



Police Dataset

Laboratory of Data Science Project

Author: *Bombino Biancamaria 561745, Hakim Sara 597478*

Instructors:	<i>Anna Monreale, Roberto Pellungrini, Cristiano Landi</i>
Master's degree:	Data Science & Business Informatics
Date:	31 dicembre 2023

Indice

1. Creazione del datawarehouse	2
1.1 Assignment 0: Costruzione dello schema del database	2
1.2 Assignment 1: Creazione tabelle	3
1.3 Assignment 2: Popolare il database	4
2. Analisi dati tramite SSIS	5
2.1 Assignment_0	5
2.2 Assignment_1	5
2.3 Assignment_2	6
3. Analisi dati tramite SSAS	7
3.1 Costruzione Datacube	7
3.2 Query MDX	7
3.3 Dashboards	8

1. Creazione del datawarehouse

In questa Sezione sono descritti i passaggi seguiti per costruire e popolare il database sia con i dati già a disposizione, sia con ulteriori attributi ricavati da essi. I file dei risultati ottenuti da questa prima parte sono riportati nella cartella **LDS_Part1_group8**, contenente anche i relativi codici ideati in Python.

1.1 Assignment 0: Costruzione dello schema del database

La parte iniziale del processo riguarda la creazione dello schema del database, raffigurato in figura 1, mediante l'utilizzo del software **SQL Server Management Studio**. Durante questa fase, sono state effettuate alcune scelte per definire la tipologia dei dati in corrispondenza dei vari attributi. In particolare, per gli attributi di tipo stringa con lunghezza variabile è stato scelto il tipo **nvarchar(50)**. Quest'ultimo è preferibile poiché permette di memorizzare qualsiasi dato UNICODE nelle colonne e gestisce dati con differenti lunghezze, a differenza del **varchar**. Per quanto riguarda le variabili *latitude* e *longitude*, presenti in *Geography*, esse sono state catalogate come **float**. L'attributo *date* della dimensione *Date* è stato invece identificato con il tipo **date**. Infine, a tutte le altre variabili numeriche è stato associato il tipo **int**. E' possibile notare come nella figura 1 non sia presente la dimensione **Incident**. Questa è una conseguenza della decisione di trattare questa tabella come una **Degenerate Dimension**, poiché non contiene ulteriori attributi oltre la chiave primaria. La sua inclusione nel database come una tabella degenerata semplifica il modello, consentendo una organizzazione più pulita dei dati.

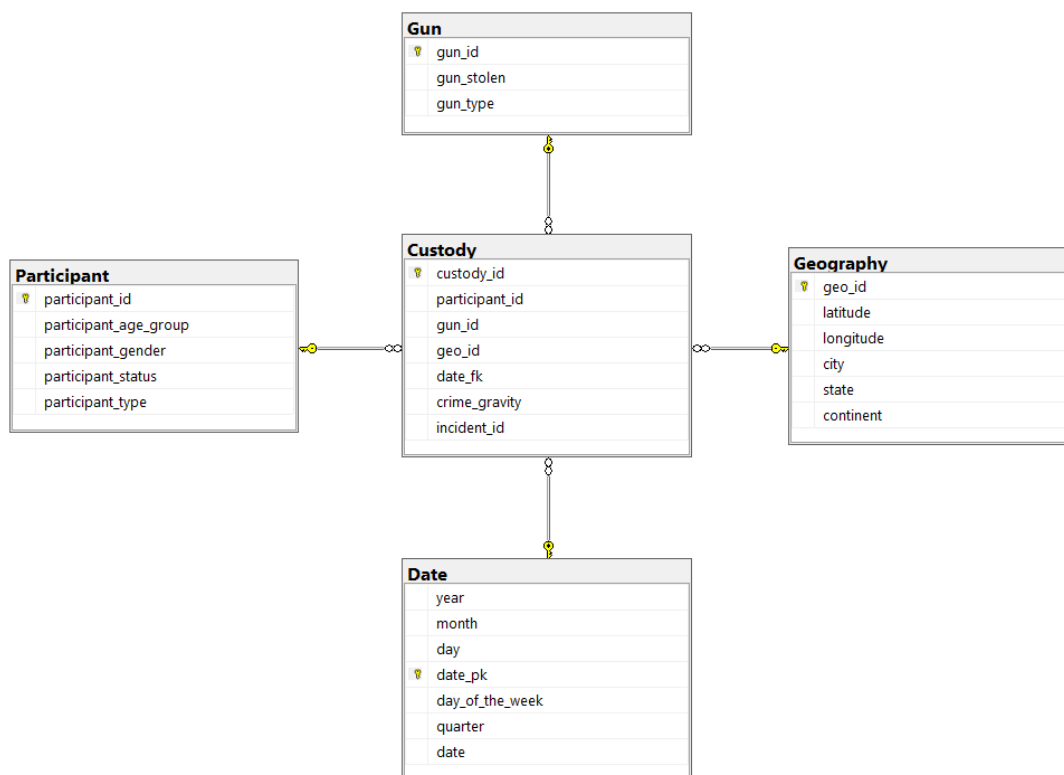


Figura 1: Database Schema

1.2 Assignment 1: Generazione tabelle

In questa Sezione è delineato il procedimento impiegato per la creazione delle tabelle presenti nello schema in figura 1 e suddivise in cinque file CSV differenti.

1.2.1 Geolocalizzazione Incidenti

In questa parte è descritto il ragionamento effettuato per ottenere la città, lo stato e il continente in cui si è verificato l'incidente. Tali deduzioni sono state condotte utilizzando le coordinate geografiche relative agli incidenti nel file **Police.csv**, rappresentati da un identificativo unico *incident_id*; e usufruendo di un dataset geografico esterno denominato **new_uscities.csv**¹. Quest'ultimo include dettagli quali la latitudine, la longitudine, la città, lo stato e il continente di vari territori americani. Per tale motivo è stato considerato il più adatto, dal momento che il file **Police.csv** documenta esclusivamente sparatorie avvenute negli Stati Uniti. L'utilizzo di questo dataset ha consentito di identificare con maggiore precisione e in una maniera computazionalmente efficiente le varie località degli incidenti. Il codice ideato è riportato nel file **geolocalizzazione.incidenti.py** e utilizza la libreria **Geopy**. Quest'ultima è valida poiché contiene la funzione geodetica per calcolare la distanza tra due punti sulla Terra, in base alle loro coordinate geografiche. Inizialmente è effettuata la lettura del dataset geografico, i cui dati vengono salvati in un dizionario. La chiave di quest'ultimo rappresenta la coppia del gruppo di coordinate, e i suoi valori sono le informazioni della città corrispondente. Questi gruppi di coordinate servono per rendere il codice più efficiente a livello computazionale, e sono stati definiti tramite due variabili a cui sono assegnate le parti "intere" della longitudine e latitudine delle città. Questi valori sono poi impiegati per determinare a quale gruppo di coordinate appartiene l'incidente considerato. E' importante sottolineare che, per semplicità, ogni gruppo di coordinate è associato alla prima città trovata nel file geografico che condivide le stesse coordinate. In seguito viene aperto il file **Police.csv** in modalità lettura, e per ogni sua riga vengono memorizzate le parti intere della latitudine e longitudine corrente. Per ogni coppia di queste coordinate viene controllato qual è il gruppo ad esse più vicino, riducendo la ricerca ai gruppi di coordinate che si differenziano di un solo grado. Di conseguenza viene calcolata la distanza attraverso la funzione **calcola_distanza**. Se quest'ultima ritorna un numero inferiore alla distanza massima che inizialmente è impostata a 100, le informazioni della città vengono salvate nella variabile **closest_city**. In conclusione, questi dati vengono registrati, insieme alle altre colonne del file **Police.csv**, in un nuovo file **police_with_geoinfo.csv**. In questo modo è possibile ottenere un'associazione geografica dettagliata per ogni incidente del dataset originale, senza una perdita eccessiva della precisione che si sarebbe ottenuta confrontando le coordinate dell'incidente con ogni singola latitudine e longitudine associata ai dati geografici esterni.

1.2.2 Tabelle

Dates

Inizialmente è stato eseguito **split.dates.py**, in cui è presente una prima funzione **conversion** utilizzata per convertire il file originale dal formato XML al CSV. Successivamente viene descritto il metodo per dividere la data dal formato completo in **mese, giorno e anno**. In questo metodo per ogni riga viene adoperata la **split** e viene creata una nuova riga con i dati suddivisi e i dati delle colonne restanti già presenti nel file. Queste righe sono poi riportate nel nuovo file. Infine viene dedotto il giorno della settimana, utilizzando **datetime, date** e **strftime** per ottenerlo nel formato di una stringa. Per trovare il quarto dell'anno è stata invece utilizzata la formula **"numero_mese -1 // 3 +1"**, in cui:

- si sottrae uno dal mese per ottenere un indice a base zero

¹<https://simplemaps.com/data/us-cities>

- si esegue una divisione per tre raggruppando i mesi in trimestri
- si aggiunge uno al risultato per spostare i trimestri da un indice a base zero a un indice a base uno.

Tutte queste informazioni relative alle date sono state salvate nel file **Date.csv**, unite alla chiave primaria già presente nel file originale.

Participan, Gun & Geography

Per quanto riguarda le tabelle *Participant*, *Gun* e *Geography* è stato eseguito un unico programma, **files_table.py**, in cui sono presenti due funzioni ripetute e adattate in base alle esigenze di ogni tabella. Nel primo metodo sono stati creati i file **Participant.csv**, **Gun.csv** e **Geography.csv**, inserendo i record univoci presenti in **police_with_geoinfo.csv**. L'unicità del record è considerata tale in base ai valori assunti dagli attributi presenti per ogni tabella nello schema 1. Per assicurare la scrittura di soli record differenti all'interno dei file, la struttura dati adoperata è il **set**. Quest'ultimo è noto poiché può contenere solo valori unici, disposti in ordine casuale, e ad esso sono aggiunti solo quei record che si differenziano per gli attributi considerati della tabella corrente. Il secondo metodo invece è utilizzato per la generazione degli IDs mancanti. Questi IDs vengono generati nei nuovi file delle varie dimensioni tramite **enumerate**, partendo da zero e incrementando di uno ogni volta.

Custody e Crime.Gravity

Inizialmente è stato calcolato l'attributo **crime_gravity**, utilizzando i file JSON disponibili. Per ciascun record nel file **police_with_geoinfo.csv** è stata applicata una formula specifica, in cui ogni file JSON assegna un valore numerico in base allo stato del record. Questo attributo è stato quindi aggiunto al file per arricchire le informazioni relative a ciascun incidente con il grado di gravità corrispondente. Nello stesso programma **crimegravity_and_ids.py** è stato poi creato un nuovo file **Police_final.csv**. In quest'ultimo, oltre all'attributo precedentemente citato, sono aggiunti i rispettivi identificativi univoci mancanti (**participant_id**, **gun_id** e **geo_id**) per tutti i record. Questi ultimi vengono disposti utilizzando una funzione in cui viene effettuata la lettura dei file **Participant.csv**, **Gun.csv** e **Geography.csv** e per ognuno di essi viene creato un dizionario. Questi contengono tutti i record dei file associati e la loro chiave è rappresentata dai valori assunti dalle colonne di confronto, mentre il valore è l'identificativo univoco per ogni record. Questi dizionari sono poi adoperati per associare l'ID corretto ad ogni riga presente nel file **police_with_geoinfo.csv**, attraverso il confronto delle colonne considerate per ogni tabella. Infine viene creato il file **Custody.csv**, rappresentate la tabella *Custody*. Tale procedura inizia aprendo in lettura il file **Police_final.csv** e prelevando da esso solo le colonne con gli attributi attesi nella dimensione *Custody*, visibili nello schema del database. Concludendo, tutte le tabelle sono state riempite con le colonne rilevanti, seguendo lo schema del data warehouse fornito inizialmente. Questa procedura è stata eseguita al fine di semplificare il processo di popolamento del database.

1.3 Assignment 2: Popolare il database

Il codice creato per popolare il database è riportato nel file **Populate_DB.py** in cui sono presenti i seguenti passaggi:

- **Apertura della connessione:** usufruendo della libreria **pyodbc** e delle informazioni di connessione presenti nella stringa **connenction_string**. La connessione viene aperta tramite il blocco **with**, in modo da garantirne la chiusura alla fine del processo.
- **Utilizzo del cursore:** ovvero una struttura dati utile per iterare sui risultati della query SQL. Esso è creato all'interno della connessione e una volta completate tutte le operazioni viene chiuso.

- **Definizione di un dizionario:** *table_files* con le associazioni tra i nomi delle tabelle e i percorsi dei file csv corrispondenti.
- **Lettura dei file:** Per ogni tabella si apre il file CSV corrispondente e si utilizza *csv.reader* per ottenere un oggetto che può essere iterato per leggere le righe del file.
- **Inserimento dei dati:** viene creata una stringa SQL per l'inserimento dei dati di ogni riga del file nella tabella corrente. In base al file corrente selezionato, viene effettuato il cast tenendo conto della tipologia di dati presenti nelle varie colonne.
- **Commit:** viene eseguito il commit di ogni transazione per applicare le modifiche al database.

2. Analisi dati tramite SSIS

In questa Sezione sono esposti i passaggi seguiti per effettuare le analisi tramite **SQL Server Integration Services (SSIS)**. Le attività sono state realizzate senza l'utilizzo di comandi SQL, ma solo con i nodi standard di SSIS. Il lavoro eseguito e i relativi risultati sono contenuti nella cartella **LDS_Part2_group8**, al cui interno sono presenti tre sottocartelle denominate: **Assignment_0**, **Assignment_1** e **Assignment_2**; rispettivamente una per ogni task richiesto dalle linee guida. Gli esiti delle analisi condotte sono stati memorizzati nei rispettivi file: **Results_Assignment_0** contenente 230 record disposti in ordine crescente dell'anno, **Results_Assignment_1** contenente 52 record disposti considerando l'ordine alfabetico degli stati e **Results_Assignment_2** contenente 63 record ordinati prima in ordine crescente dei mesi e poi degli anni. Il salvataggio dei dati in questi file CSV è avvenuto tramite il componente **Destinazione File Flat**.

2.1 Assignment_0

L'*Assignment_0* richiede di ordinare i partecipanti in base al numero totale di custodies per ogni anno. I componenti SSIS utilizzati per ultimare questo task sono i seguenti:

- **Origine OLE DB:** per accedere alla tabella *Custody* e selezionare le colonne utili, ovvero *custody_id*, *date_fk* e *participant_id*.
- **Ricerca:** per creare un join delle colonne aggiuntive al flusso dei dati, selezionando l'attributo *year* dalla tabella *Date* grazie al mapping tra chiave primaria *date_pk* e chiave esterna *date_fk*.
- **Aggregazione:** per eseguire il conteggio del numero di custodies, raggruppando per anno e per partecipanti.
- **Ordinamento:** per ordinare per ogni anno le *Total_Custodies*.

2.2 Assignment_1

L'*Assignment_1* richiede di calcolare, per ogni stato, l'indice di gravità rubata ottenuto come il rapporto tra la gravità totale delle custodies quando *gun_stolen* è uguale a "Stolen" e la gravità di tutte le custodies (indipendentemente dal valore assunto dal precedente attributo). I componenti SSIS utilizzati per compiere questo task sono i seguenti:

- **Origine OLE DB:** per l'accesso alla tabella *custody*.
- **Ricerca:** due componenti di questo tipo: il primo per accedere ad informazioni aggiuntive della colonna *gun_stolen*, il secondo per accedere alle informazioni di *state*.

- **Multicast:** per creare un nuovo ramo del flusso di dati.
- **Suddivisione Condizionale:** usato per il flusso partito dal ramo destro del multicast, in modo da ottenere solo i record il cui valore di *gun_stolen* è uguale a *Stolen*.
- **Aggregazione:** due componenti usati uno per il primo ramo e l'altro per il secondo ramo, in modo da calcolare *Overall_Gravity* e *Stolen_Gravity*.
- **Ordinamento:** due nodi di questo tipo, uno per ogni aggregazione, per eseguire l'ordine in base alla variabile *State*.
- **Merge Join:** in modo da combinare i due flussi di dati ordinati tramite join.
- **Colonna Derivata:** per effettuare il calcolo della variabile finale *Stolen_Gravity_Index*.

2.3 Assignment 2

L'*Assignment_2* richiede di calcolare, per ogni mese, la percentuale di gravità totale rispetto al totale annuale.

E' stato deciso di eseguire il calcolo per ogni mese di ogni singolo anno, piuttosto che considerare l'insieme di tutti gli anni per ogni mese. In quest'ultimo caso, infatti, si otterrebbe per ogni mese un valore generale di gravità basato su tutti gli anni considerati. Invece, con il metodo adottato è possibile dedurre maggiori informazioni per l'utente finale. Per esempio, si possono effettuare indagini sulla gravità degli incidenti, in modo da comprendere quali sono i mesi e i periodi (invernale, estivo,...) in cui la gravità risulta essere superiore. Di conseguenza, l'utente potrebbe effettuare ulteriori analisi per capire le cause e le correlazioni dell'aumento di gravità rilevata in specifici mesi e/o periodi. Infine, in questa maniera è possibile desumere anche i risultati che si otterrebbero adoperando l'altra metodologia. I componenti SSIS utilizzati per svolgere questo task sono i seguenti:

- **Origine OLE BD:** per effettuare l'accesso alla tabella *Custody*.
- **Ricerca:** con cui vengono ricavate le informazioni aggiuntive tramite le colonne *month* e *year* della tabella *Date*.
- **Multicast:** in modo da avere due rami del flusso di dati che conducono ai due nodi di aggregazione.
- **Aggregazione:** due componenti con cui viene eseguita una GroupBy per anno e una per mese, calcolando rispettivamente la *crime_gravity* totale per ogni anno e per ogni mese di ogni anno.
- **Ordinamento:** due nodi utilizzati per effettuare l'ordinamento in base all'anno.
- **Merge Join:** per combinare i due flussi di dati ordinati.
- **Colonna Derivata:** per calcolare la percentuale.
- **Ordinamento:** per effettuare l'ordinamento principale per mese.

Per l'*Assignment_1* e l'*Assignment_2*, è stato ritenuto opportuno gestire gli errori che potrebbero presentarsi come una conseguenza del nodo *Colonna Derivata*. Tramite quest'ultimo, sono computati rispettivamente l'indice di *stolen_gravity* e la percentuale di gravità totale. Poiché entrambi i calcoli adoperano la divisione come operazione, gli eventuali errori (se presenti) saranno riportati nei file *LogErrors_1* e *LogErrors_2*. Concludendo, in tutti li Assignment è stato adoperato il componente **Attività Esegui Processo** della sezione Flusso di controllo, in modo da automatizzare l'apertura dei file CSV contenenti i rispettivi esiti.

3. Analisi dati tramite SSAS

In questa Sezione sono delineate le analisi condotte tramite **SQL Server Analysis Services (SSAS)**, che è un componente di Microsoft SQL Server, progettato per supportare l'analisi di dati multidimensionali e il data mining. Il lavoro eseguito e i relativi risultati sono contenuti nella cartella **LDS.Part3.group8**, al cui interno sono presenti i file dei vari task richiesti dalla traccia.

3.1 Costruzione Datacube

In questa fase viene creato il cubo multidimensionale che organizza i dati in modo da poterli esplorare facilmente in più dimensioni. Il cubo è denominato *Group ID 8 DB* e i file relativi alla sua costruzione sono presenti nella sottocartella **Datacube**. Nella struttura del cubo sono presenti quattro dimensioni: *Participant*, *Gun*, *Date* e *Geography*. La prima dimensione contiene gli attributi: *Participant Age Group*, *Participant Gender*, *Participant Id*, *Participant Status* e *Participant Type*. La seconda comprende *Gun Id*, *Gun Stolen* e *Gun Type*, mentre la terza è composta da *Date Pk*, *Day of the Week*, *Day Week*, *Month*, *Month of the Year*, *Quarter* e *Year*. Infine l'ultima dimensione include le variabili *City*, *Continent*, *Geo Id* e *State*. Gli attributi *Day Week* e *Month of the Year* sono stati creati per l'ordinamento dei valori. Per esempio, *Day Week* è servito ad ordinare i giorni della settimana dal lunedì alla domenica, in modo da non seguire l'ordine lessicografico originario. Sono state definite anche le dipendenze funzionali, in particolare tra *Month of the Year* e *Month*, e tra *Day of the Week* e *Day Week*. Ciò è stato fatto per assicurare che le indagini siano basate su relazioni logiche tra gli elementi delle dimensioni e le misure. La soluzione presenta due gerarchie nelle dimensioni *Date* e *Geography*, visibili nelle Figure 2 e 3, denominate *DayMonthQuarterYear* e *Geo*. Le precedenti sono gerarchie *non flat*, le cui dimensioni sono organizzate in una struttura gerarchica con livelli, sottodimensioni e relazioni gerarchiche tra i membri della stessa dimensione.

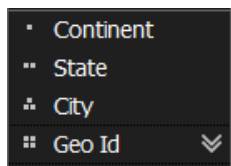


Figura 2: Gerarchia "Geo"

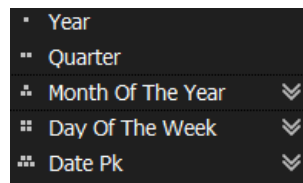


Figura 3: Gerarchia "DayMonthQuarterYear"

3.2 Query MDX

In questa fase sono descritti gli assignment eseguiti con il *Multidimensional Expressions*, infatti SSAS utilizza MDX come linguaggio di interrogazione per i cubi multidimensionali. I task richiesti per questa fase sono salvati nei file **Assignment_1**, **Assignment_2** e **Assignment_3**.

Assignment_1

Con questo primo assignment viene richiesto di mostrare la gravità totale della criminalità per ciascuna città e il totale complessivo rispetto allo stato. Per eseguire questa attività viene creata una query MDX, utilizzando WITH per creare il membro calcolato *Total_Crime*. Quest'ultimo è definito come il totale della *Crime Gravity* per ogni stato, utilizzando *currentmember* e accedendo al genitore attraverso *.parent*. Con una clausola SELECT, vengono specificate le colonne e le righe del risultato. Sulle colonne, vengono elencate le misure da visualizzare, che sono *[Measures].[Crime Gravity]* e il membro calcolato *Total_Crime*. Le

righe sono definite come le città (*[Geography].[City].[City]*) e gli stati (*[Geography].[Geo].[State]*). L'insieme non vuoto (indicato con la clausola NON EMPTY) è utilizzato per assicurarsi che vengano visualizzati solo i membri che hanno dati associati.

Assignment 2

Nel secondo assignment si richiede di evidenziare l'aumento o diminuzione di percentuale delle risposte totali sulla gravità del reato rispetto all'anno precedente per ciascuna fascia di età. Per realizzare questa query MDX, inizialmente viene definito il membro calcolato *crime_gravity* con WITH. Di conseguenza viene adoperata l'espressione condizionale *iif (If-Then-Else)*, per impostare il valore di *crime_gravity* a 0 se la misura *[Measures].[Crime Gravity]* è nulla, altrimenti assume il valore della misura. Successivamente, un altro membro calcolato *prev_crime* è definito come la gravità del crimine per il periodo precedente. Utilizza la funzione *.prevmember* per ottenere il membro precedente nella gerarchia *[Date].[DayMonthQuarterYear]* e calcola la gravità del crimine corrispondente. Il membro calcolato *Percentage_Variation* rappresenta la variazione percentuale tra il valore corrente di *crime_gravity* e il valore precedente *prev_crime*. Utilizza la formula $(crime_gravity - prev_crime) / prev_crime$ per calcolare la variazione percentuale, evitando divisioni per zero tramite la clausola IIF. Il risultato viene formattato come percentuale. Infine nella clausola SELECT vengono specificate le colonne e le righe del risultato, e tramite l'utilizzo di NONEMPTYCROSSJOIN si genera una combinazione di anni e gruppi di età del partecipante non vuoti. Invece, con la funzione ORDER il risultato viene ordinato in base alla variazione percentuale in modo decrescente (BDESC). Concludendo, sulle colonne si ottiene la variazione e la *crime_gravity*, mentre sulle righe gli anni e *age_group*. Questo tipo di query può essere utile per analizzare trend e variazioni nel tempo all'interno di diverse categorie di partecipanti e anni.

Assignment 3

Con il terzo assignment si desidera mostrare il rapporto tra la gravità totale della criminalità di ciascun anno rispetto all'anno precedente. Per la risoluzione di questo task, inizialmente viene precisato il membro calcolato *Ratio.Crime.Gravity* con la clausola WITH. Con la funzione IIF viene verificato se il membro precedente (*PrevMember*) nel livello *[Date].[DayMonthQuarterYear]* è nullo. Di conseguenza viene computato il rapporto tra la misura *Crime Gravity* relativa all'anno corrente, identificato tramite *currentmember*, e quella dell'anno precedente, individuato tramite *currentmember.lag(1)*. Per gestire il caso in cui l'anno corrente rappresenti il primo nel periodo complessivo, si fa ricorso al comando condizionale IIF. Tale comando restituisce il valore 0 nel caso in cui il membro corrente non disponga di membri precedenti, circostanza verificata mediante la condizione *currentmember.prevmember IS NULL*. La query principale seleziona le misure *[Measures].[Crime Gravity]* e *Ratio.Crime.Gravity* per le colonne. Inoltre utilizza la dimensione *[Date].[DayMonthQuarterYear].[Year]* per le righe, mostrando in questo modo il risultato del rapporto del *Crime Gravity* negli anni rispetto all'anno precedente. La clausola NON EMPTY viene impiegata per escludere eventuali anni privi di dati. Questo tipo di query è utile per analizzare la dinamica del crimine nel tempo e identificare variazioni significative nei tassi di gravità.

3.3 Dashboards

In questa Sezione sono illustrate le dashboard realizzate in **PowerBI**. Il lavoro compiuto è presente all'interno dei file **assignment 4** e **assignment 5**. Inizialmente, è stata stabilita la connessione al server selezionando l'opzione *Analysis Services* e inserendo <http://lds.di.unipi.it/olap/msmdpump.dll> come stringa di connessione. Successivamente, è stato selezionato il Cubo del Database del Gruppo numero 8 per proseguire con le analisi.

Assignment 4

Il primo compito concretizzato con questo software è la creazione di una dashboard che mostri la distribuzione geografica della gravità totale della criminalità in ciascuna fascia di età. Al fine di rappresentare questa ripartizione dei *total_crime_gravity* nelle varie categorie, sono state impiegate tre mappe separate (visibili in Figura 4), ciascuna incentrata su una specifica fascia d'età. La decisione di utilizzare tre mappe è stata intrapresa per avere una maggiore chiarezza visiva, poiché la configurazione delle bolle avrebbe reso difficile l'identificazione dei diversi gruppi di età rappresentati da colori differenti. Un vantaggio ottenuto da questa scelta progettuale risiede nella possibilità di poter individuare visivamente l'assenza di criminalità nelle Hawaii per la classe *Child 0-11*. Questa novità rappresenta la differenza principale tra i tre gruppi studiati. In aggiunta, è stato introdotto anche un grafico a barre raffigurante i totali dei reati relativi alle diverse fasce d'età. Tramite questo schema, è possibile sottolineare che la categoria *Adult 18+* è quella caratterizzata da una maggiore incidenza criminale.

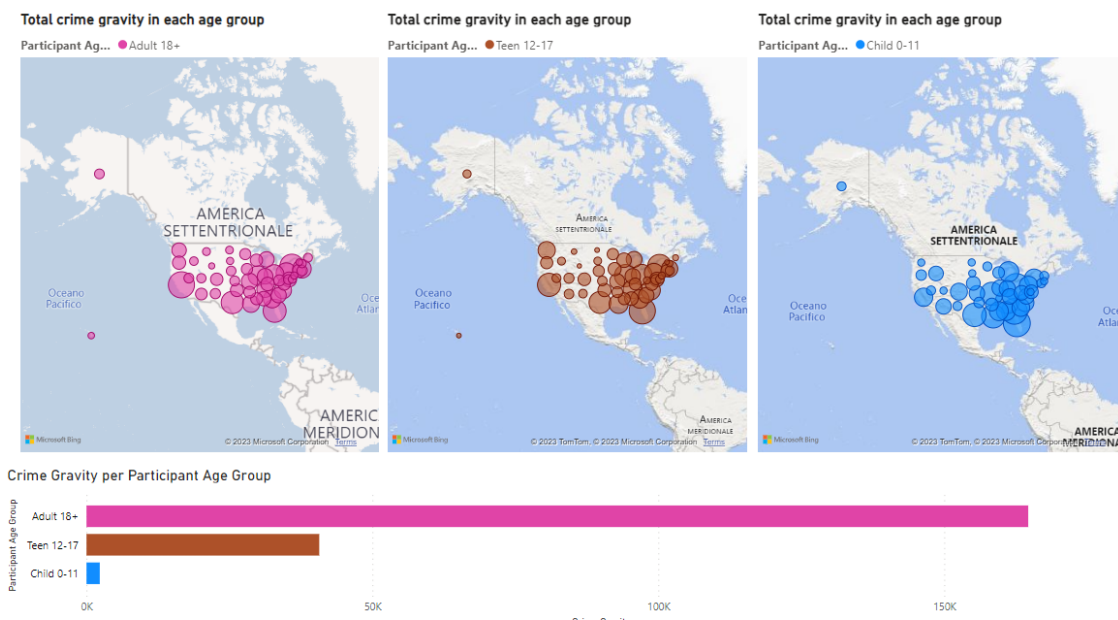


Figura 4: Distribuzione geografica della *total_crime_gravity* per ogni fascia d'età

Assignment 5

Per il task a scelta, è stato ritenuto interessante confrontare per ogni anno il valore della gravità dei crimini eseguiti nel caso di *gun_stolen* e *gun_not_stolen*. In questa dashboard è stato utilizzato un istogramma a colonne raggruppate, un grafico a linee e una tabella per rappresentare il valore di *crime gravity* per entrambe le tipologie d'arma (*stolen* e *not stolen*) e per ogni anno. Tutte queste raffigurazioni sono riportate nella Figura 5. L'istogramma fornisce una visione immediata delle differenze dei livelli di gravità criminale tra le differenti categorie di armi. Invece, il grafico a linee traccia le tendenze temporali, consentendo agli utenti di identificare eventuali variazioni nel corso degli anni e di visualizzare la relazione dinamica tra i due tipi di *gun*. La tabella facilita il confronto diretto dei dati, agevolando l'analisi dettagliata della distribuzione della *crime gravity*. Con l'ausilio di questi tre grafici, è possibile evidenziare come la gravità dei crimini aumenta notevolmente con il passare degli anni. In particolare l'incremento maggiore corrisponde ai crimini compiuti con armi rubate. Inoltre, andando avanti con gli anni, la differenza tra la *crime gravity* nel caso in cui l'arma è rubata e nel caso in cui non è rubata diventa più rilevante. Da ciò è possibile

ipotizzare che chiunque voglia compiere atti criminali lesivi diventa sempre più incline all'utilizzo di armi rubate.

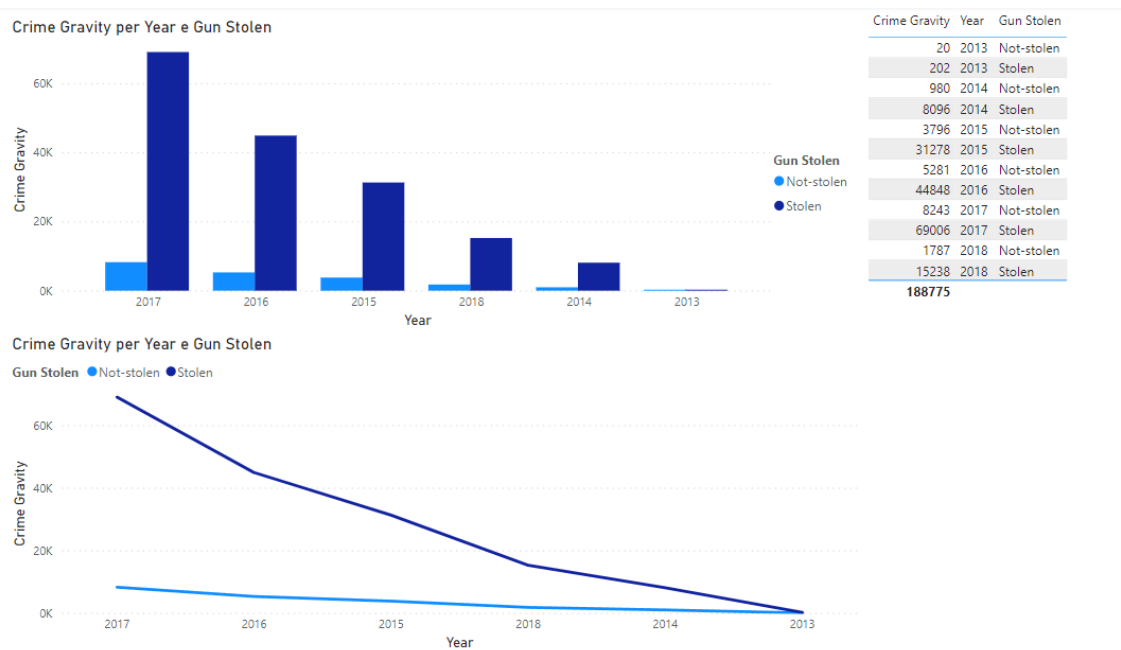


Figura 5: Crime Gravity per Year e Gun Type

Infine, è stato ritenuto di particolare interesse visualizzare anche come sono sparsi gli incidenti (in base all'id) nei vari stati, controllando nuovamente la ripartizione dei crimini in base al tipo di *gun* e riconfermando le ipotesi precedenti. Questa distribuzione è ritratta attraverso la mappa in Figura 6.

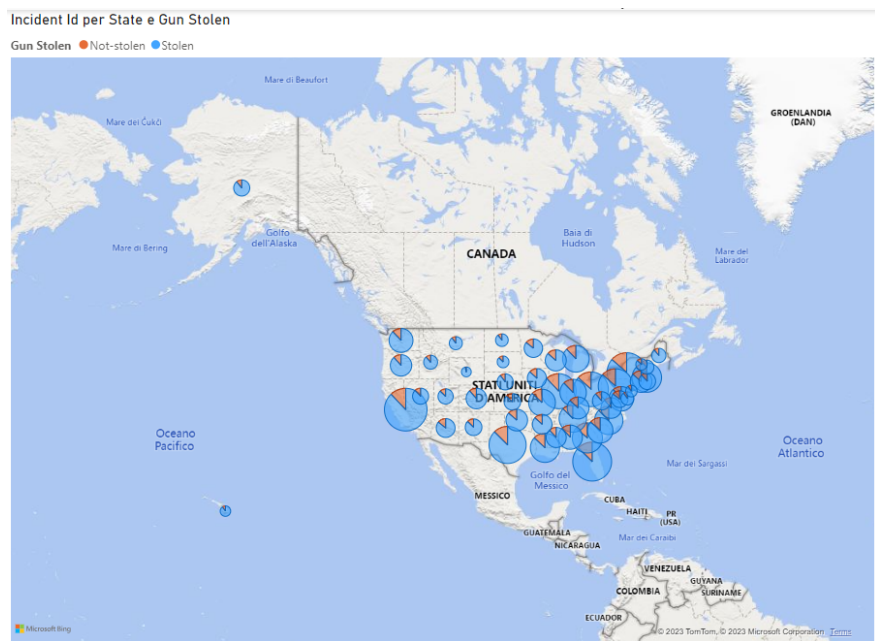


Figura 6: Distribuzione Incident.Id per State e Gun Type