Criminal cases database

SQL server final project

Temima Munic

February 4, 2024

# Project Overview:

The database I created for this project is a basic prototype of what a police force might have to keep track of cases in the city or country, and to cross reference information about them in order to find trends and better ways to stop crimes.

The database in my project lacks a lot of functionality and tables that would probably be needed in a real-life version, but as I said before, this is just a prototype.

## The major tables:

Cases: the main table in the dB, is used for keeping the basic info about every case (where and when it happened, what were the losses and so on)

Criminals: used for keeping track of basic info about every criminal the force has encountered before.

Investigators: keeps track of the investigators currently working for the force and their basic info.

Weapons: a minor table used for keeping track of the different weapons that might be used in a crime scene.

Case Types: another minor table used to ensure that all case types are standardized, and also allows the association of one crime with many types.

logCases: a log table on cases table

## The cross-reference tables:

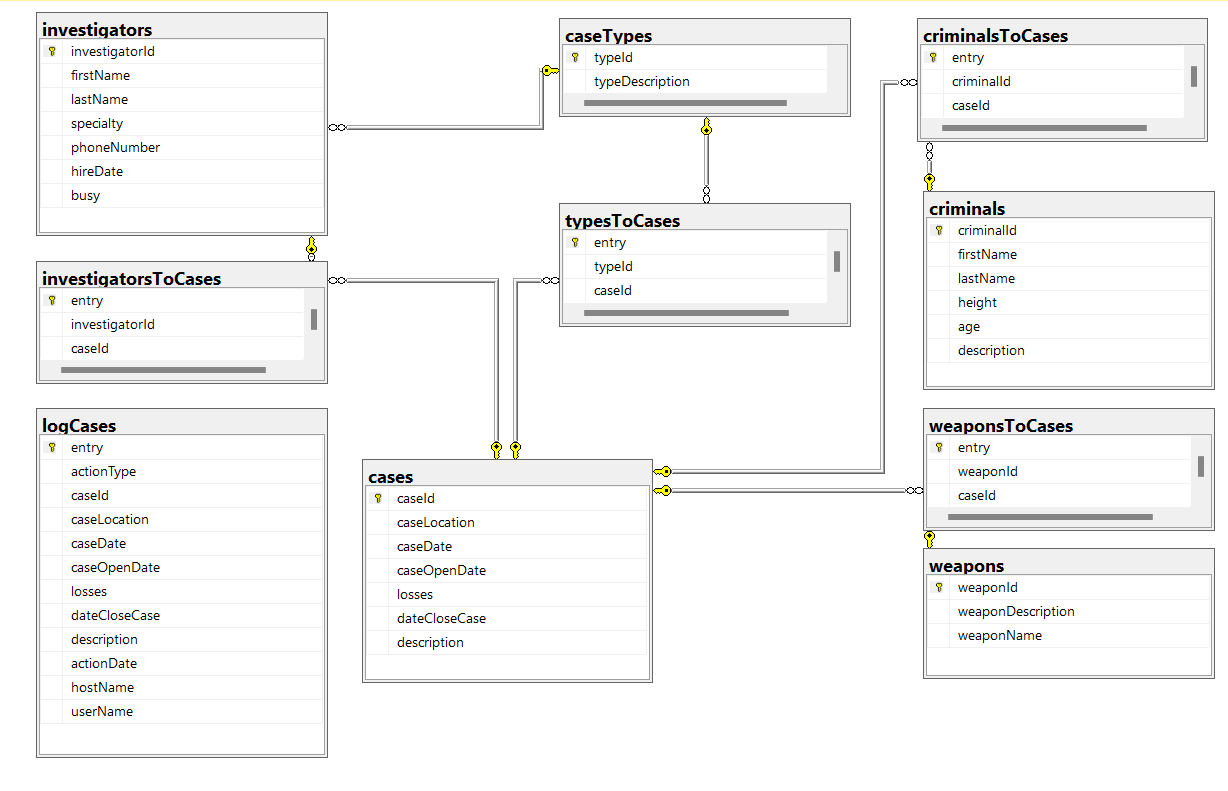
TypesToCases: an entry for every association between case and type.

InvestigatorsToCases: an entry for every investigator on every case they were assigned to.

CriminalsToCases: an entry for every criminal on every case they were proven to commit.

WeaponsToCases: an entry for every weapon on every case where it was used.

## The database Diagram:



## 

## The Views:

* combinedCaseData: a view that combines all the available data about a case into one table.
* criminalsToCrimeTypes: a view that creates an alignment between criminals and the types of crimes they have committed.
* most\_wanted: creates a rating of criminals based on the number of cases they have been associated with.
* first\_two\_cases: creates an alignment between every investigator to their first two cases.

## The Procedures:

* PR\_add\_case: the procedure receives the info about a case, adds it to case table and adds cross reference information about the weapons used, and the types of the case.
* PR\_assign\_investigators: the receives a caseId and the number of desired investigators on the case, and then assigns relevant investigators to the case by saving an entry in the investigatorsToCases table and sets the busy column for those investigatros to true. The procedure uses the FN\_recommend\_Investigators[[1]](#footnote-1) function to find relevant investigators.
* PR\_close\_case: the procedure assigns a close date to the given case and marks the investigators on that case as not busy.
* PR\_solve\_case: the procedure receives a case Id, and the Ids of the criminals involved and creates cross reference entries in the criminalsToCases table.

## The Functions:

* FN\_possible\_suspects: the function receives a case Id, and returns a table with the possible suspects from the criminals known to the force, based on their involvement in similar cases in the past (using the criminalsToCrimeTypes[[2]](#footnote-2) view)
* FN\_recommend\_Investigators: the function receives a case Id and recommend investigators based on their specialty; this function is used[[3]](#footnote-3) in the assignInvestigagators procedure.
* FN\_which\_criminal\_to\_catch\_first: the function receives to criminal ids and decides which one is more wanted based on the most\_wanted[[4]](#footnote-4) view.

## The triggers:

* newCase: when a new case is added to the database, the trigger saves the information in the database log.
* deleteCase: when a case is deleted from the database, the trigger saves the information in the database log.
* updateCase: does the same for updates, but the data is saved in two rows, one from the delete history, and the other from the insert history, this implementation might seem redundant to some, but it might be helpful when trying to isolate updates (which might happen a lot I this database as the case unfolds)

1. See functions. [↑](#footnote-ref-1)
2. See views. [↑](#footnote-ref-2)
3. See footnote one in procedures. [↑](#footnote-ref-3)
4. See views. [↑](#footnote-ref-4)