Atlanta	G	Almond Park
5	5	United States	Atlanta	Y	Amal Heights
6	6	United States	Atlanta	E	Ansley Park
7	7	United States	Atlanta	C	Arden/Habersham
8	8	United States	Atlanta	E	Ardmore
9	9	United States	Atlanta	C	Argonne Forest
10	10	United States	Atlanta	P	Arlington Estates
11	11	United States	Atlanta	P	Ashley Courts
12	12	United States	Atlanta	T	Ashview Heights
13	13	United States	Atlanta	F	Atkins Park
14	14	United States	Atlanta	G	Atlanta Industrial Park
15	15	United States	Atlanta	T	Atlanta University Center
16	16	United States	Atlanta	E	Atlantic Station
17	17	United States	Atlanta	H	Baker Hills
18	18	United States	Atlanta	H	Bakers Ferry
19	19	United States	Atlanta	K	Bankhead

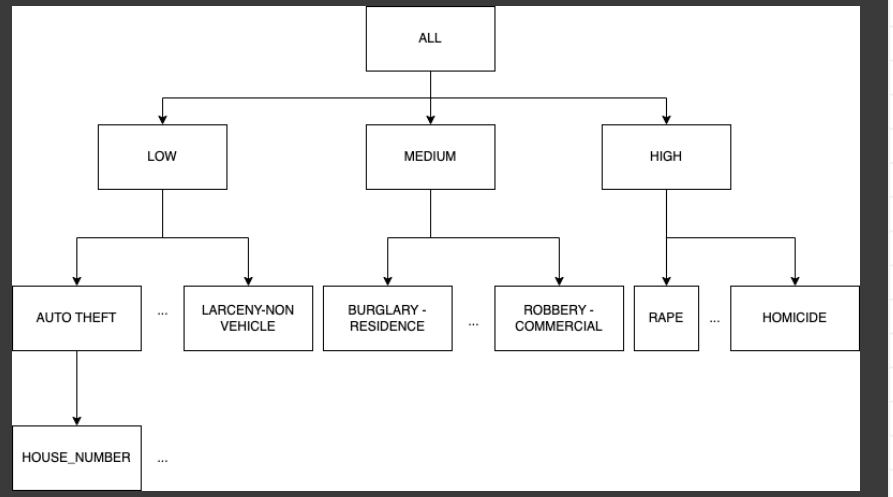
```
CREATE TABLE DimNPU (
    NPU_ID INT IDENTITY(1,1) PRIMARY KEY,
    country VARCHAR(50),
    city VARCHAR(50),
    npu CHAR(1),
    neighborhood VARCHAR(50)
);
```

Crime Concept & Schema Hierarchy & SQL Table





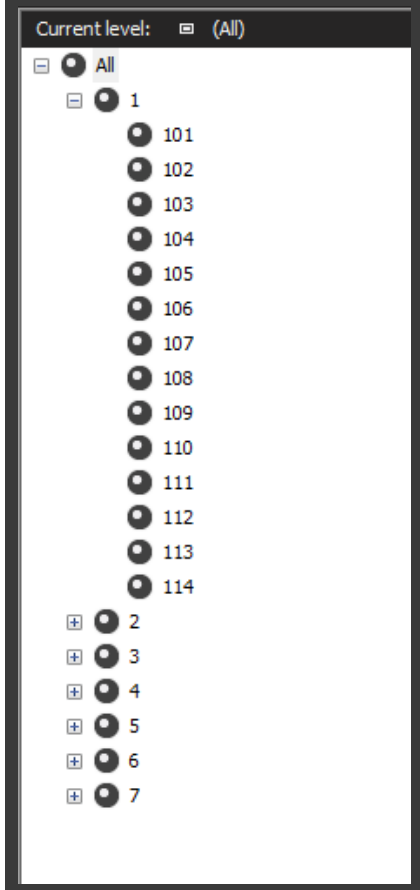
These are the concept and schema hierarchy for the crime DIM table. The ALL level would represent the highest level of aggregation for the crime data set. It would include all reported crimes, regardless of severity or type. This level would provide an overall picture of the crime situation in a given area or time period.
The SEVERITY level would represent the next level of aggregation, focusing on the severity of crimes reported. This level would categorize crimes based on their severity, such as LOW, MEDIUM and HIGH. Each category would contain a subset of the crimes included in the ALL level. The CRIME level would represent the next level of aggregation, focusing on what kind of crimes were committed, such as homicide, robbery, larceny etc. The TYPE would represent the lowest level of aggregation, focusing on the location and any additional information available about that specific crime that was committed.
This schema hierarchy allows for a more detailed analysis of crime data, starting with a broad overview of all reported crimes and then drilling down to specific crime types and severity levels.

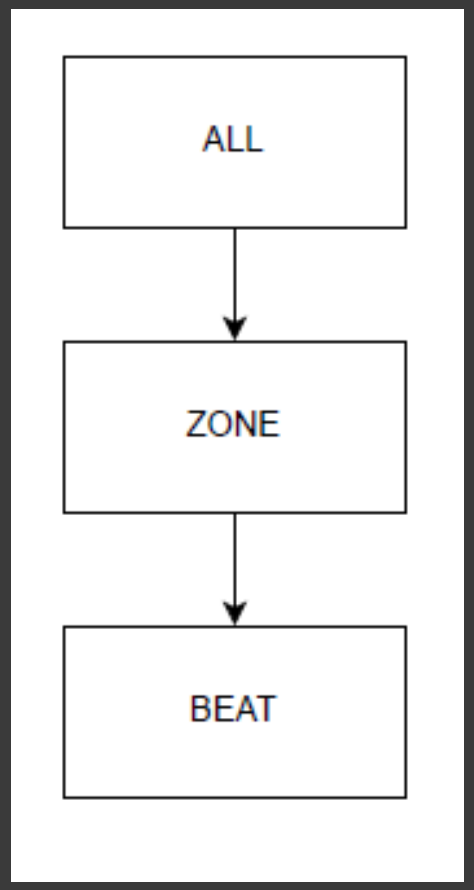


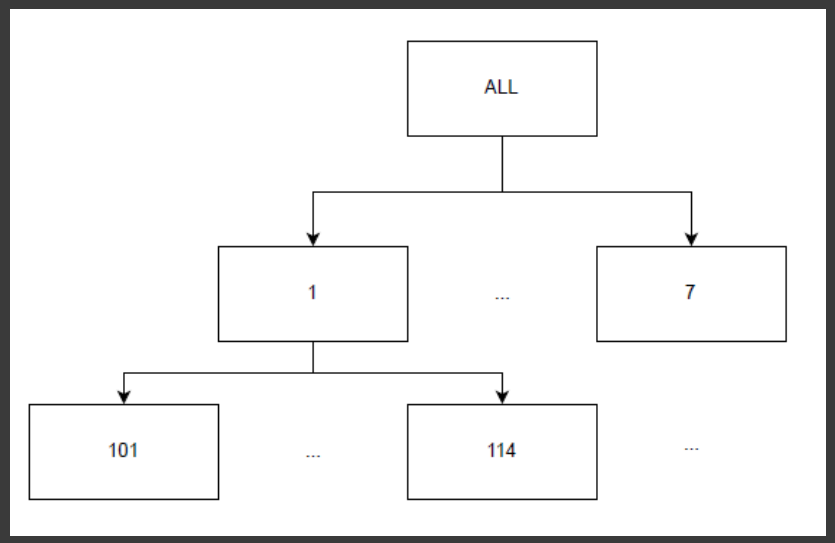
	CRIME_ID	crime	type	severity
1	1	BURGLARY-RESIDENCE	house_number	Medium
2	2	AGG ASSAULT	house_number	High
3	3	LARCENY-NON VEHICLE	house_number	Low
4	4	ROBBERY-PEDESTRIAN	amenity	Medium
5	5	AUTO THEFT	house_number	Low
6	6	BURGLARY-NONRES	house_number	Low
7	7	LARCENY-FROM VEHICLE	house_number	Low
8	8	AUTO THEFT	place	Low
9	9	LARCENY-FROM VEHICLE	place	Low
10	10	ROBBERY-PEDESTRIAN	house_number	Medium
11	11	AGG ASSAULT	place	High
12	12	LARCENY-NON VEHICLE	shop	Low
13	13	LARCENY-NON VEHICLE	amenity	Low
14	14	AUTO THEFT	amenity	Low
15	15	RAPE	house_number	High
16	16	AGG ASSAULT	amenity	High
17	17	ROBBERY-RESIDENCE	house_number	Medium
18	18	LARCENY-FROM VEHICLE	shop	Low
19	19	LARCENY-NON VEHICLE	place	Low

```
CREATE TABLE DimCrime (

    CRIME_ID INT IDENTITY(1,1) PRIMARY KEY,

    crime VARCHAR(MAX),

    type VARCHAR(50),

    severity VARCHAR(50)

);
```

Zone Dim Concept & Schema Hierarchy & SQL Table





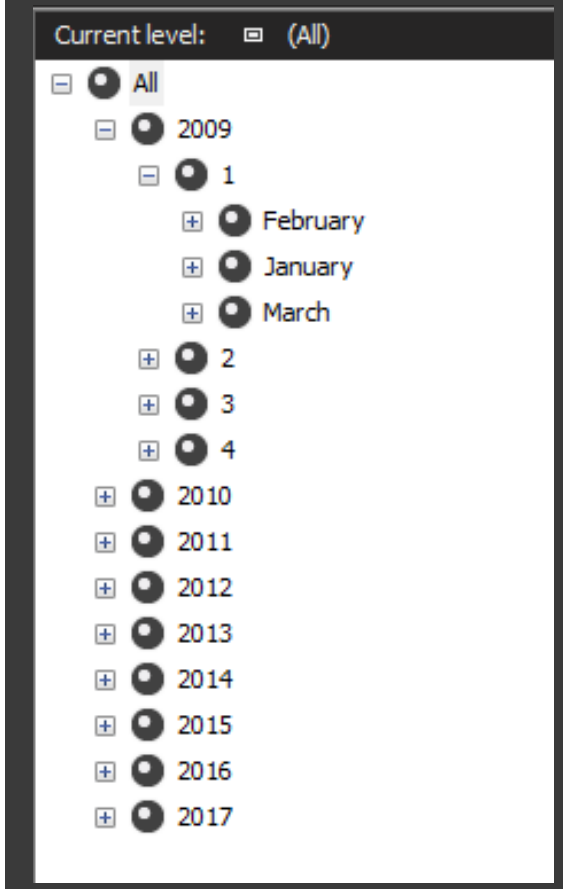


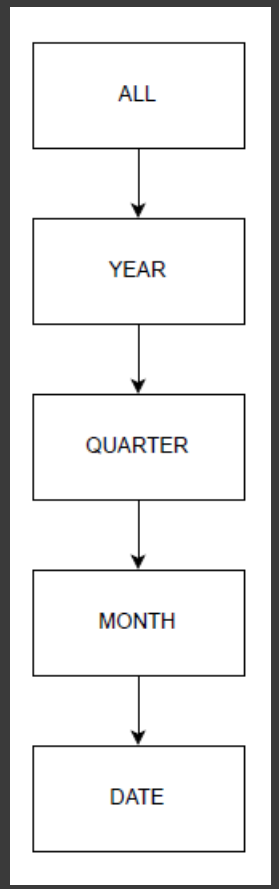
These are the concept and schema hierarchy for the NPU DIM table. In this schema hierarchy, the highest level is "ALL", which would include all the crimes reported in the dataset regardless of location. The next level down is "ZONE", which would represent all the crimes reported within a specific geographic zone. The level below that is "BEAT", which would represent all the crimes reported within a specific beat or patrol area within a zone.
Using this schema hierarchy, crime data can be analyzed at different geographic levels, from the broadest level of all reported crimes down to specific beats within a zone. This allows for a more detailed understanding of crime patterns and trends at different levels of granularity.

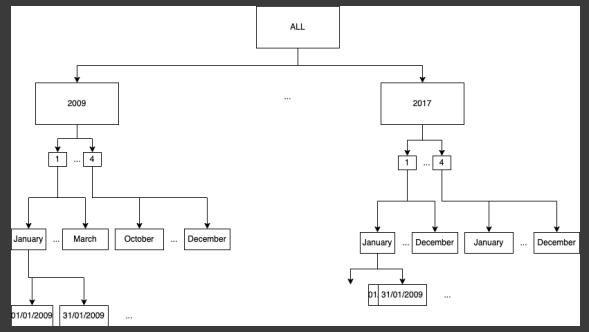
	ZONE_ID	zone	beat
1	1	3	301
2	2	3	303
3	3	3	304
4	4	4	401
5	5	1	101
6	6	5	507
7	7	3	302
8	8	4	403
9	9	3	307
10	10	7	702
11	11	4	409
12	12	4	410
13	13	1	108
14	14	4	408
15	15	4	407
16	16	4	406
17	17	4	411
18	18	4	412
19	19	4	413

```
CREATE TABLE DimZone (

    ZONE_ID INT IDENTITY(1,1) PRIMARY KEY,

    zone VARCHAR(50),

    beat VARCHAR(50)

);
```

Date Concept & Schema Hierarchy & SQL Tables





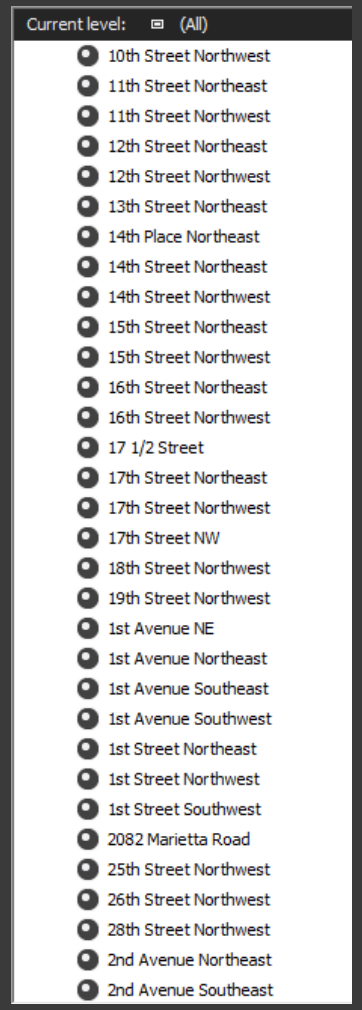


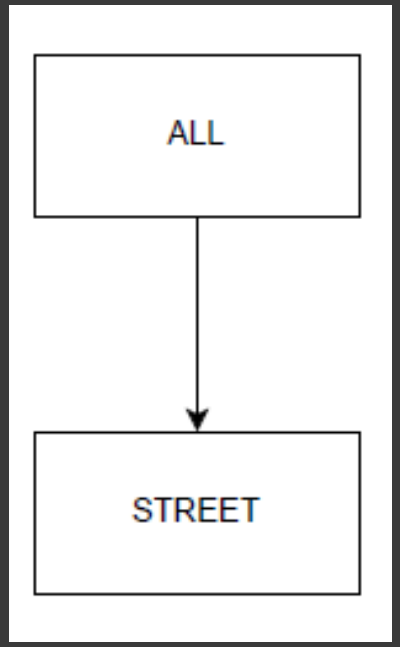
These are the concept and schema hierarchy for the Date DIM table. In this schema hierarchy, the highest level is "ALL", which would include all the crimes reported in the dataset regardless of the time period. The next level down is "YEAR", which would represent all the crimes reported within a specific year. The level below that is "QUARTER", which would represent all the crimes reported within a specific quarter of the year. The next level is "MONTH", which would represent all the crimes reported within a specific month. Finally, the "DATE" level would represent all the crimes reported on a specific day.
Using this schema hierarchy, crime data can be analyzed at different levels of granularity, from the broadest level of all reported crimes down to specific dates. This allows for a more detailed understanding of crime patterns and trends over time.

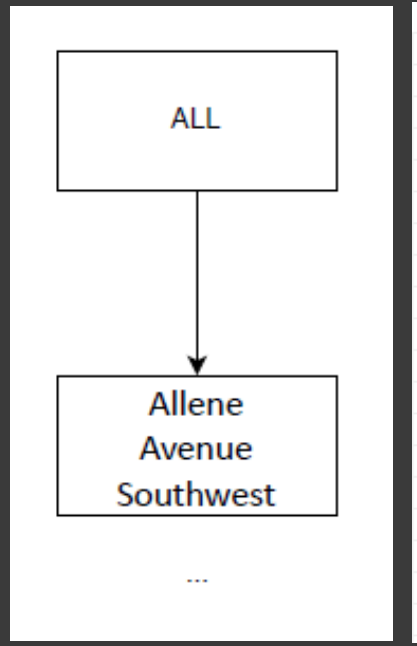
	DATE_ID	date	year	quarter	month
1	1	1/01/2009	2009	1	January
2	2	2/01/2009	2009	1	January
3	3	3/01/2009	2009	1	January
4	4	4/01/2009	2009	1	January
5	5	5/01/2009	2009	1	January
6	6	6/01/2009	2009	1	January
7	7	7/01/2009	2009	1	January
8	8	8/01/2009	2009	1	January
9	9	9/01/2009	2009	1	January
10	10	10/01/2009	2009	1	January
11	11	11/01/2009	2009	1	January
12	12	12/01/2009	2009	1	January
13	13	13/01/2009	2009	1	January
14	14	14/01/2009	2009	1	January
15	15	15/01/2009	2009	1	January
16	16	16/01/2009	2009	1	January
17	17	17/01/2009	2009	1	January
18	18	18/01/2009	2009	1	January
19	19	19/01/2009	2009	1	January

```
CREATE TABLE DimDate (
    DATE_ID INT IDENTITY(1,1) PRIMARY KEY,
    date VARCHAR(50),
    year VARCHAR(50),
    quarter VARCHAR(50),
    month VARCHAR(50)
);
```

Street Concept & Schema Hierarchy & SQL Tables







	STREET_ID	street
1	1	Allene Avenue Southwest
2	2	University Avenue Southwest
3	3	Murphy Avenue Southwest
4	4	Northside Drive Southwest
5	5	Pearce Street Southwest
6	6	Lexington Avenue Southwest
7	7	Tift Avenue Southwest
8	8	Brookline Street Southwest
9	9	Metropolitan Parkway Southwest
10	10	Elbert Street Southwest
11	11	Tift Street Southwest
12	12	Mayland Avenue Southwest
13	13	Shelton Avenue Southwest
14	14	Bonnie Brae Avenue Southwest
15	15	Whitehall Street Southwest
16	16	Wells Street Southwest
17	17	Gillette Avenue Southwest
18	18	Catherine Street Southwest
19	19	Oakhill Avenue Southwest

```sql
CREATE TABLE DimStreet (

    STREET_ID INT IDENTITY(1,1) PRIMARY KEY,

    street VARCHAR(MAX)

);
```

These are the concept and schema hierarchy for the Street DIM table. In this schema hierarchy, the highest level is "ALL", which would include all the crimes reported in the dataset regardless of the location. The next level down is "STREET", which would represent all the crimes reported on a specific street.
Using this schema hierarchy, crime data can be analyzed at the street level, which allows for a more focused understanding of crime patterns and trends within specific areas. it's important to note, however, that this schema hierarchy may not be suitable for all crime datasets, particularly those that involves crimes occurring across a wider range of locations.
in such cases, a more complex schema hierarchy with additional levels may be necessary to properly analyze the data which can be a further enhancement for the future.

BULK INSERT OF TABLES

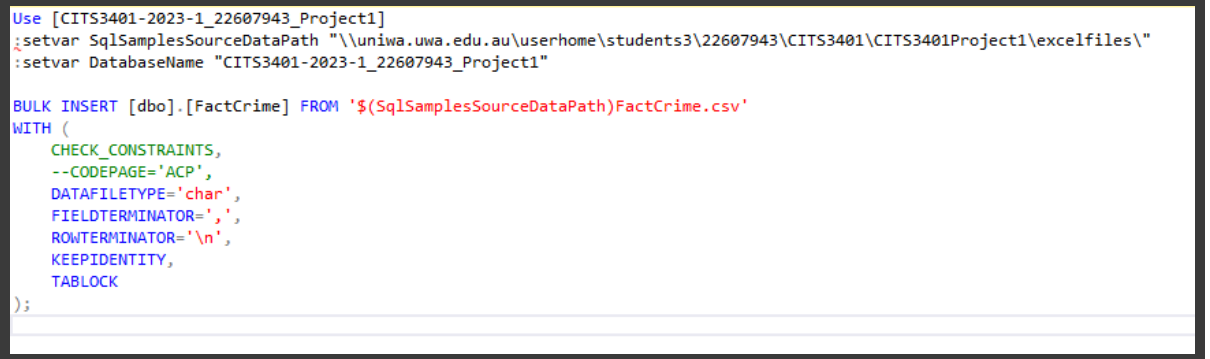
To bulk insert into the different tables there are a few steps to follow and note:

1. The first line sets the name of the project to "CITS3401-2023-1_22607943_Project1".
2. The second line sets the variable "SqlSamplesSourceDataPath" to the file path of the CSV file that will be used for the bulk insert operation. The file path is "\uniwa.uwa.edu.au\userhome\students3\22607943\CITS3401\CITS3401Project1\excelfiles".
3. The third line sets the variable "DatabaseName" to the name of the database where the bulk insert operation will take place. The database name is "CITS3401-2023-1_22607943_Project1".
4. The bulk insert operation starts with the BULK INSERT statement, which specifies the name of the table ("[dbo].[insert db name here]") that will be populated with data from the CSV file.
5. The WITH clause specifies various options for the bulk insert operation. The CHECK_CONSTRAINTS option ensures that data inserted into the table meets any constraints defined for the table. The DATAFILETYPE option specifies that the data in the CSV file is in character format. The FIELDTERMINATOR option specifies that the fields in the CSV file are separated by commas. The ROWTERMINATOR option specifies that the end of each row in the CSV file is marked by a newline character. The KEEPIDENTITY option specifies that any identity values in the table should be preserved during the insert operation. The TABLOCK option specifies that a table-level lock should be taken during the bulk insert operation.

The script will run and execute to insert all the data in the respective CSV files into the dimension and fact tables. It is good to note as well that the bulk insert of the Fact table and the other tables are different. This is due to the rigorous testing of the constraints on the data and configuration of the database. However you can combine both and execute.

The SQL script files are provided in the submission zip.

```
Use [CITS3401-2023-1_22607943_Project1]
:setvar SqlSamplesSourceDataPath "\\uniwa.uwa.edu.au\userhome\students3\22607943\CITS3401\CITS3401Project1\excelfiles\"
:setvar DatabaseName "CITS3401-2023-1_22607943_Project1"

BULK INSERT [dbo].[DimCrime] FROM '$(SqlSamplesSourceDataPath)DimCrime.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    KEEPIDENTITY,
    TABLOCK
);

BULK INSERT [dbo].[DimDate] FROM '$(SqlSamplesSourceDataPath)DimDate.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    KEEPIDENTITY,
    TABLOCK
);

BULK INSERT [dbo].[DimNPU] FROM '$(SqlSamplesSourceDataPath)DimNPU.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    KEEPIDENTITY,
    TABLOCK
);
BULK INSERT [dbo].[DimStreet] FROM '$(SqlSamplesSourceDataPath)DimStreet.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    KEEPIDENTITY,
    TABLOCK
);
```



```
Use [CITS3401-2023-1_22607943_Project1]
:setvar SqlSamplesSourceDataPath "\\uniwa.uwa.edu.au\userhome\students3\22607943\CITS3401\CITS3401Project1\excelfiles\"
:setvar DatabaseName "CITS3401-2023-1_22607943_Project1"

BULK INSERT [dbo].[FactCrime] FROM '$(SqlSamplesSourceDataPath)FactCrime.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    KEEPIDENTITY,
    TABLOCK
);
```

ETL Processes

Cleaning and combining data from CSV

The ETL process for this project was carried out using a python script called 'ETL.py', which heavily utilized the 'csv' and 'pandas' modules along with others. Since the data was provided in multiple CSV files, the script not only extracted and cleaned the data but also concatenated all CSVs together and merged the data once processing was complete. The data was further split into their respective Dimension/Fact Files.

The majority of the processing occurred in the 'combine_files(folder_path, output_file)' method. This method reads in all the CSV files provided for the dataset and concatenated them into a single data frame. This data frame was then processed in four ways. Firstly, any unused columns ere dropped from the data set since they were not necessary. Secondly, new columns were added with data determined by existing data in the data frame. Thirdly, any data that were incorrect were modified/replaced. Fourthly, data that were in mixed formats that would cause issues when pushing data into the SQL tables were reformatted, such as dates. Finally, any entries that could not be salvaged were removed.

one example of the data processing is the NPU data. Upon researching and analyzing the NPU data, it was discovered that the neighborhoods did not correspond to the respective NPU assigned to them. This information was taken from the official police website of Atlanta. hence, a section of the ETL process was dedicated to removing the existing NPU data and replacing it with the data of the correct NPU that corresponded to their respective neighborhoods.

The screenshots show sections of what was described such as NPU cleaning along with data cleaning and manipulation.

The full code will be provided in the 'ETL.py' file that has been submitted together

```
# Open clean NPU file to replace existing NPU as it is broken, file generated from
# https://citycouncil.atlantaga.gov/other/neighborhood-planning-unit
clean_npu_df = pd.read_csv('npu_clean.csv')

# Replace blank entries in 'neighbourhood_lookup' with neighborhood data from the same row
combined_df['neighbourhood_lookup'].fillna(combined_df['neighborhood'], inplace=True)

# Add new column "Severity", results based of crime committed
combined_df['Severity'] = combined_df['crime'].apply(determining_severity)

# Add new column "zone" based on data of beat || First digit of beat is the zone that that record belongs in
combined_df['zone'] = combined_df['beat'].astype(str).str[0]

# drop county cause its broken based on lab tutor JiChuYang recommendations. remove NPU as well as it will be added
# back later. NPU is broken as well. Other columns has been dropped as well as it is irrelevant
combined_df.drop(columns=['county', 'Unnamed: 0.1', 'Unnamed: 0', 'npu', 'location'], inplace=True)

# Merge 'combined_df' with 'clean_npu_df' based on 'neighborhood' column
combined_df = combined_df.merge(clean_npu_df[['neighborhood', 'npu']], on='neighborhood', how='inner')

# Format the date to YYYY-MM-DD for SQL processing
combined_df['date'] = pd.to_datetime(combined_df['date'], format='mixed').dt.strftime('%Y-%m-%d')

# Replace all city values to Atlanta as to avoid error in data processing later on
combined_df['city'] = 'Atlanta'
```

Data Extraction into respective Dim/Fact tables

```python
def npu_DIM():
    """
    Generate npu_DIM csv

    Returns:
        npu_DIM.csv file
    """

    df = pd.read_csv('complete_data.csv')

    # Drop duplicate rows based on 'country', 'city', 'npu', 'neighborhood' columns
    df.drop_duplicates(subset=['country', 'city', 'npu', 'neighborhood'], inplace=True)

    # Reset the index after dropping duplicates
    df.reset_index(drop=True, inplace=True)

    # Add a new column with sequential values in the format "1" to "n"
    num_entries = len(df)

    df['NPU_ID'] = [(i + 1) for i in range(num_entries)]

    # Extract country, city, npu, neighborhood columns and add in NPU_ID
    selected_columns = df[['NPU_ID', 'country', 'city', 'npu', 'neighborhood']].copy()

    # Export to CSV file
    selected_columns.to_csv('npu_DIM.csv', index=False)

    # Print a message upon successful export
    print("CSV file exported successfully as npu_DIM.csv")
```

```python
def crimes_FACT():
    """
    Generate crimes_FACT from the 'crimeInfo_DIM.csv', 'zone_DIM.csv', 'npu_DIM.csv', 'dates_DIM.csv', and
    'complete_data.csv' files.

    Returns:
        None
    """

    # Open CSV files to be matched
    crimeInfo_df = pd.read_csv('crimeInfo_DIM.csv')  # Read 'crimeInfo_DIM.csv' into a DataFrame
    zone_df = pd.read_csv('zone_DIM.csv')  # Read 'zone_DIM.csv' into a DataFrame
    npu_df = pd.read_csv('npu_DIM.csv')  # Read 'npu_DIM.csv' into a DataFrame
    date_df = pd.read_csv('dates_DIM.csv')  # Read 'dates_DIM.csv' into a DataFrame
    street_df = pd.read_csv('street_DIM.csv')  # Read street_DIM.csv

    master_df = pd.read_csv('complete_data.csv')  # Read 'complete_data.csv' into a DataFrame

    # Replace 'NPU' column in master_df with 'ID' from npu_df by matching on 'country', 'city', 'npu', 'neighborhood'
    tmp_df = pd.merge(npu_df, master_df, on=['country', 'city', 'npu', 'neighborhood'], how='inner').drop(
        ['country', 'city', 'npu', 'neighborhood'], axis=1)

    # Replace 'crime' and 'type' columns in tmp_df with 'ID' from crimeInfo_df by matching on 'crime', 'type', 'Severity'
    tmp_df = pd.merge(crimeInfo_df, tmp_df, on=['crime', 'type', 'Severity'], how='inner').drop(
        ['crime', 'type', 'Severity'], axis=1)

    # Add 'ZONE_ID' column in tmp_df and populate with the first digit of 'beat' column from zone_df
    tmp_df = pd.merge(zone_df, tmp_df, on=['zone', 'beat'], how='inner').drop(['zone', 'beat'], axis=1)

    # Replace 'Date' column in tmp_df with 'ID' from date_df by matching on 'date'
```

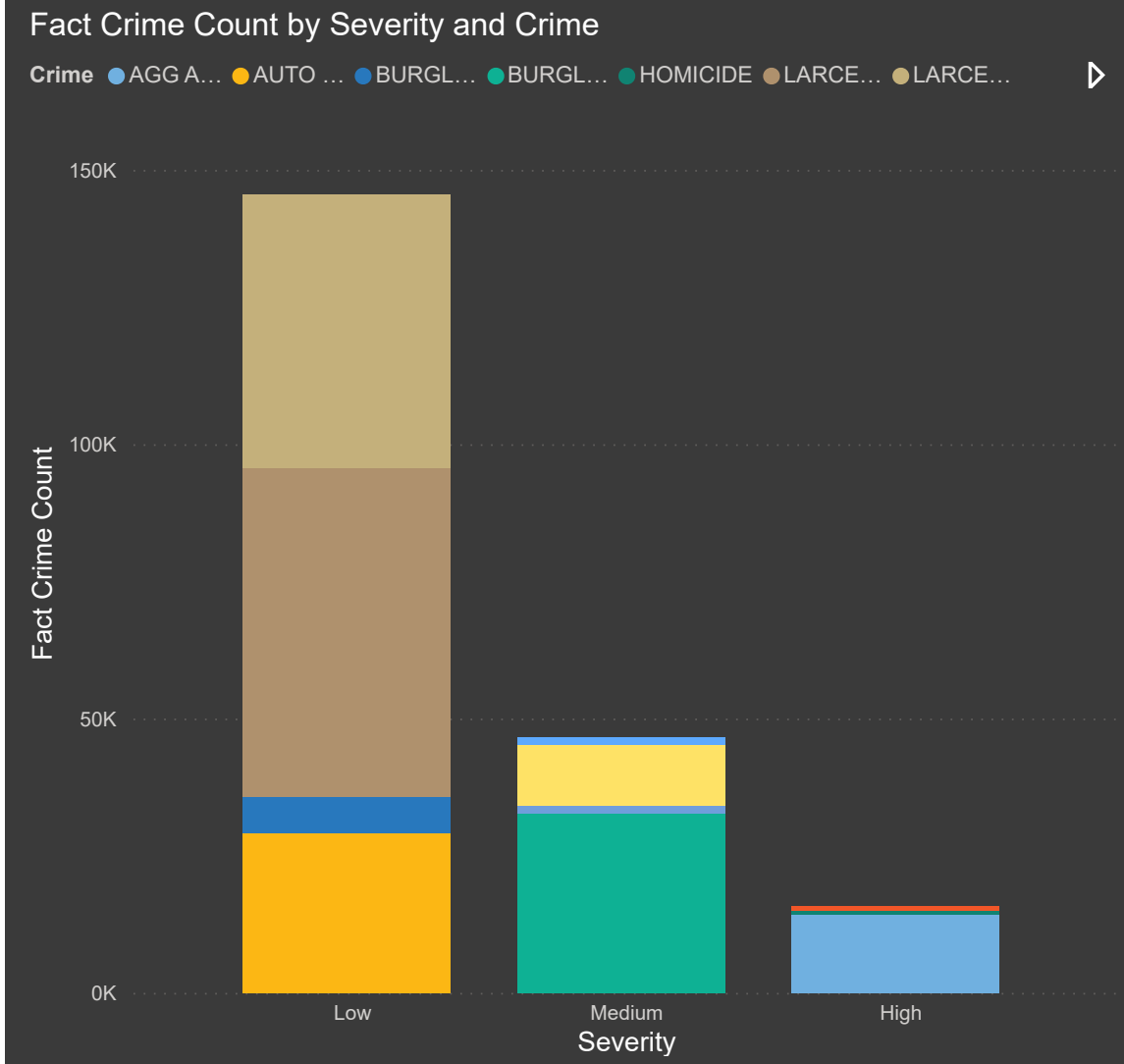
OTHER MISCELLANEOUS FUNCTIONS

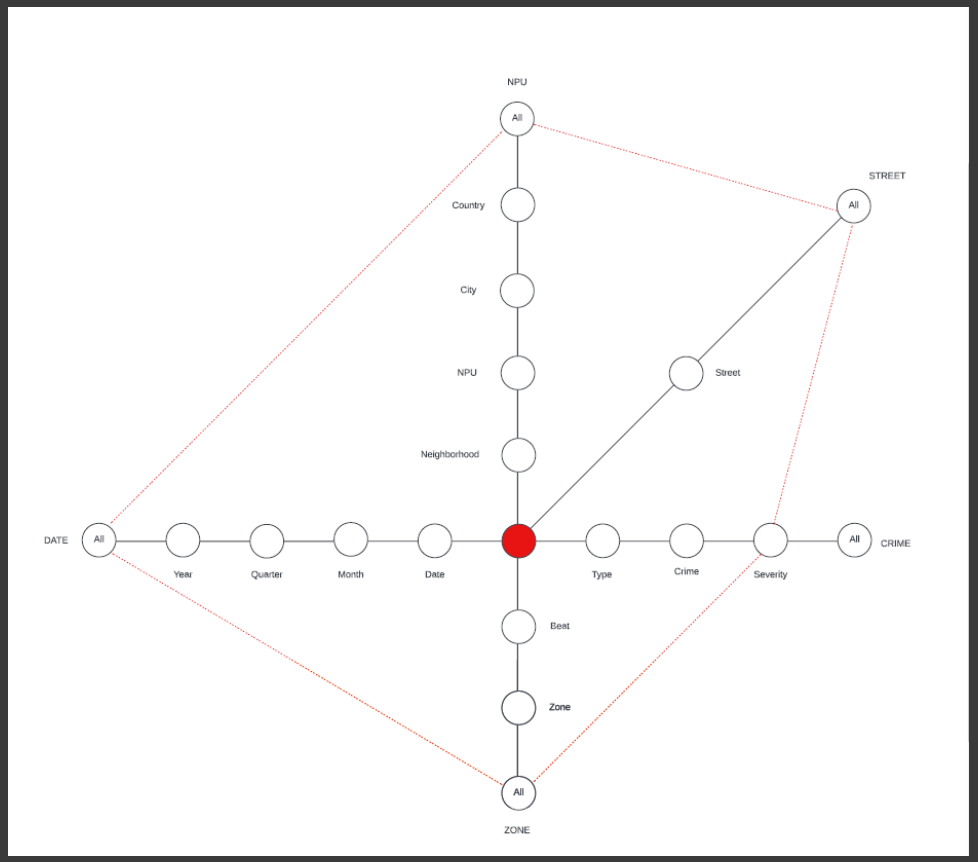
```python
def remove_csv_headers(file_path):
    """
    Removes the headers from a CSV file.

    Args:
        file_path (str): The file path of the CSV file.

    Returns:
        None
    """
    # Read the CSV file into a pandas DataFrame
    df = pd.read_csv(file_path)

    # Write the DataFrame back to the CSV file without headers
    df.to_csv(file_path, index=False, header=False)

    print(f"Headers removed from {file_path}")
```

```python
def determining_severity(data):
    """
    Determine the severity level based on the data in the crime column.

    Args:
        data (str): Data in the second column.

    Returns:
        str: Severity level.
    """
    # Logic for determining severity level
    if data == "HOMICIDE" or data == "RAPE" or data == "AGG ASSAULT":  # Check if data matches high severity crimes
        return "High"
    elif data == "ROBBERY-PEDESTRIAN" or data == "ROBBERY-RESIDENCE" or data == "ROBBERY-COMMERCIAL" or data == "BURGLARY-RESIDENCE":  # Check if data matches medium
        return "Medium"
    else:
        return "Low"  # If data does not match high or medium severity crimes, assign low severity
```

THE 5 BUSINESS QUERIES

Business Query 1: How many low, medium and high crimes are there?



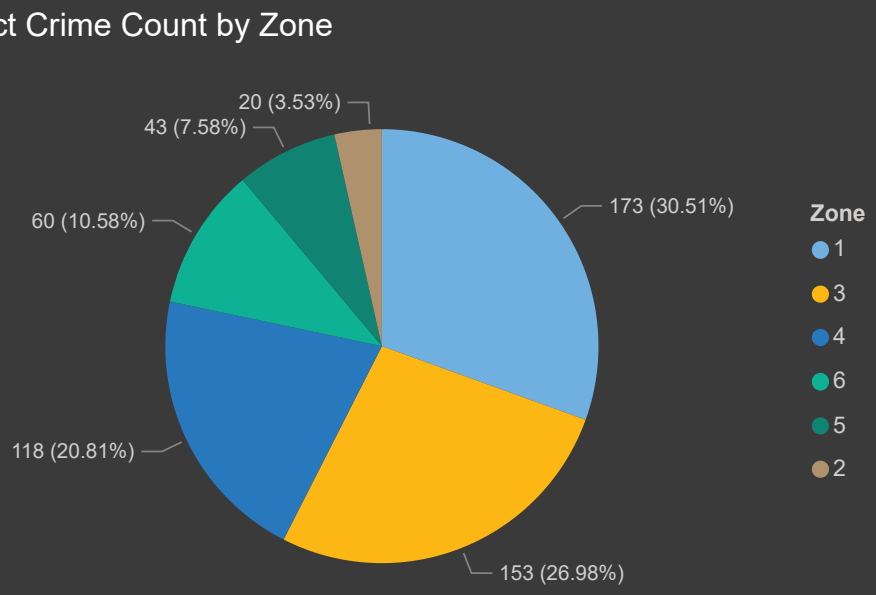


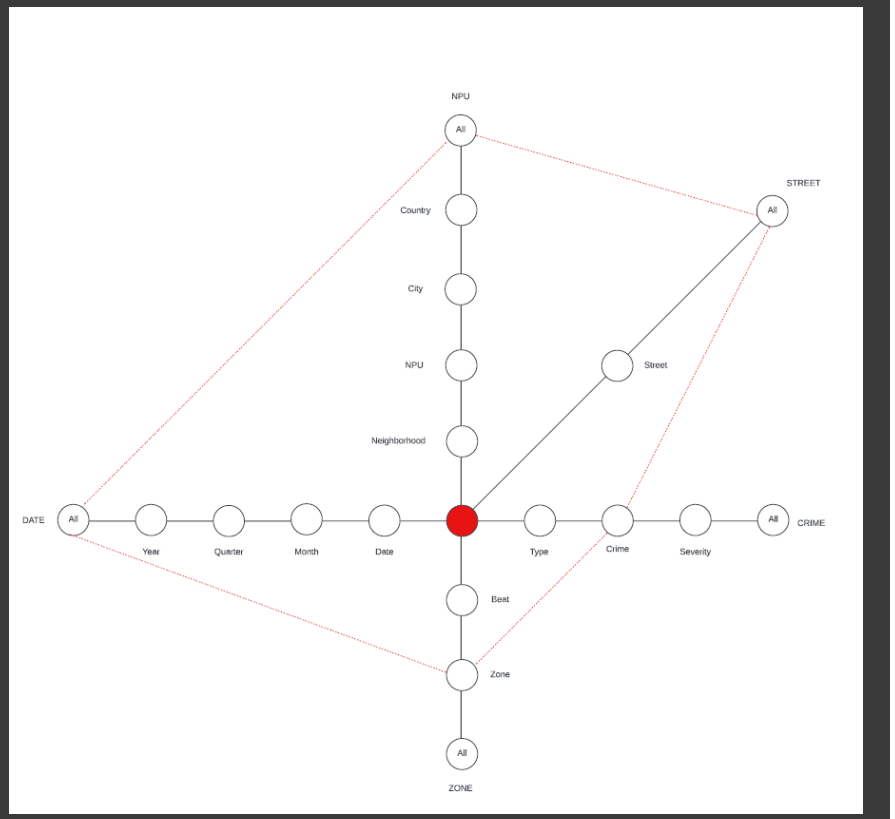
For this business query, the Main dimension that is being utilized is the crime dim table that splits the different crimes that have been committed into their respective severities. Hence the diagram shows the total number of 'LOW', 'MEDIUM' and 'HIGH' crimes that have been committed in the data set that has been provided. By indicating it as a hierarchy with crime as well we can better visualize the different crimes occurrences as well as drilling down to get more specific data if required

Business Query 2: Which Zone has the highest number of Homicides?

Crime

- [ ] AGG ASSAULT
- [ ] AUTO THEFT
- [ ] BURGLARY-NONRES
- [ ] BURGLARY-RESIDENCE
- [x] HOMICIDE
- [ ] LARCENY-FROM VEHI...
- [ ] LARCENY-NON VEHICLE
- [ ] RAPE
- [ ] ROBBERY-COMMERCIAL
- [ ] ROBBERY-PEDESTRIAN
- [ ] ROBBERY-RESIDENCE

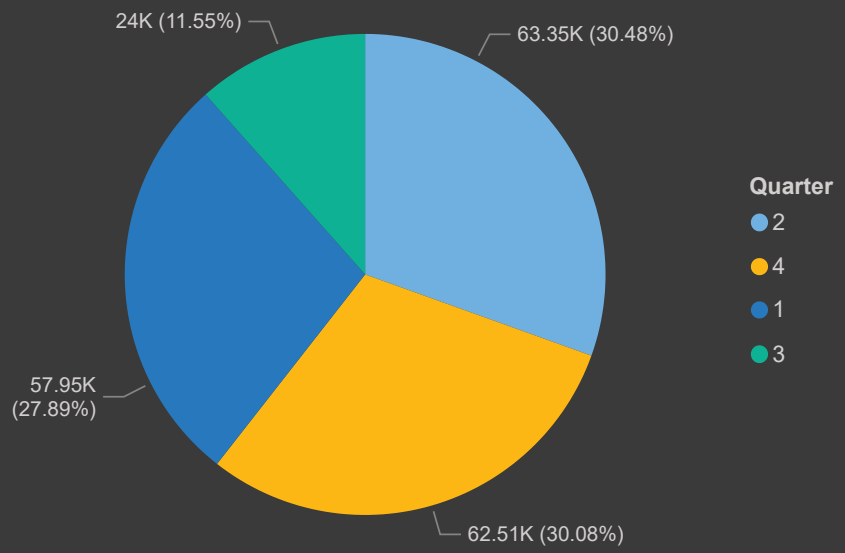


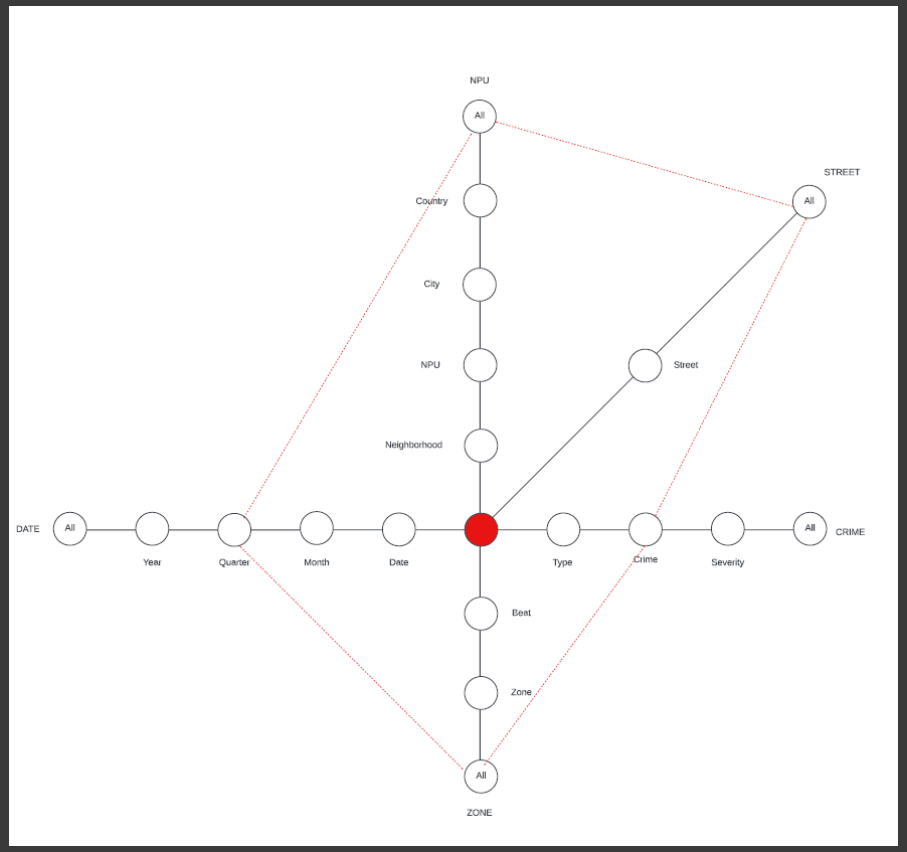


For this query, it expands into using multiple dimension tables. the 2 dimension tables that were utilize to get the results are the Zone and Crime dimension table where the zone was being extracted as well as the crimes committed in their respective zones. a filter has been applied as well to not only allow you to filter homicides but all other crimes should you wish to.

Business Query 3: What is the distribution of crime percentage wise among the 4 quarters?

Fact Crime Count by Quarter





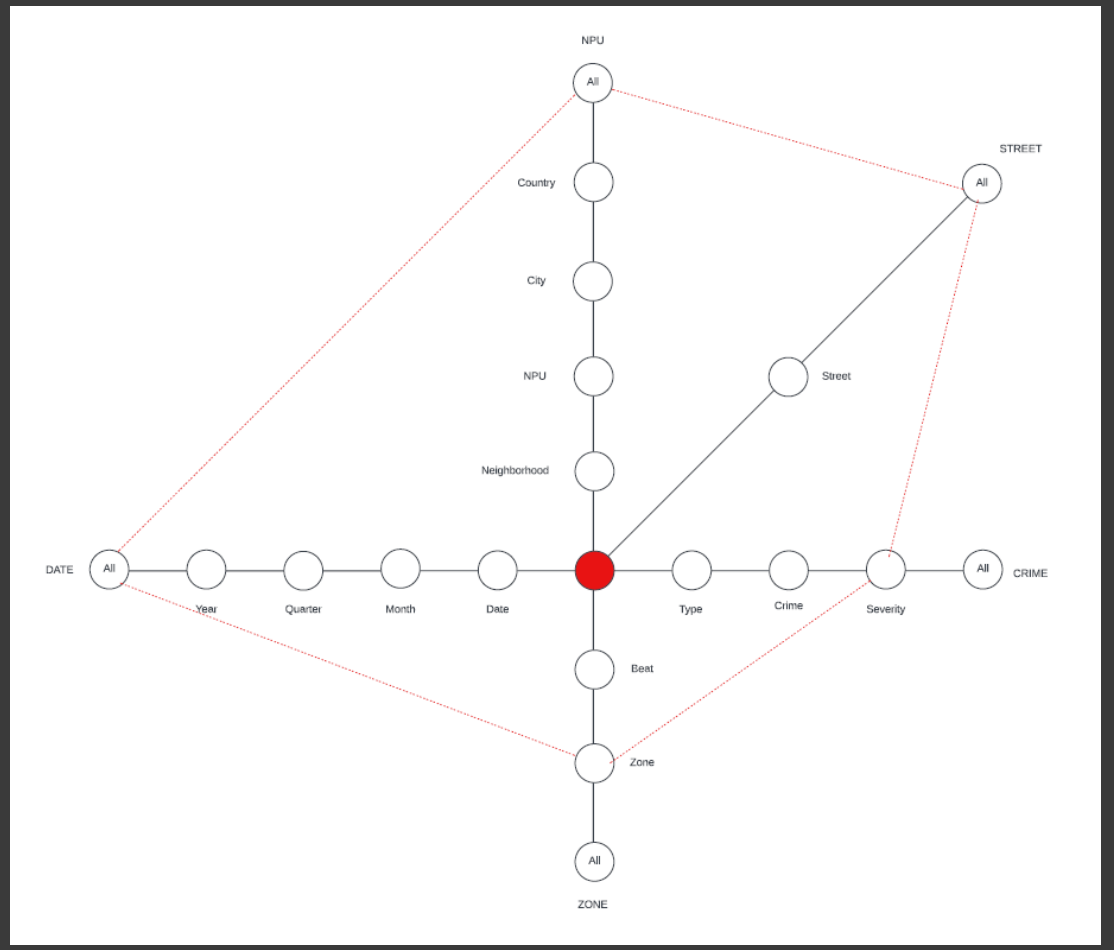
For this query, both the Date and Crime dimension tables were used together to pull the data for all 4 quarters throughout the year. This is the total of each quarter in the whole data set from the year 2009-2017. should you wish to drill down even further to a specific year, you can do that as it is a hierarchical design.

Business Query 4: Of the high crimes, how many crimes (split into crime types) occur in each zone and their respective beats?

Zone
- 1
- 2
- 3
- 4
- 5
- 6
- 7

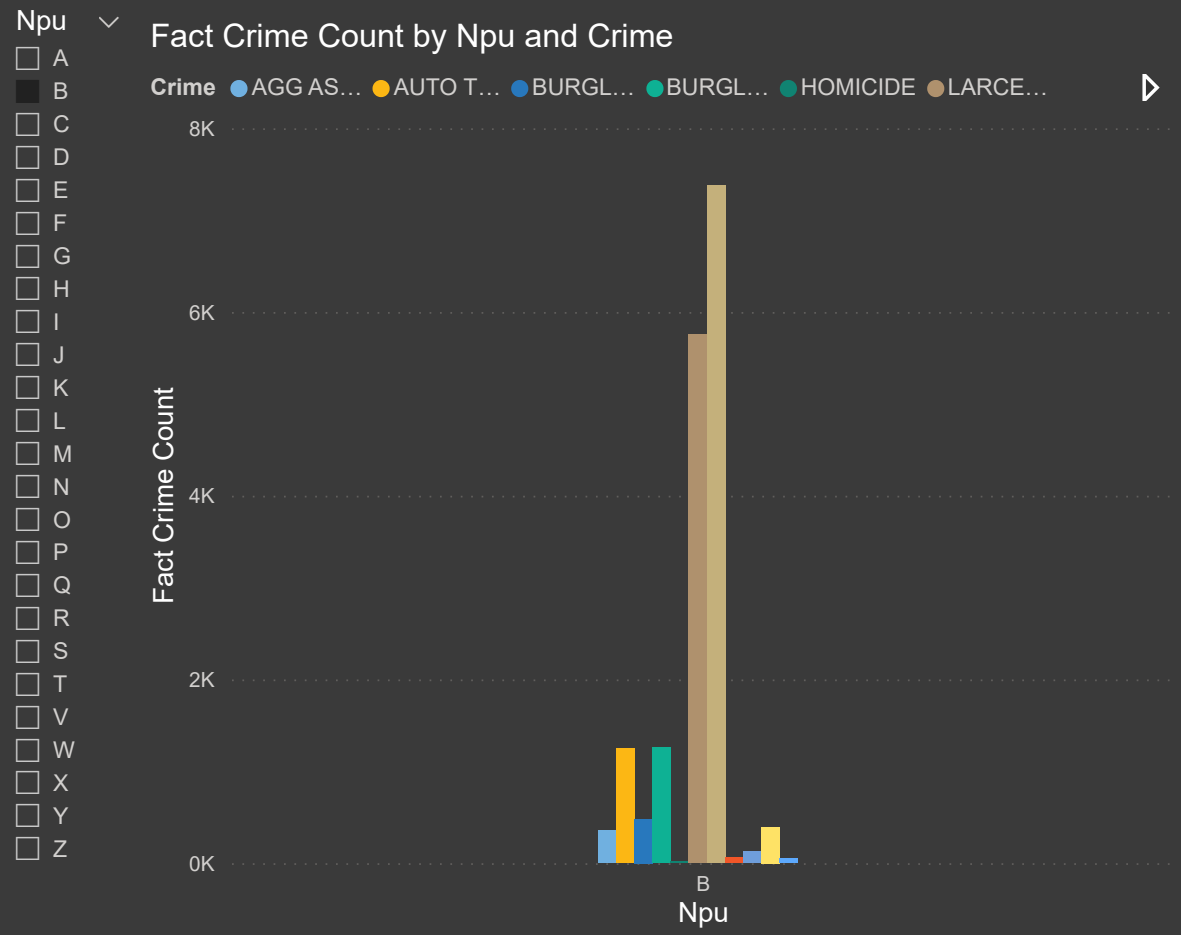
Severity
- High
- Low
- Medium

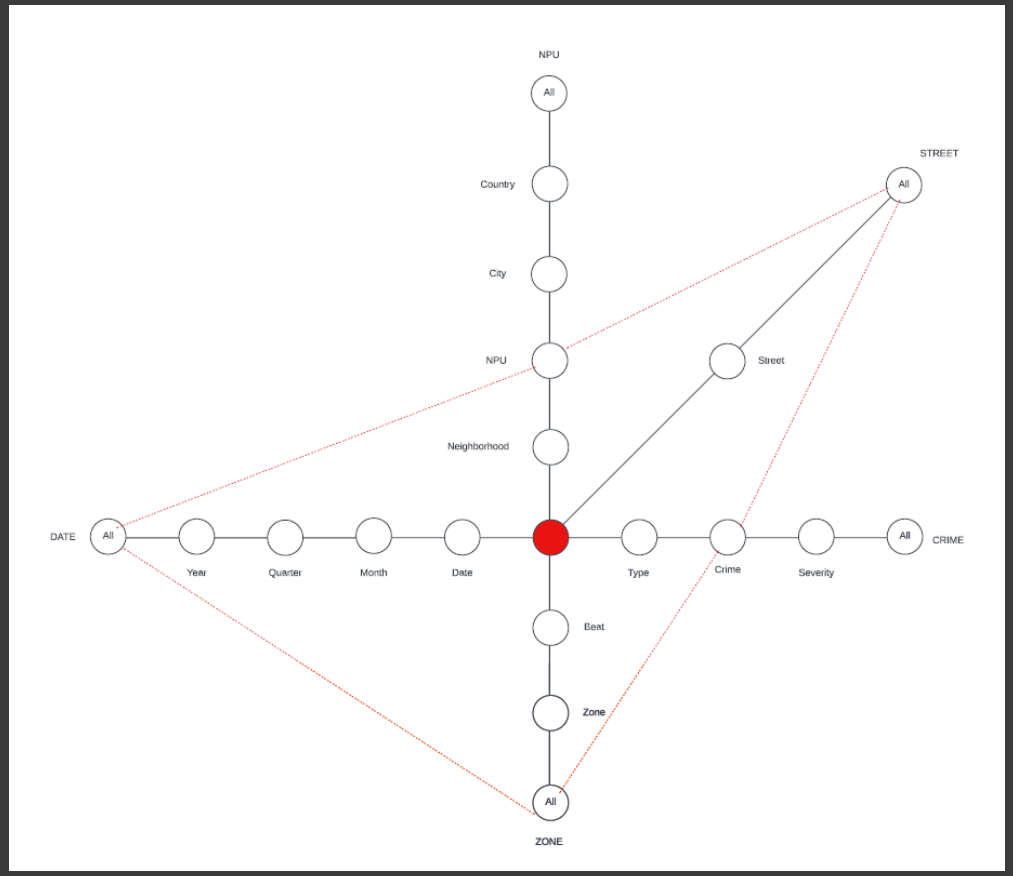
Beat	AGG ASSAULT	HOMICIDE	RAPE	Total
101	117	7	12	**136**
102	273	10	13	**296**
103	401	27	22	**450**
104	298	16	11	**325**
105	389	12	10	**411**
106	322	18	8	**348**
107	280	18	11	**309**
108	283	7	10	**300**
109	243	15	11	**269**
110	276	7	10	**293**
111	137	5	4	**146**
112	209	8	12	**229**
113	332	12	11	**355**
114	183	11	12	**206**
Total	**3743**	**173**	**157**	**4073**



This query uses a matrix to display the desired data as it gives clarity to which crime in which zone and their total count. It further goes into each beat in that zone as well. This is accomplish by establishing 2 filters one for severity of crime hence to minimize the matrix not to display every single crime should you wish not to (although its good to note that it is possible to do that), as well as a filter for zone hence to show all the different beat's data in that specific zone.

Business Query 5: For each NPU, what is the distribution of crimes they handle?

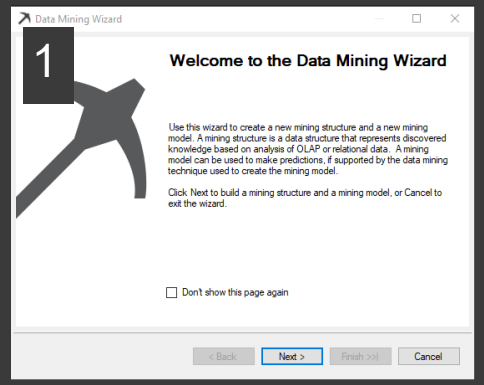


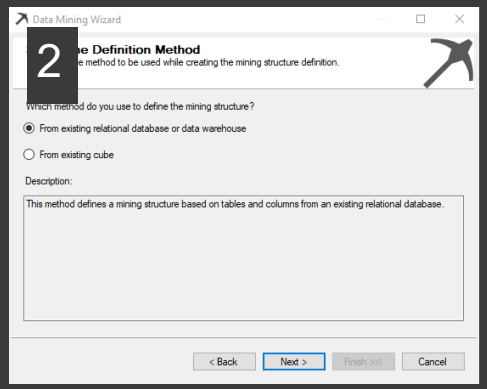


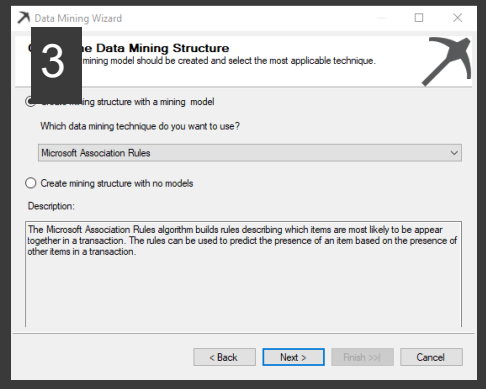
This query is one of the more complex queries as it requires the data to be pulled from very specific sources and demonstrates what a hierarchical design can do. This query pulls data that links all the crime that each NPU has handled and breaks them down into what type of crime they are and what are their exact numbers. The filter at the side is used to specify a specific NPU however you can remove the filter and it will show you all the NPU data.

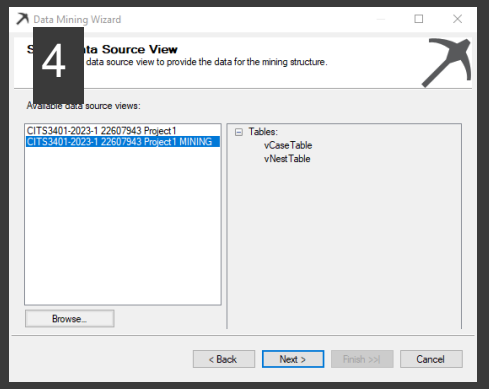
ASSOCIATION RULE MINING

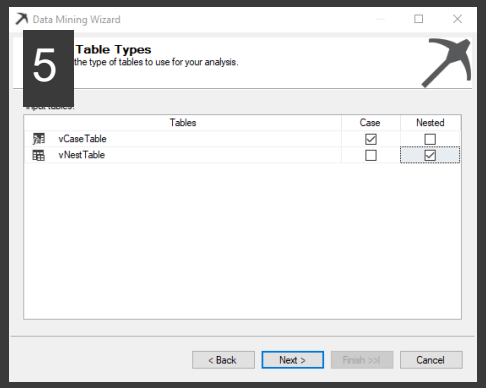
RULE MINING PROCESS

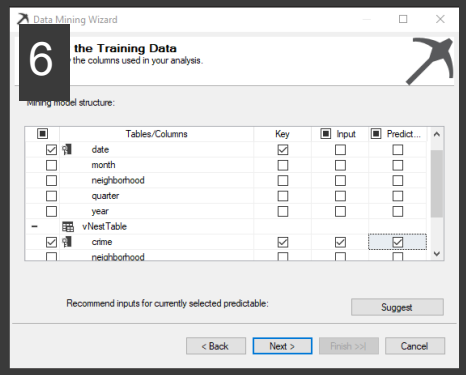


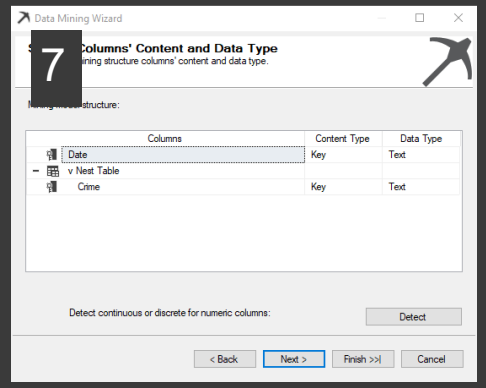


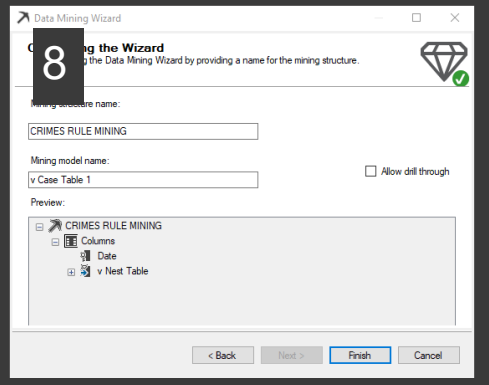


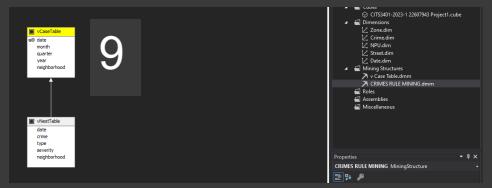












The screenshots shows the process and steps that were taken to create the rule association mining. The key thing to note is step 5. The configured names of the 2 view tables are fairly straight forward hence a mistake is less likely to happen. but in a situation where the tables have different name it is imperative that the correct case and nested tables are selected to avoid wrong data being mined.

CASE AND NESTED VIEW TABLES

To explore the occurrence of unique combinations of crimes on specific dates, I formulated the statement "How many unique crimes happen in combination with each other on a specific date" as the basis of our data mining analysis. Using this statement, we designed a Case table and nested tables with the date as they key. The nested tables were populated with all the unique crimes committed on each date from 2009-2017. The Case table holds the unique dates, year, quarter, and month, while the nested tables hold the unique crimes that occurred on each day.

The main difference between the Case and nested tables is that multiple crimes can happen on the same day, resulting in a repetition of dates with different crimes in the nested table. In contrast, the Case table only has unique date values, and no repetition of any sorts.

Overall, our approach allows us to investigate the occurrence of unique combinations of crimes on specific dates by analyzing the data in the nested tables. We can then use the information gathered from this analysis to identify patterns, correlations, and trends in the data, which can be used to inform decision-making and improve processes.

	date	crime	type	severity	neighborhood
1	2/01/2009	BURGLARY-RESIDENCE	house_number	Medium	Capitol View
2	6/01/2009	BURGLARY-RESIDENCE	house_number	Medium	Capitol View
3	9/01/2009	BURGLARY-RESIDENCE	house_number	Medium	Capitol View
4	13/01/2009	ROBBERY-RESIDENCE	house_number	Medium	Adair Park
5	14/01/2009	LARCENY-NON VEHICLE	house_number	Low	Adair Park
6	26/01/2009	RAPE	house_number	High	Adair Park
7	13/02/2009	LARCENY-NON VEHICLE	house_number	Low	Adair Park
8	15/02/2009	AUTO THEFT	house_number	Low	Adair Park
9	15/02/2009	AUTO THEFT	house_number	Low	Capitol View
10	9/03/2009	AUTO THEFT	house_number	Low	Capitol View
11	13/03/2009	BURGLARY-RESIDENCE	house_number	Medium	Capitol View
12	29/03/2009	BURGLARY-RESIDENCE	house_number	Medium	Adair Park
13	3/04/2009	AGG ASSAULT	house_number	High	Capitol View
14	26/04/2009	AGG ASSAULT	house_number	High	Adair Park
15	2/05/2009	AUTO THEFT	house_number	Low	Adair Park
16	15/05/2009	BURGLARY-RESIDENCE	house_number	Medium	Capitol View
17	23/05/2009	AUTO THEFT	house_number	Low	Capitol View
18	23/05/2009	RAPE	house_number	High	Adair Park
19	30/05/2009	LARCENY-NON VEHICLE	house_number	Low	Adair Park

Nested

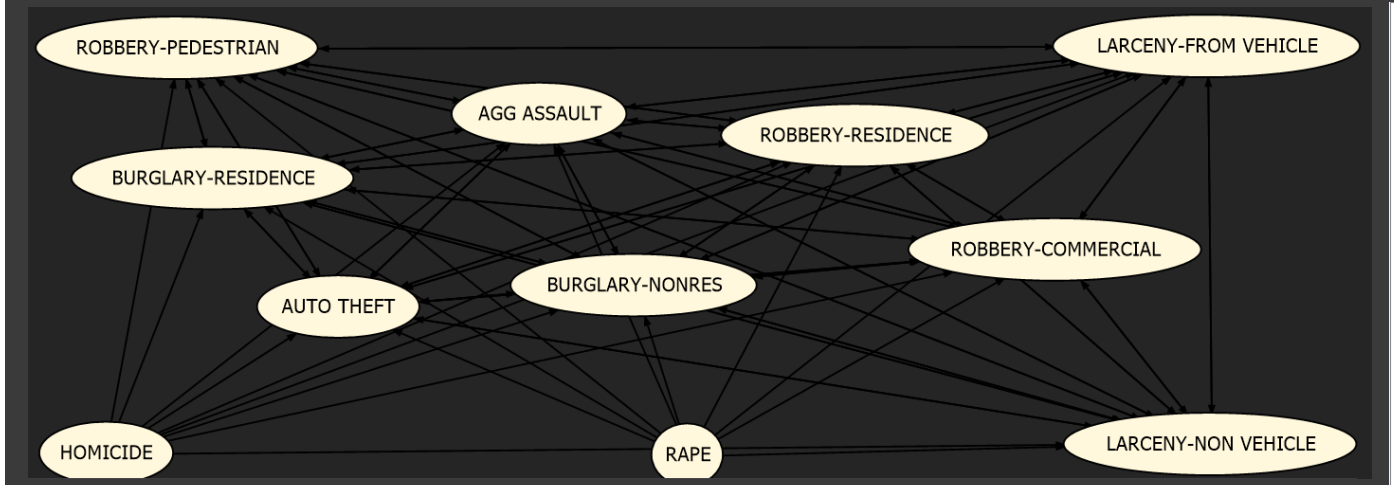
	date	month	quarter	year	neighborhood
1	2/01/2009	January	1	2009	Capitol View
2	6/01/2009	January	1	2009	Capitol View
3	9/01/2009	January	1	2009	Capitol View
4	13/01/2009	January	1	2009	Adair Park
5	14/01/2009	January	1	2009	Adair Park
6	26/01/2009	January	1	2009	Adair Park
7	13/02/2009	February	1	2009	Adair Park
8	15/02/2009	February	1	2009	Adair Park
9	15/02/2009	February	1	2009	Capitol View
10	9/03/2009	March	1	2009	Capitol View
11	13/03/2009	March	1	2009	Capitol View
12	29/03/2009	March	1	2009	Adair Park
13	3/04/2009	April	2	2009	Capitol View
14	26/04/2009	April	2	2009	Adair Park
15	2/05/2009	May	2	2009	Adair Park
16	15/05/2009	May	2	2009	Capitol View
17	23/05/2009	May	2	2009	Capitol View
18	23/05/2009	May	2	2009	Adair Park
19	30/05/2009	May	2	2009	Adair Park

Case Table

```
USE [CITS3401-2023-1_22607943_Project1]
GO

/****** Object:  View [dbo].[vNestTable]    Script Date: 23/04/2023 8:36:57 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

 -- creating views for association rule mining

CREATE VIEW [dbo].[vNestTable] AS
SELECT DISTINCT DATE.[date], CRIME.[crime], CRIME.[type], CRIME.[severity], NPU.neighborhood
FROM [dbo].[FactCrime] AS FACT
LEFT JOIN [dbo].[DimDate] AS DATE ON FACT.DATE_ID = DATE.DATE_ID
LEFT JOIN [dbo].[DimCrime] AS CRIME ON FACT.CRIME_ID = CRIME.CRIME_ID
LEFT JOIN [dbo].[DimNPU] AS NPU ON FACT.NPU_ID = NPU.NPU_ID


GO
```

```
USE [CITS3401-2023-1_22607943_Project1]
GO

/****** Object:  View [dbo].[vCaseTable]    Script Date: 23/04/2023 8:36:23 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

 -- creating views for association rule mining
CREATE VIEW [dbo].[vCaseTable] AS
SELECT DISTINCT DATE.[date], DATE.[month], DATE.[quarter], DATE.[year], NPU.neighborhood
FROM [dbo].[FactCrime] AS FACT
LEFT JOIN [dbo].[DimDate] AS DATE ON FACT.DATE_ID = DATE.DATE_ID
LEFT JOIN [dbo].[DimNPU] AS NPU ON FACT.NPU_ID = NPU.NPU_ID
GO
```

Rules and Dependency network



Pr...	Importance	Rule
0.998	0.476	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.476	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.476	AUTO THEFT = Existing -> BURGLARY-RESIDENCE = Existing
0.993	0.474	AUTO THEFT = Existing -> AGG ASSAULT = Existing
0.993	0.474	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> AGG ASSAULT = Existing
0.993	0.474	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> AGG ASSAULT = Existing
0.884	0.423	AUTO THEFT = Existing -> BURGLARY-NONRES = Existing
0.884	0.423	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> BURGLARY-NONRES = Existing
0.884	0.423	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> BURGLARY-NONRES = Existing
0.425	0.407	BURGLARY-RESIDENCE = Existing -> ROBBERY-RESIDENCE = Existing
0.425	0.407	BURGLARY-RESIDENCE = Existing, LARCENY-FROM VEHICLE = Existing -> ROBBERY-RESIDENCE = Existing
0.425	0.407	BURGLARY-RESIDENCE = Existing, AUTO THEFT = Existing -> ROBBERY-RESIDENCE = Existing
0.425	0.407	BURGLARY-RESIDENCE = Existing, LARCENY-NON VEHICLE = Existing -> ROBBERY-RESIDENCE = Existing
1.000	0.301	LARCENY-NON VEHICLE = Existing -> LARCENY-FROM VEHICLE = Existing
1.000	0.301	LARCENY-FROM VEHICLE = Existing -> LARCENY-NON VEHICLE = Existing
0.999	0.301	LARCENY-FROM VEHICLE = Existing -> AUTO THEFT = Existing
0.999	0.301	LARCENY-NON VEHICLE = Existing -> AUTO THEFT = Existing
0.999	0.301	LARCENY-NON VEHICLE = Existing, LARCENY-FROM VEHICLE = Existing -> AUTO THEFT = Existing
0.998	0.300	LARCENY-FROM VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.300	LARCENY-NON VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.300	LARCENY-NON VEHICLE = Existing, LARCENY-FROM VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.993	0.298	LARCENY-FROM VEHICLE = Existing -> AGG ASSAULT = Existing
0.993	0.298	LARCENY-NON VEHICLE = Existing, LARCENY-FROM VEHICLE = Existing -> AGG ASSAULT = Existing
0.993	0.298	LARCENY-NON VEHICLE = Existing -> AGG ASSAULT = Existing
0.981	0.293	LARCENY-FROM VEHICLE = Existing -> ROBBERY-PEDESTRIAN = Existing
0.981	0.293	LARCENY-NON VEHICLE = Existing, LARCENY-FROM VEHICLE = Existing -> ROBBERY-PEDESTRIAN = Existing
0.981	0.293	LARCENY-NON VEHICLE = Existing -> ROBBERY-PEDESTRIAN = Existing
0.884	0.247	LARCENY-FROM VEHICLE = Existing -> BURGLARY-NONRES = Existing

To generate rules based on the occurrence of crimes, we can use a data mining algorithm to analyze the data in the nested tables. The rules generated by this algorithm would be of the form "IF condition THEN crime type", where the condition is a combination of attribute-value pairs, and the crime type is the predicted output value.

The top k rules, according to importance or probability, would be those that are most likely to accurately predict the occurrence of a specific crime based on the values of the attributes in the condition. The value of k would depend on the specific context of the analysis and the desired level of granularity.

For example, if we wanted to identify the top 10 rules for predicting the occurrence of homicides, we would look for the rues that have the highest accuracy or probability of predicting homicides based on the values of the attributes in the condition. These rules would be ranked in order of importance or probability, with the most relavant or probable rules placed at the top of the list.

Once the top k rules have been identified, they can be used to analyze the occurrence of crimes on specific dates by applying the rules to the data in the nested tables. This can help identify patterns and correlations in the data, which can be used to inform decision-making and improve processes related to crime prevention and law enforcement.

in summary, the top k rules based on importance or probability are those that are most likely to accurately predict the occurrence of a speciic crime based on the values of the attributes in the condition. These rules can be used to analyze the data in the nested tables and identify patterns and correlations related to crime.

TOP 10 RULES BASED ON IMPORTANCE

Pr...	Importance	Rule
0.998	0.476	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.476	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> BURGLARY-RESIDENCE = Existing
0.998	0.476	AUTO THEFT = Existing -> BURGLARY-RESIDENCE = Existing
0.993	0.474	AUTO THEFT = Existing -> AGG ASSAULT = Existing
0.993	0.474	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> AGG ASSAULT = Existing
0.993	0.474	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> AGG ASSAULT = Existing
0.884	0.423	AUTO THEFT = Existing -> BURGLARY-NONRES = Existing
0.884	0.423	AUTO THEFT = Existing, LARCENY-NON VEHICLE = Existing -> BURGLARY-NONRES = Existing
0.884	0.423	AUTO THEFT = Existing, LARCENY-FROM VEHICLE = Existing -> BURGLARY-NONRES = Existing
0.425	0.407	BURGLARY-RESIDENCE = Existing -> ROBBERY-RESIDENCE = Existing

The explanation will be for the first 2 of the rules as the rest would be repetitive and follow the same concept.

1. If auto theft and larceny-non vehicle happens, burglary of residence is very likely to occur as well
2. if auto theft and larceny-from vehicle happens, burglary of residence is very likely to occur as well

As we can see from the above screenshot, if we take the first entry. based on importance we can derive that if there is an auto theft that happens in combination with larceny of a non vehicle type, the probability that burglary happening in a residence is almost certain based on the given data set. Hence we can than use the importance and probability in unison to predict in the future a high likely hood that if crime A and crime B happens together, crime C will more than likely occur as well

note: due to technical issues I am unable to highlight the dependency network on visual studios due to errors such as "there is not enough disk space" as well as "server is down, please ensure server is switched on". Tried to fix the errors with google solutions as well as documentation but still the error persist. Screenshots has been provided to described the said errors.