Montreal Police Intelligent Management System and Database

Group 10

Pawel Mankowski, Morgane Ciot, Nicolas Webster

I. Requirement Analysis

Introduction

Purpose
This application will store the information and constraints required to efficiently distribute Montreal's policing resources amongst the city's boroughs, based on factors such as crime rates and budget limitations. Resource optimization is a well-explored method that can prevent an organization from saturating its resources while satisfactorily meeting the city's crime-fighting demands. Cost-efficient distribution of resources, primarily police officers and budgets, therefore motivates the design of the database.

Scope and special requirements
We limit our sphere of interest to Montreal proper and consider the city's boroughs as our primary geographical unit of criminal investigation. We focus on criminal felonies such as gang violence as well as misdemeanors like traffic infractions to calculate the most cost-effective and crime-reducing deployment of police forces among the nineteen boroughs.

Terminology
crime rate = number of crimes / (population / 1000)

Resources
How to calculate the crime rate: http://www.icpsr.umich.edu/icpsrweb/NACJD/NCVS/rates.jsp

Database description

Entities and their attributes
The following entities will be stored in tables of a relational database:
Offender: An offender is someone who has committed an offense. An offender is associated with several attributes, including his or her last name, first name, race, address, gender, and date of birth, and is identified by a unique primary key called oID.
Offense: An offense is-a Crime or is-a Traffic Violation. The database keeps track of when the offense was committed, the address where the crime took place, and a description of the offense. It can be identified by its primary key, oiID.
Crime: A crime is a subcategory of the more general offense entity. It inherits all the features of an offense and supports its own attributes, including the type of crime committed and the severity of the crime.
Traffic Violation: A traffic violation is also a sub-category of the offense entity. Each violation is related to the intersection entity in order to localize areas which have a high rate of traffic violations. This data can be used to dictate where police officers can be allocated to maximize penalizing traffic violators.
Police officer: A police officer is the basic unit of resource at the department's disposal. A police

officer is associated with a police station in a particular borough and has a numerical rank attribute weighted by the officer's importance in the police force hierarchy. Officers also have a function attribute describing their role in the department (the full list can be seen in the Appendix section). Keeping track of this information will allow officers to be dispatched to areas where there specialty is required. The officer also has a first name, last name, salary, and can be identified by its primary key, poID.

Police station: A police station groups together all officers working in a particular borough of Montreal. Along with an address a police station has a non-salary cost per officer. This is used to help calculate the station's budget. A police station can be referenced by its unique primary key, psID.

Borough: A borough is a region governed by its own mayor and council. Montreal is divided into nineteen boroughs, each of which has its own name and police station. A borough can be referenced by its unique primary key, bID.

Intersection: An intersection consists of two cross-streets and can be identified by a unique primary key, iID. An intersection is located in a specific borough and is used to localize traffic violations. Intersections with a high volume of traffic infractions can thus easily pinpointed as destinations for increased patrol officers.

Population: A population is a weak entity that stores the populations of a borough for every year. The population of a given year is defined by that year and the foreign key of its borough, bID. Population information can be used to calculate current and future crime rates.

Relationships

commits: An *offender* commits an *offense.* This is a many-to-many relationship, because an offender can commit multiple offenses, and an offense can be committed by multiple offenders.

committed at: A *traffic violation* is committed at an *intersection.* This is a one-to-many relationship, because a violation can only be committed at one intersection, but many violations can transpire at one intersection. As aforementioned, this relationship is motivated by the idea of being able to localize areas where officers can be deployed in the future to monitor high-risk traffic areas.

committed in: A *crime* is committed in a *borough.* This is a one-to-many relationship, since a crime has only one borough in which it can be committed, but a borough can be the host of multiple crimes. This is again motivated by the idea of being able to circumscribe high- or particular-risk areas in order to more efficiently allocate policing resources.

in a: An *intersection* is in a *borough*. This is a one-to-many relationship, because an intersection can only physically occur once in a borough, and a borough can contain more than one intersection.

located in: A *police station* is located in a *borough*. This is a one-to-one relationship, since a police station cannot span several boroughs, and a borough has a unique police station.

works at: A *police officer* works at a *police station*. This is a one-to-many relationship, because a police officer can only work in his or her current assigned station, and there are multiple police officers in a police station. The works at relationship also has an attribute indicating when the officer first started working at that police station.

Application description

Overview
The application seeks to efficiently allocate Montreal's police forces around the city and determine an optimal budget for each borough based on statistics compiled during runtime. Storing the statistics in the database would require redundant calculations, which can all be performed when the optimization algorithm is run at regular intervals.

Preliminary calculations
The algorithm first retrieves data to compile statistics for each borough. For example, to calculate the crime rate for one borough, the algorithm extracts the population data from that borough, sums up the total amount of crimes committed in that area and plugs these values into the crime rate formula (see *Terminology*). Since the gravity of a crime or traffic infraction is indicated by a numerical ranking system, boroughs can be classified in terms of crime rates weighted by the aggregate severity of the crimes committed in that area. This allows the algorithm to prioritize certain boroughs over others when deciding how many police officers to deploy to each area.
Function examples
crimeRate(borough): This function returns a ranking indicating how dangerous the borough is.
expectedOfficers(borough): Given the crime rate in a borough, this function returns the estimated amount of officers needed to effectively combat crime in that borough.

Algorithm description
The heart of the algorithm lies not in its preliminary calculations, but in how it will solve the matrix of constraints and variables extracted from the database such that each borough receives the resources it requires to effectively fight crime and discourage traffic misdemeanors.
Variables to be determined
- A borough's **predicted future budget**, based on the maintenance cost per officer, the officers' minimum salaries, the borough's resource demands, and previous years' information on crime rates using exponential smoothing (this gives greater weight to more recent crime rates while still taking into consideration previous crime rates)
- **Number of police officers per borough**, based on overall crime rate in that borough
- Breakdown of **type of police officer per borough,** based on types of crimes committed in that area
- **High-risk traffic areas**, based on traffic infraction statistics
- **Number of patrol officers** assigned to high-risk traffic areas, based on preceding information

Appendix

Hierarchy of the Montreal police organization

function	*rank*
Chief Inspector	1
Inspector	2
Commander	3
Lieutenant	4
Sergeant	5
Constable	6
Tactical response officer	7
Intervention officer (riot police)	7
K-9 unit officer	7
Patrol officer	8
Investigator	8

Types of crimes

crime	*severity**	*examples*
violent crime	3	homocide, robbery, sexual offense, abduction
drug crime	2	possession, intent to sell, trafficking
property crime	1	fraud, breaking and entering

*Severity is measured on a scale of 1 to 3, 3 being the most violent type of crime.

III. Relations

Entities:

Offender(oID, gender, date_of_birth, fName, lName, race, address)

Offense(ofID, date_committed, address, offense_type, description)

traffic_violation(oID)

crime(severity, oID)

Intersection(iID, first_street_name, second_street_name)

Borough(bID, name)

PoliceOfficer(poID, fName, lName, rank, salary, function)

Police Station(psID, numberCars, address, non_salary_cost_per_officer)

Weak Entity:

Population(year, pop_count, bID)

Relationships:
commits(ofUD, oID)

works_at(psID, poID, since)

station_in(bID, psID)

intersection_in(bID, iID)

committed_at(iID, ofID)

crime_committed_in(bID, ofID)

We do not feel there is any way to combine any of the entities we have. For example, if we were to combine the traffic violation entity and the crime entity and just use the offense entity, we would still be able to use an attribute to differentiate between the two, but we would no longer easily be able to match intersections with traffic violations without compromising the efficiency of actions with the data that are not relevant to intersection.