# Requêtes SQL : Meurtre à SQL City

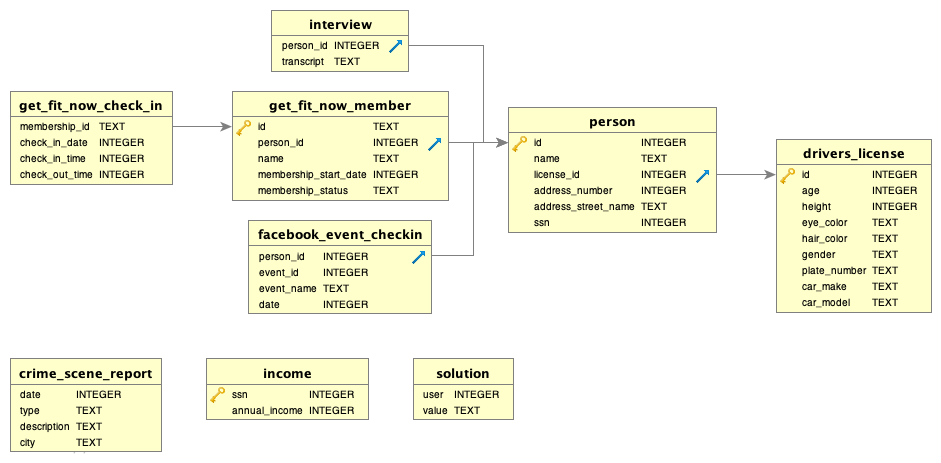
|  |  |
| --- | --- |
| Objectifs | S’entrainer aux requêtes SQL |
| Méthodologie | Travail individuel |
| Durée estimée | 45 min (avec le corrigé) |
| Type de base de données : | SQLite |

Grâce à vos nouvelles connaissances en SQL, vous avez décidé d'ouvrir votre propre agence de détective privé. Votre client vous a mandaté pour retrouver un meurtrier car il n'a pas confiance dans le travail de la police.

Après un petit peu de « fouinage », vous avez pu récupérer la base de données de la police ainsi que quelques informations incontestables :

* le meurtre a eu lieu le 15 janvier 2018 ;
* le meurtre s'est déroulé à SQL City ;
* le club de sport "Get Fit Now !" a un lien avec cette affaire.

La base de données **sql-murder-mystery** contient les tables suivantes :



Pour commencer, vous devez télécharger le dump de cette base de données, à avoir le fichier **sql-murder-mystery.sql**.

Attention ! Nous n’allons pas utiliser le SGBDR MySQL pour cet exercice. Nous allons utiliser exceptionnellement SQLite.

Vous aurez besoin de [DB Browser](https://sqlitebrowser.org/) pour interagir avec cette base de données.

**ATTENTION !**

La base de données que vous allez interroger n’est pas toujours très fiable.

* Pour les interrogations sur les chaînes de caractères, il est conseillé d'utiliser le mot-clef **LIKE**
* Ou lorsque vous utilisez l’opérateur **=** vous pouvez utiliser :

**WHERE** **LOWER**(city) = 'sql city' pour éviter les problèmes de « casse »

**Votre mission** :

Recherchez l'assassin en formulant des requêtes SQL successives dans l'éditeur de DB Browser. Expliquer votre démarche et les requêtes que vous avez effectuées. Le but est de trouver l’assassin en un minimum de requêtes.

**Comment savoir si on a trouvé l’assassin ?**

Une fois que vous pensez avoir identifié cet assassin, terminez par les deux requêtes suivantes :

1. Insertion d'un nouvel enregistrement dans la table solution, dans lequel vous remplacer la chaîne 'le nom du suspect' par le nom de votre suspect écrit entre apostrophes :

|  |
| --- |
| **INSERT** **INTO** solution  **VALUES** (1, 'le nom du suspect'); |

1. Vérification de cette proposition à l'aide d'un déclencheur (hors programme) :

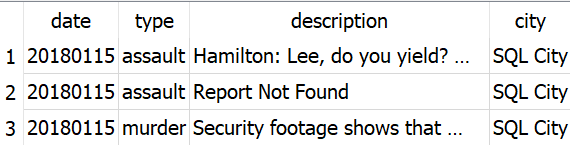
|  |
| --- |
| **SELECT** value  **FROM** solution; |

**Solution :**

1ère requête sur la table 'crime\_scene\_report'

A l’aide des informations du départ, on sélectionne « Rapport sur les lieux du crime »

|  |
| --- |
| SELECT \*  FROM crime\_scene\_report  WHERE date = '20180115'  AND lower(city) = 'sql city'; |



On obtient 3 enregistrements mais seulement 1 est un Meurtre.

Voici la traduction de la description :

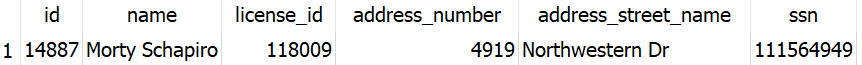
Les images de sécurité montrent qu’il y avait 2 témoins.

Le premier témoin vit dans la dernière maison de la « Northwestern Dr ».

Le deuxième témoin, nommé Annabel, vit quelque part sur « Franklin Avenue ».

Le 1er témoin :

|  |
| --- |
| SELECT \*  FROM person  WHERE address\_street\_name LIKE '%Northwestern Dr%'  ORDER BY address\_number DESC  LIMIT 1; |



Le 2ème témoin :

|  |
| --- |
| SELECT \*  FROM person  WHERE address\_street\_name LIKE '%Franklin%'  AND name LIKE 'Annabel%'; |



Maintenant que l’on connaît les 2 témoins, on peut trouver leur déposition :

|  |
| --- |
| SELECT name, transcript  FROM person  JOIN interview ON interview.person\_id = person.id  WHERE person\_id IN ( 14887, 16371); |

Déposition (colonne **transcript**) de Morty Schapiro :

« J’ai entendu un coup de feu, puis j’ai vu un homme s’enfuir.

Il avait un sac « Get Fit Now Gym ».

Le numéro de membre sur le sac commençait par « 48Z ».

Seuls les membres Gold ont ces sacs.

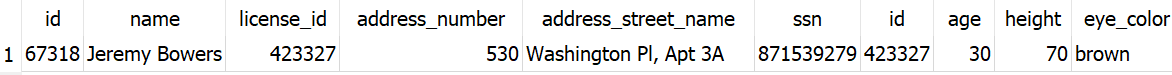
L’homme est monté dans une voiture avec une plaque sur laquelle figurait « H42W » ».

Déposition de Annabel Miller :

« J’ai vu le meurtre se produire, et j’ai reconnu le tueur de mon gymnase quand je m’entraînais la semaine dernière, le 9 janvier. »

A partir des informations des témoins, nous pouvons faire la requête suivante :

|  |
| --- |
| SELECT \*  FROM person  JOIN drivers\_license  ON person.license\_id = drivers\_license.id  JOIN get\_fit\_now\_member  ON get\_fit\_now\_member.person\_id = person.id  JOIN get\_fit\_now\_check\_in  ON get\_fit\_now\_check\_in.membership\_id = get\_fit\_now\_member.id  WHERE plate\_number LIKE '%H42W%'  AND get\_fit\_now\_member.id LIKE '48Z%' AND membership\_status = 'gold'  AND check\_in\_date LIKE '20180109%'  AND membership\_id IN ('48Z7A', '48Z55'); |



Maintenant que notre suspect est Jeremy Bowers, il ne nous reste plus qu’à vérifier cette solution :

|  |
| --- |
| INSERT INTO solution  VALUES (1, 'Jeremy Bowers');  SELECT value  FROM solution; |

Félicitations, vous avez trouvé le meurtrier! Mais attendez, il y a plus... Si vous pensez que vous êtes prêt à relever un défi, essayez d’interroger la transcription de l’interview du meurtrier pour trouver le vrai villian derrière ce crime. Si vous vous sentez particulièrement confiant dans vos compétences SQL, essayez de terminer cette dernière étape avec pas plus de 2 requêtes. Utilisez cette même instruction INSERT avec votre nouveau suspect pour vérifier votre réponse.

Donc on fait une requête avec l’id de Jeremy Bowers

|  |
| --- |
| SELECT transcript FROM interview where person\_id = 67318; |

On obtient :

« J’ai été embauché par une femme avec beaucoup d’argent. Je ne connais pas son nom, mais je sais qu’elle mesure environ 5 pi 5 po (65 po) ou 5 pi 7 po (67 po). Elle a les cheveux roux et elle conduit une Tesla Model S. Je sais qu’elle a assisté au SQL Symphony Concert 3 fois en décembre 2017. »

Ensuite :

|  |
| --- |
| SELECT \*  FROM drivers\_license  JOIN person ON person.license\_id = drivers\_license.id  JOIN facebook\_event\_checkin ON facebook\_event\_checkin.person\_id = person.id  WHERE facebook\_event\_checkin.date like "201712%" ;  AND facebook\_event\_checkin.event\_name LIKE "%symphony%"  AND drivers\_license.car\_model = "Model S"; |