UNIVERSITY OF PISA

DEPARTMENT OF INFORMATICS

Master Degree in Data Science and Business Informatics

Laboratory of Data Science project

Matteo Rofrano - Francesco Lanci Lanci

Academic Year 2023/2024

Contents

Chapter 1: First Part

First of all, we have created all the tables and relations between the primary key and foreign key by using SQL server management studio. Then we created 6 different scripts (one for each table) where we extracted and computed the relevant features to be inserted in the relative table on the server and we wrote the SQL query necessary to upload row by row the instances in the database. It is important to note that first of all we created the dimension files and then the last one custody file which uses data obtained from the previous files.

1.1 Gun table

To create the gun table we took the necessary information (is_stolen and gun_type) from police.csv and then created a unique id for each unique pair of is_stolen and gun_type.

1.2 Participant table

To build the participant table we used the same process used for the gun table, so we took the necessary data from police.csv and created a unique id for each unique combination.

1.3 Geography table

The most challenging task to be computed for the Geography table is to get the geographical information from the latitude and longitude features. We have, first of all, checked that all the coordinates refer to American's places so in order to retrieve the city, state, and country by using the latitude and longitude we have used an external csv file named "full_dataset" which contains information about the Canadian and USA cities. We built a KDtree by using the information from the external file and then we looped our police csv row by row, and by using the latitude and longitude coordinates to access the leaf of the tree we retrieved the closest city, state and country associated with those coordinates. In addition to the KDtree approach, we also tried: simply calculating the Euclidean distance between our coordinates and those of cities and also GeoPy to trace the city directly, but both approaches took more hours, while the method with the KDtree only took a few minutes.
To get the continent we have used another library which is "getconti" which takes in input the country and returns the continent and as expected all the continents will be "North America" (we decided to keep the continent even though the values are all the same, because new values may come in the future).

1.4 Date table

To build the date table we first of all parsed the XML file to get the date associated with the "date_fk". Once we got the date information we used the library "datetime" to extract all the information from the date format.

1.5 Incident table

Incident represent a degenerated dimension that we have decided to include in order to have more flexibility since in the future could be useful to add more attributes and details for each incident. To populate this table we have just used the information available in the "police.csv" file.

1.6 Custody table

After having built all other dimensional tables we have built the fact table "custody" by using the ids of the dimensional tables already constructed. Moreover, by loading the different mappings contained in the json files, we have computed the "crime_gravity" feature. The most challenging task we faced building the custody table is to assign the values to the foreign keys to the dimensional tables. To solve this issue we have created a dictionary for each file already created (one for gun, one for date and so on) assigning as key the attribute that are superkey for the table (for example for the geography table 'latitude' and 'longitude') and as value the entire row. Then we scanned row by row the police data and if the column value that represent the superkey of the dimensional tables matched the dimensional tables data we get the corresponding value of the dictionary (which is the row of the dimensional table which include the id too).

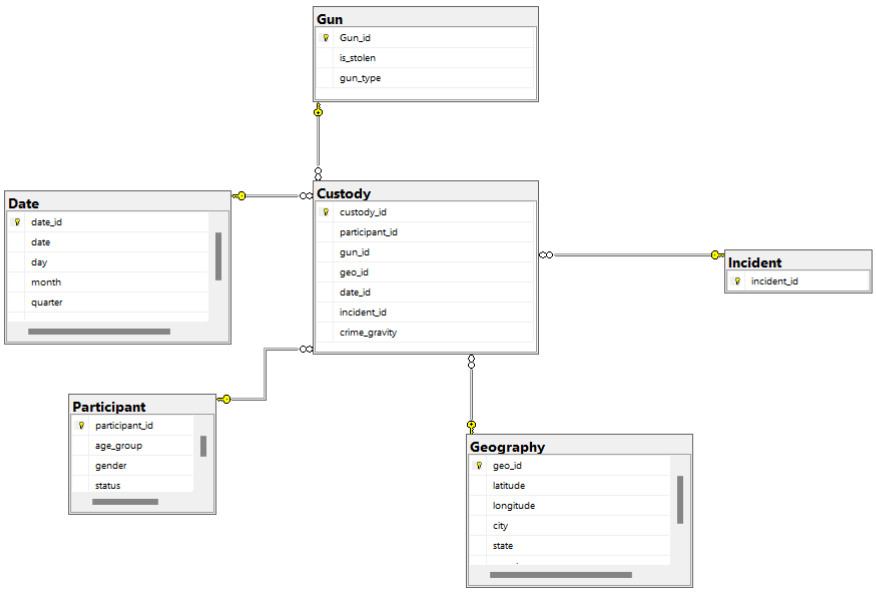

Figure 1.1: Logical Schema

Chapter 2: Second part

2.1 Assignment 0

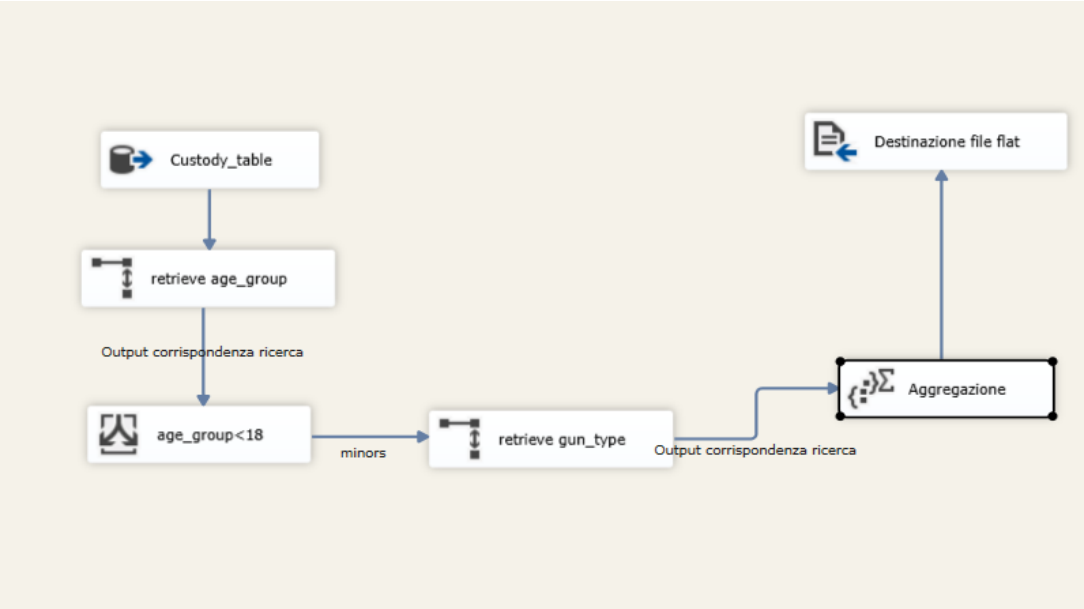

Figure 2.1: Assignment 0 data flow

Since our task for the assignment 0 was "For every type of gun, determine the number of custodies of young participants (under the age of 18)" we started first of all by accessing the data of the custody table. In particular, we took the data from the gun_id, custody_id, and participant_id columns. Then we used the lookup node to take the data from the age_group column (contained in the Participant table) that have a matching partecipant_id.
After this, we have used the conditional split to filter the rows with age_group different than "Adult 18+" (since there are 3 groups, in this way we have taken all the people that are less than 18 years old). Since now we have less data (so less matching foreign key to search) we decided to make another lookup in order to take the corresponding gun_type from the Gun table with the same gun_id of our actual data.
At the end we used the aggregation node to group by gun_type and for each group to have the count of the custody_id. The results are reported in the csv flat file.

2.2 Assignment 1

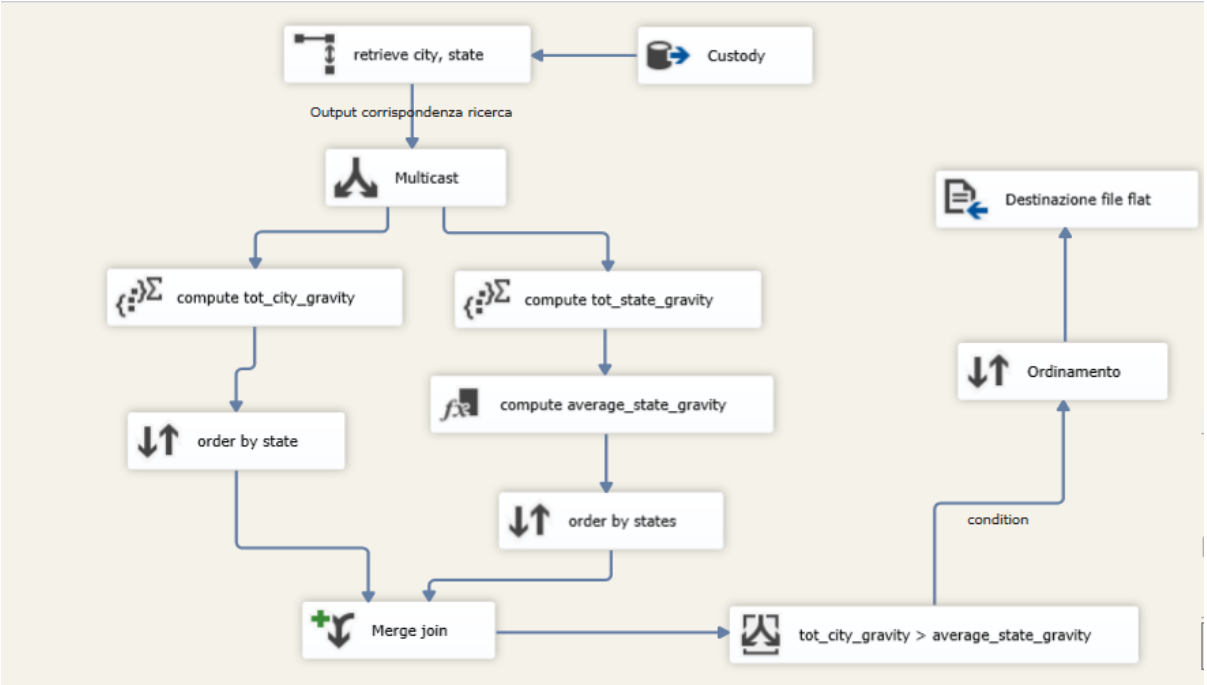

Figure 2.2: Assignment 1 data flow

The second task request is "A city is classified as dangerous if the total gravity of custodies within that city is higher than the average gravity for that city's state. List all the dangerous cities".
First of all, we accessed the custody table to get the data from custody_id, geo_id, gun_id, crime_gravity and then we used the lookup node to get the city and state from the geography table. After this we used the multicast node to create 2 data flows which use the same data.
The one on the left is used to compute the total crime gravity for each city by using an aggregation node. The other data flow on the right is used to compute the total crime gravity for each state and then by using this information we computed the average crime gravity for each state by dividing the total crime gravity of a state by the number of cities of that state.
We have merged these 2 data flows on the state attribute and then we have used a conditional split node to get only those cities with tot_city_gravity greater that the average_state_gravity.

2.3 Assignment 2

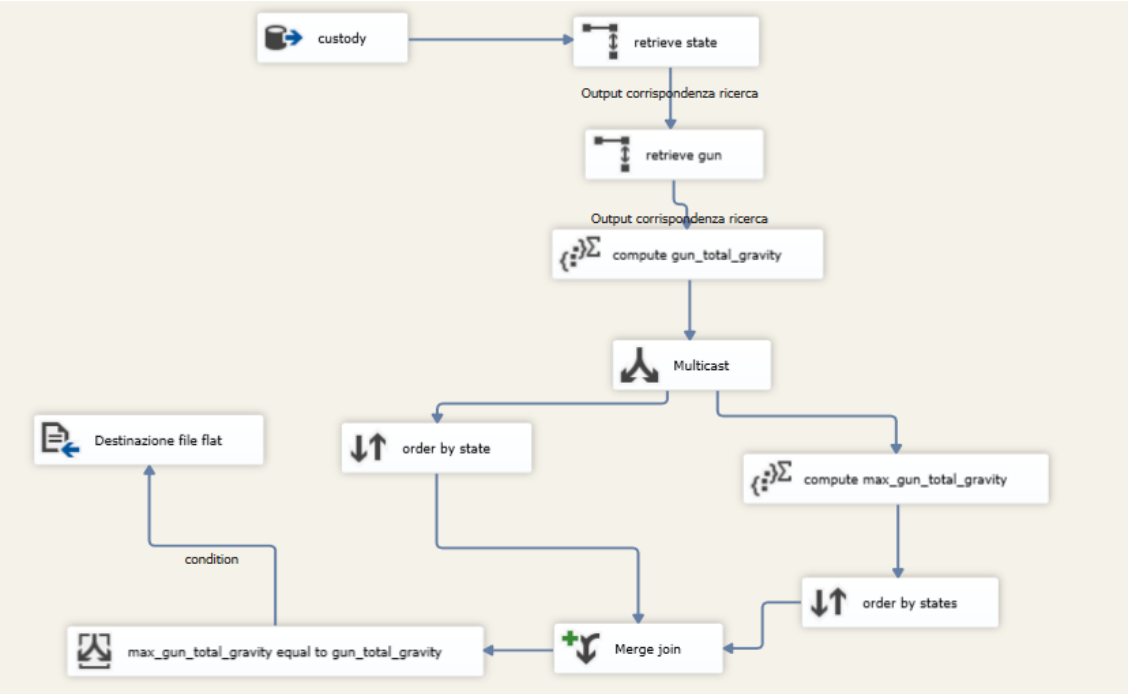

Figure 2.3: Assignment 2 data flow

The final task is the following one "For each state, determine the gun type with the highest total gravity" so, to begin we access gun_id, geo_id, crime_gravity from the custody table and then by using 2 look up nodes we get the state and the gun_type from the geography and gun tables. We group by state and gun_type and we compute the total crime gravity for each gun in each state. Then we use a multicast node.
In the left data flow we do nothing except to order by state while in the right data flow we aggregate again by state and as aggregation function we use the max, so in this way we find the maximum total crime gravity realized by an unknown gun within that state and we store the result in the attribute max_gun_total_gravity. We order by state and then we merge the 2 flows by the state attribute.
At the end we use a conditional split to get those gun_type which have the gun_total_gravity equal to the max_gun_total_gravity. So we get the gun_type that has the total crime_gravity matching the highest total gravity of a certain state. This is needed since in the aggregation before the merge join we lost the information about the gun_type.

Chapter 3: Third part

3.1 Assignment 0

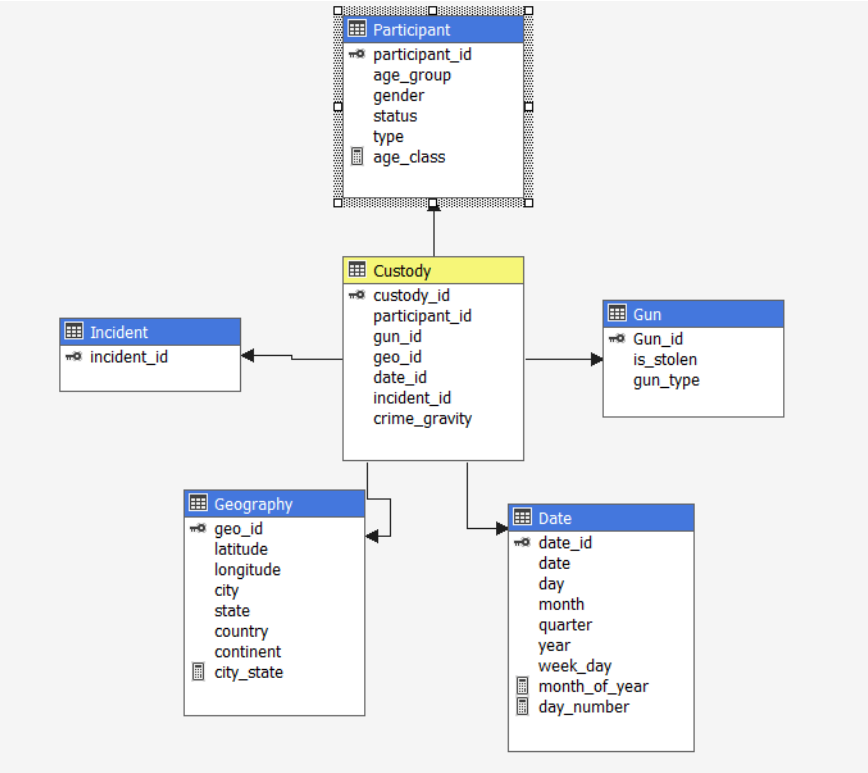

Figure 3.1: Cube structure

To create our cube that we will use for the mdx queries and PowerBI plots we have first of all defined the different dimensions and hierarchies. In particular, for the Geography dimension we have created first the city_state attributes by combining the city attribute value to the state attribute value. We have taken this choice because in our data we have cities with the same name but in different states such as Arlington which is a city of both Virginia and Texas. By using this solution we are able to create a geographical hierarchy of functional dependencies where GEO_id determines city_state which determines state which determines country and so on. For the Date dimension, since our data do not allow us to have functional dependencies between months and years we decided to create a structure where all the attributes are determined by date_id. We have created the month_of_year (which contains the month number) and day_number to sort the attributes month and week_day according to the order of these 2 new column's values.
Since for all the other dimensions, we do not have a structural hierarchy all the attributes are determined by their primary key. To order the age_group attribute based on the age (so first the children, then the teenagers, and ultimately the adults) we have created the column age_class which takes 1 if age_group is equal to "Child 0-12" and so on.

3.2 Assignment 1

```
--1)Show the city with the highest crime gravity for each state.

select [Measures].[Crime Gravity] on columns,
nonempty(generate([Geography].[State].[State],
topcount(
([Geography].[State].currentmember, [Geography].[City State].[City State]),
1,
[Measures].[Crime Gravity]))) on rows
from [Group ID 14 DB]
```

Figure 3.2: MDX assignment 1

To answer the first assignment "Show the city with the highest crime gravity for each state" we have decided to use the generate function which takes as first input the state for which we want to loop for applying the topcount function to identify the city with the highest crime gravity.

3.3 Assignment 2

```
--2)For each state, show the incident with the highest ratio between his total crime gravity and the
--  average crime gravity of that state.

with member crime_average as avg([Incident].[Incident Id].[Incident Id], [Measures].[Crime Gravity])
    member ratio as [Measures].[Crime Gravity]/crime_average
select ratio on columns,
nonempty(generate([Geography].[State].[State],
topcount(
([Geography].[State].currentmember,
[Incident].[Incident Id].[Incident Id]),
1,
[Measures].[Crime Gravity]))) on rows
from [Group ID 14 DB]
```

Figure 3.3: MDX assignment 2

Here, for the assignment "For each state, show the incident with the highest ratio between its total crime gravity and the average crime gravity of that state" we decided to first create 2 members: crime_average as the average crime gravity based on incidents, and ratio as the ratio between crime gravity and crime_average. After that, on the columns, we selected ratio only, based on all nonempty State-Incident_Id pairs with the highest crime gravity (and so to the highest ratio).

3.4 Assignment 3

```
--3)For each city, show the difference between each quarter's total
--  crime gravity and the previous quarter's total crime gravity.

with member diff_prev as
([Date].[TIME_hierarchy].currentmember, [Measures].[Crime Gravity])-
([Date].[TIME_hierarchy].prevmember, [Measures].[Crime Gravity])

select {[Measures].[Crime Gravity], diff_prev} on columns,
nonempty((
      [Geography].[City State].[City State],
      [Date].[Year].[Year],
      [Date].[TIME_hierarchy].[Quarter]
    )) on rows
from [Group ID 14 DB]
```

Figure 3.4: MDX assignment 3

For the assignment "For each city, show the difference between each quarter's total crime gravity and the previous quarter's total crime gravity" we first decided to construct the diff_prev member to calculate the difference in crime gravity between the current and the past member. After that, we put the crime gravity and diff_prev on the columns, based on the nonempty rows city-state, year, and quarter. The year is not strictly necessary, but it is for a better interpretation of the results (quarter 1 and 2 are related only if they are from the same year).

3.5 Assignment 4&5

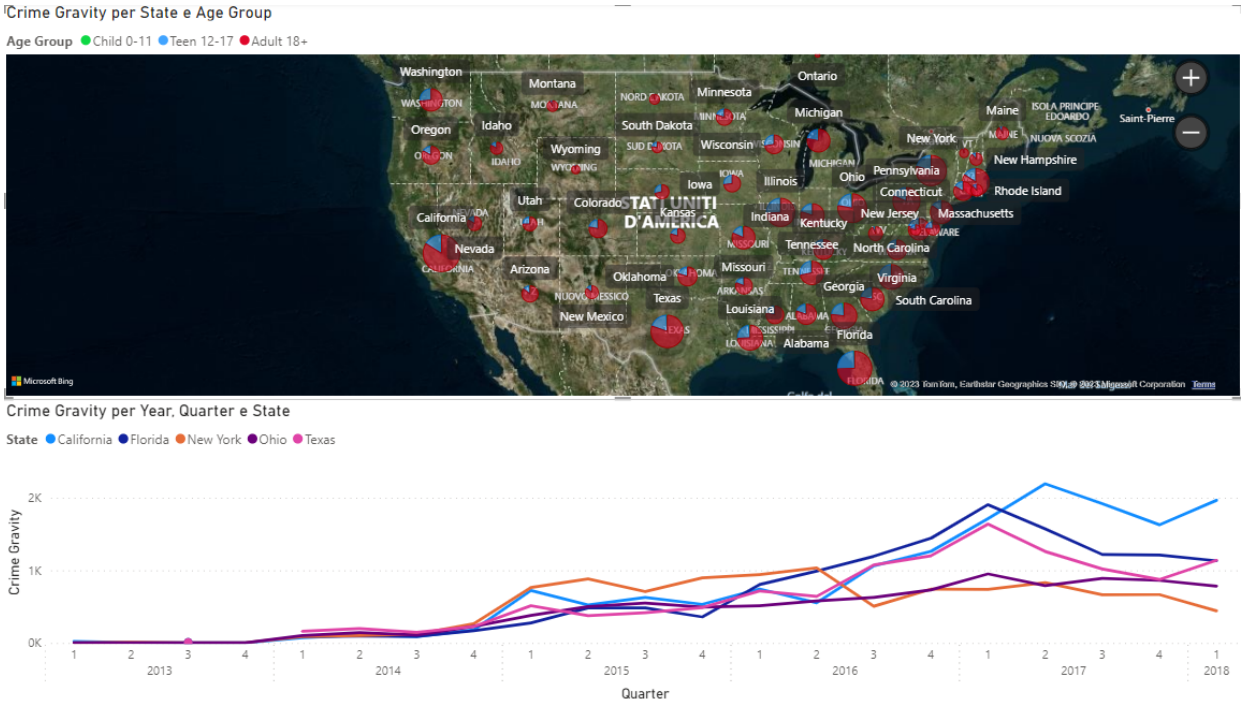

Figure 3.5: Geographical and temporal distributions

From the plot of the geographical distribution of the crime gravity, we can see that each state follows the same distribution, in fact, we can see that most of the custodies are people with more than 18 years. There are some states where almost all the custodies have as subject adults for example the North Dakota and the Montana.
The second plot shows how the total crime gravity of the top 5 states with the highest crime_gravity has changed during the years. For example, we can see a general pattern for which we have a rise in the total crime_gravity during the years with an eadge around the 2017 except for the State of New York which was the most dangerous city until the 2016 and then has decreased its total crime_gravity.
We can even observe how the California has experienced an impressive rise in the total crime_gravity. In fact, from the 2016 and the 2017 it has more than doubled its crime_gravity.