

Licence 3 MIASHS 2018-2019

Base de données

Rapport de mini-projet

Quels quartiers de Montpellier sont les mieux protégés ?

Auteurs :

Samy AYED
Ayoub HASSANI

Encadrante :

Sandra BRINGAY



Version 1 du
13 décembre 2018

Table des matières

1	Introduction	3
2	Objectifs	3
3	MCD et MOD	3
4	Liste des requêtes	4
5	Interprétation des requêtes	6
6	Difficultés rencontrées	7
7	Perspectives	7
8	Conclusion	8
9	Liens vers les données	8

1 Introduction

Le sujet de notre mini-projet porte sur la protection des quartiers de la ville de Montpellier. Cette thématique va nous amener à nous interroger sur les différences de niveau de sécurité entre les quartiers de l'agglomération, eux-mêmes découpés en sous-quartiers affinant ainsi la précision de notre étude. Ces derniers ont bien sûr des caractéristiques variables, et sont couverts par un certain nombre de caméras de vidéoprotection, lesquelles nous serviront de référentiel pour déterminer le niveau de sécurité de chacun. Ainsi, seules les caméras installées par la collectivité rentrent dans notre champ d'étude : les bases de données exploitées n'apportent pas d'informations sur les installations de quartiers ou d'entreprises privés. L'idée est donc de faire usage de deux bases de données fournies par l'Open Data de Montpellier, nous permettant d'exploiter le meilleur des informations disponibles pour répondre avec le plus de précision possible à la problématique posée. Par conséquent, le langage SQL est un outil essentiel pour manipuler les données présentes en grande quantité et obtenir des réponses à travers nos requêtes.

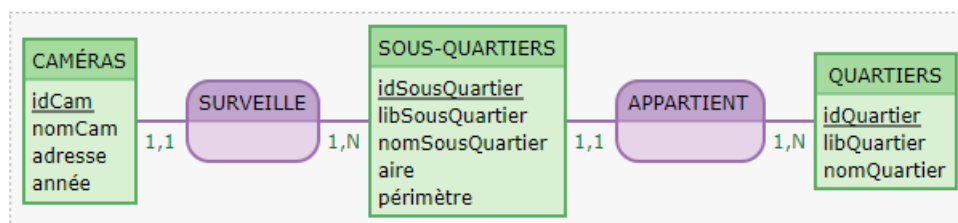
2 Objectifs

Dans une démarche de lecture et d'exploitation objective des données, nous avons tout d'abord pour objectif de faire une comparaison inter-quartier. Nous dresserons ensuite un classement des zones en fonction de leur niveau de sécurité, pour potentiellement donner des indices sur les lieux où il ferait bon d'habiter. Nous révélerons par ailleurs les potentielles carences de sécurité, dans l'idée qu'elles soient peut-être un jour amenées à être corrigées..

Une autre dimension importante concerne l'évolution de la protection à travers le temps, à savoir si les équipements de surveillance augmentent toujours un peu plus chaque année, s'ils stagnent ou varient plus faiblement dans leur nombre effectif. Autrement dit, est-ce que la ville tend à se sécuriser ?

Nous tenterons de répondre à ces questions en prenant en compte 3 tables faisant le lien entre les données : les caméras de surveillance, les quartiers et les sous-quartiers.

3 MCD et MOD



CAMÉRAS (idCam, nomCam, adresse, année, *idSousQuartier*)

SOUS-QUARTIERS (idSousQuartier, libSousQuartier, nomSousQuartier, aire, périmètre, *idQuartier*)

QUARTIERS (idQuartier, libQuartier, nomQuartier)

4 Liste des requêtes

1. Établissez un classement décroissant des sous-quartiers selon le nombre de caméras de vidéoprotection qu'ils disposent (a). Établissez de la même façon un classement des quartiers (b).
 - (a)

```
SELECT sous_quartiers.idSousQuartier, sous_quartiers.nomSousQuartier,
COUNT(caméras.idCam) AS 'nbCam' FROM sous_quartiers LEFT JOIN caméras
ON sous_quartiers.idSousQuartier = caméras.idSousQuartier GROUP BY
sous_quartiers.idSousQuartier ORDER BY COUNT(caméras.idCam) DESC;
```
 - (b)

```
SELECT quartiers.idQuartier, quartiers.nomQuartier, COUNT(*) AS 'nbCam'
FROM sous_quartiers, caméras, quartiers WHERE sous_quartiers.idSousQuartier =
caméras.idSousQuartier AND quartiers.idQuartier = sous_quartiers.idQuartier
GROUP BY quartiers.idQuartier ORDER BY COUNT(*) DESC;
```
2. Quel est le nombre moyen de caméras dans les quartiers (a) et sous-quartiers de la ville (b)?
 - (a)

```
SELECT COUNT(caméras.idCam)/(SELECT COUNT(idQuartier)
FROM quartiers) AS 'nb moyen' FROM caméras;
```
 - (b)

```
SELECT COUNT(caméras.idCam)/(SELECT COUNT(idSousQuartier)
FROM sous_quartiers) AS 'nb moyen' FROM caméras;
```
3. Donnez pour chaque année le nombre de caméras ayant été installées (a).
 - (a)

```
SELECT caméras.année, COUNT(*) AS nbCam FROM caméras GROUP BY caméras.année;
```
4. Établissez un classement décroissant des quartiers en fonction du nombre de caméras installées en 2016 (a). Faites la même chose pour les sous-quartiers (b).
 - (a)

```
SELECT nomquartier, COUNT(*) AS nbCam FROM caméras, sous_quartiers,quartiers WHERE
caméras.idsousquartier = sous_quartiers.idSousQuartier AND quartiers.idQuartier =
sous_quartiers.idQuartier AND année = 2016 GROUP BY nomquartier ORDER BY nbCam
DESC;
```
 - (b)

```
SELECT nomSousQuartier, COUNT(*) AS nbCam FROM caméras, sous_quartiers WHERE
caméras.idsousquartier = sous_quartiers.idSousQuartier AND année = 2016 GROUP BY
nomSousQuartier ORDER BY nbCam DESC;
```
5. Quels sous-quartiers n'ont pas été équipés en 2016 (a)?
 - (a)

```
SELECT nomSousQuartier FROM sous_quartiers WHERE idSousQuartier NOT in
(SELECT sous_quartiers.idSousQuartier FROM sous_quartiers, caméras WHERE
sous_quartiers.idSousQuartier = caméras.idSousQuartier AND caméras.année = 2016);
```
6. Faites un classement décroissant des quartiers selon leur superficie (a).
 - (a)

```
SELECT quartiers.nomQuartier, SUM(sous_quartiers.aire) AS superficie FROM sous_quartiers,
quartiers WHERE quartiers.idQuartier = sous_quartiers.idQuartier GROUP BY quartiers.idQuartier
ORDER BY superficie DESC;
```

7. Déterminez la densité de caméras dans le ou les sous-quartiers les plus équipés (a).
Même question pour les sous-quartiers les moins équipés (b).

(a) `SELECT cnt.nom, cnt.aireSQ/cnt.nbCam AS surfaceCam FROM
(SELECT sous_quartiers.nomSousQuartier AS nom, sous_quartiers.aire AS
aireSQ, COUNT(*) AS nbCam FROM sous_quartiers, caméras WHERE
caméras.idSousQuartier = sous_quartiers.idSousQuartier GROUP BY
sous_quartiers.idSousQuartier) AS cnt WHERE cnt.nbCam = (SELECT MAX(cnt.nbCam)
FROM (SELECT sous_quartiers.nomSousQuartier AS nom, sous_quartiers.aire
AS aireSQ, COUNT(*) AS nbCam FROM sous_quartiers, caméras
WHERE caméras.idSousQuartier = sous_quartiers.idSousQuartier
GROUP BY sous_quartiers.idSousQuartier) AS cnt);`

(b) `SELECT cnt.nom, cnt.aireSQ/cnt.nbCam AS surfaceCam FROM
(SELECT sous_quartiers.nomSousQuartier AS nom, sous_quartiers.aire AS
aireSQ, COUNT(*) AS nbCam FROM sous_quartiers, caméras WHERE
caméras.idSousQuartier = sous_quartiers.idSousQuartier GROUP BY
sous_quartiers.idSousQuartier) AS cnt WHERE cnt.nbCam = (SELECT MIN(cnt.nbCam)
FROM (SELECT sous_quartiers.nomSousQuartier AS nom, sous_quartiers.aire
AS aireSQ, COUNT(*) AS nbCam FROM sous_quartiers, caméras
WHERE caméras.idSousQuartier = sous_quartiers.idSousQuartier
GROUP BY sous_quartiers.idSousQuartier) AS cnt);`

8. Déterminez le quartier qui dispose de la plus grande densité de caméras (a). Idem
pour celui où elle est minimale (b).

(a) `SELECT
t1.nomQuartier, t1.nbCam, t2.superficie, t2.superficie/t1.nbCam AS surfaceCam
FROM
(SELECT quartiers.idQuartier, quartiers.nomQuartier, COUNT(*) AS nbCam FROM
quartiers, caméras, sous_quartiers WHERE caméras.idSousQuartier = sous_quartiers.idSousQuartier
AND sous_quartiers.idQuartier = quartiers.idQuartier GROUP BY quartiers.idQuartier) t1
INNER JOIN
(SELECT quartiers.idQuartier, SUM(sous_quartiers.aire) AS superficie FROM sous_quartiers,
quartiers WHERE quartiers.idQuartier = sous_quartiers.idQuartier GROUP BY quartiers.idQuartier)
t2 ON t1.idQuartier = t2.idQuartier ORDER BY surfaceCam LIMIT 1;`

(b) `SELECT
t1.nomQuartier, t1.nbCam, t2.superficie, t2.superficie/t1.nbCam AS surfaceCam
FROM
(SELECT quartiers.idQuartier, quartiers.nomQuartier, COUNT(*) AS nbCam FROM
quartiers, caméras, sous_quartiers WHERE caméras.idSousQuartier = sous_quartiers.idSousQuartier
AND sous_quartiers.idQuartier = quartiers.idQuartier GROUP BY quartiers.idQuartier) t1
INNER JOIN
(SELECT quartiers.idQuartier, SUM(sous_quartiers.aire) AS superficie FROM sous_quartiers,
quartiers WHERE quartiers.idQuartier = sous_quartiers.idQuartier GROUP BY quartiers.idQuartier)
t2 ON t1.idQuartier = t2.idQuartier ORDER BY surfaceCam DESC LIMIT 1;`

5 Interprétation des requêtes

1. (a) Dans cette requête, LEFT JOIN est employé pour faire apparaître dans le classement les sous-quartiers qui ne disposeraient d'aucune caméra. On en retrouve 2 en l'occurrence : Les Aubes et Les Cévennes. En outre, Centre Historique et Comédie arrivent ex-aequo en 1ère position avec 18 caméras.
- (b) Les sous-quartiers qui appartiennent au même quartier sont agrégés et la somme de leurs caméras est effectuée. Chaque quartier en possède ici au moins une. Le quartier Montpellier Centre arrive à la 1ère position avec 97 caméras, loin devant Mosson qui est en 2ème place avec 34 caméras. Près d'Arène se retrouve dernier au classement avec 7 caméras.
2. (a) Le nombre total de caméras dans l'agglomération est compté dans une première requête, puis il est divisé par le nombre de quartiers dans une deuxième. On obtient 29,7 caméras en moyenne pour chaque quartier.
- (b) Le raisonnement est similaire à la (a). Un résultat de 6,7 caméras par sous-quartier est obtenu.
3. (a) Chaque année durant laquelle au moins une caméra a été installée apparaît dans ce classement. 2016 fut l'année la plus prolifique avec 36 installations. On observe une hausse non-négligeable des installations depuis 2014 (+100% en 2015, +50% en 2016, comparativement à l'année précédente).
4. (a) On ne s'intéresse ici qu'à l'année 2016, cette condition après WHERE est donc rajoutée. On observe que Montpellier Centre est largement en tête avec 20 caméras installées, soit 5 fois plus que le quartier Les Cévennes (à ne pas confondre avec le sous-quartier portant le même nom) qui se trouve en 2ème position. En outre, plus de 50% des installations de l'année 2016 sont intervenues dans le quartier Montpellier Centre.
- (b) Le raisonnement est similaire à la (a) mais on ne s'intéresse qu'aux sous-quartiers. Antigone est le sous-quartier le plus représenté, 8 caméras y ont été installées en 2016.
5. (a) Une sous-requête est utilisée pour récupérer l'identifiant de tous les sous-quartiers dans lesquels des installations ont eu lieu. Si l'identifiant d'un sous-quartier dans la table sous_quartier n'est pas présent, on le récupère dans notre résultat. 11 sous-quartiers n'ont pas été l'objet d'installations.
6. (a) La fonction SUM est utilisée pour additionner la superficie de chaque sous-quartier d'un même quartier. On obtient ainsi la superficie totale du quartier. Le résultat obtenu étant en m^2 , on peut le diviser par 10^6 pour connaître le résultat en km^2 . Hôpitaux-Facultés arrive donc en tête avec $12,5 km^2$.
7. (a) Une première sous-requête nous permet de grouper les mêmes sous-quartiers, joints à la table caméras, et de récupérer pour chacun leur nombre de caméras. Une deuxième sous-requête après WHERE nous permet de récupérer tous les

éventuels ex-aequo en valeur max, en l'occurrence Centre Historique et Comédie qui possèdent tous deux 18 caméras. Diviser la superficie d'un sous-quartier par son nombre de caméras nous permet d'obtenir sa densité de caméras. On obtient par exemple pour Comédie : 1 caméra / 18915m². Que l'on pourrait également traduire par 1 cam. / 0,019 km² soit 52,9 cam. / km².

- (b) Le raisonnement est identique mais toutes les valeurs min sont récupérées. Néanmoins, ne sont pris en considération les sous-quartiers qui ne possèdent aucune caméra (Les Aubes et Les Cévennes). Les sous-quartiers retenus ici possèdent tous qu'une seule caméra. La densité caméra / X m² sera donc toujours obtenue avec X égal à la superficie du sous-quartier.
- 8. (a) Une première sous-requête nous donne une table avec le nombre de caméras par quartier. Une deuxième nous apporte leur superficie. A l'aide d'une clé commune qui n'est autre que l'identifiant du quartier, les 2 tables sont jointes via un INNER JOIN. Il ne reste qu'à diviser la superficie par le nombre de caméras, et la densité est obtenue. La table est triée dans l'ordre croissant et une limite de 1 est fixée pour obtenir la densité la plus importante (ne pas oublier que le résultat s'obtient sous la forme : 1 cam. / X m²). Montpellier Centre détient la plus grande densité de caméras avec 1 cam. / 82 756 m².
- (b) Le raisonnement est identique mais le tri est effectué dans l'ordre décroissant pour récupérer la plus faible densité. Hôpitaux-Facultés détient la plus faible avec 1 cam. / 1 386 781 m².

6 Difficultés rencontrées

Des informations manquantes dans l'un des 2 jeux de données à notre disposition nous ont posé quelques difficultés. En effet, les quartiers et sous-quartiers d'appartenance de toutes les caméras de surveillance installées après 2016 n'étaient pas précisés. Cette non mise à jour pourrait dans certains cas nous pousser à chercher des alternatives pour récupérer facilement les informations manquantes, mais nous reconsidérerons ici simplement la période sur laquelle se base notre étude. Hormis cela, nous n'avons pas rencontré de difficulté technique ou organisationnelle.

7 Perspectives

Il serait intéressant pour la suite de comparer le niveau de sécurité des quartiers avec leur taux de délinquance, de vol ou de cambriolage. Il semble à première vue y avoir une corrélation évidente : une meilleure protection devrait limiter les faits de délinquance. Mais en mesurer la force nous permettrait de savoir si les moyens déployés par la ville sont efficaces, et s'il ne serait pas de bon augure de renforcer cette sécurité dans certains quartiers.

8 Conclusion

De manière succincte, nous pouvons d’ores et déjà affirmer que la répartition des caméras de vidéoprotection dans l’agglomération de Montpellier est assez hétérogène. Si les quartiers et sous-quartiers au cœur de la ville profitent d’un nombre non-négligeable de caméras qui assurent une certaine sécurité (Montpellier Centre : Comédie, Gares, Centre Historique), d’autres en revanche ne sont pas encore au même niveau (Les Cévennes, Près d’Arènes). En effet, la ville sécurise en priorité les quartiers qu’elle considère les plus importants et les plus affluents. Toutefois, on observe depuis 2014 une hausse constante des installations de caméras de vidéoprotection, permettant d’améliorer progressivement la sécurité dans les zones les moins protégées.

9 Liens vers les données

- [Emplacement des caméras](#)
- [Sous-quartiers de Montpellier](#)