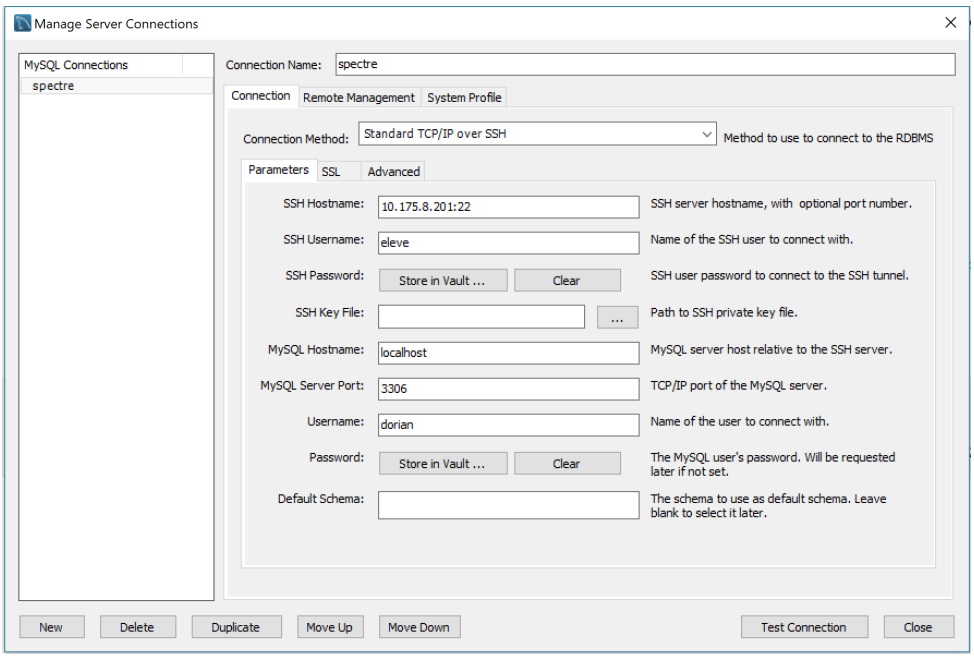
I Connexion aux bases

Ouvrir le logiciel MySQL Workbench (icône dauphin).
Si l'icône suivant apparaît cliquer dessus.

S'il n'apparaît pas, il faut configurer une nouvelle connexion en cliquant sur le bouton "+" et compléter les paramètres comme sur la figure ci-dessous. Une fois la connexion cinfigurée, il faudra préciser les mots de passe : `eleve` pour `eleve` et `b306` pour `dorian`.



Lorsque la nouvelle connexion a été configurée, l'icône mentionné précédemment apparaît, cliquer dessus.

II Triangles

Double cliquer sur la base de données `triangles` (colonne de gauche, « SCHEMAS ») qui est une base de données constituée d'une seule table également nommées `triangles` dont le schéma relationnel [1] est le suivant : `triangles (idt:int, ab:int, ac:int, bc:int)`
Chaque ligne représente un triangle défini par la taille de ses trois côtés (AB, AC et BC).

1. Après avoir réfléchi à leurs significations, lancer successivement les requêtes suivantes :
   ```
   SELECT COUNT (*) FROM triangles;
   SELECT * FROM triangles WHERE ab+ac+bc=100;
   SELECT ab*ac*bc FROM triangles WHERE ab+ac+bc>=100;
   ```
2. Déterminer à l'aide de requêtes SQL :
   (a) la plus petite valeur des produits AB x AC x BC pour les triangles ABC de périmètre supérieur ou égal à 100 ;
   (b) les longueurs AB, AC et BC correspondants aux triangles pour les lesquels le minimum précédent est atteint ;
   (c) tous les triangles rectangles en A ;

1. Le schéma relationnel est le nom de la table, suivi des différents attributs et des domaines (entiers, texte, etc.) de ces attributs.

(d) le nombre de triangles rectangles en A ;

(e) le maximum des périmètres des triangles rectangles en A ;

(f) tous les triangles équilatéraux ;

(g) tous les triangles tels que $\frac{AB+AC+BC}{3} = 42$

(h) exporter [2] les résultats de la requête précédente dans un fichier au format csv.

III Communes, département et régions

1. Ouvrir la base de données `communes` en double cliquant dessus et écrire sur papier les schémas relationnels des trois tables qu'elle contient. On obtient le schéma relationnel d'une table en sélectionnant cette table dans la barre latérale gauche puis en cliquant sur l'onglet « Objetc Info ».
2. Exécuter la requête suivante :
   ```
   SELECT communes.nom, departements.nom
   FROM communes JOIN departements
   ON communes.dep=departements.id
   ```
   Pour simplifier l'écriture de telles requêtes, on peut utiliser des alias pour chacune des tables, grâce à l'opérateur `AS`. Par exemple, la requête précédente peut s'écrire :
   ```
   SELECT C.nom, D.nom FROM communes AS C JOIN departements AS D ON C.dep=D.id
   ```
3. En vous inspirant du modèle précédent, déterminer la liste de toutes les communes avec pour chacune son département, sa région et sa population.
4. (a) Déterminer la liste des villes de plus de cent milles habitants avec leur population et leur région.
   (b) Trier la liste précédente par ordre décroissant de population. On pourra pour cela utiliser la commande `ORDER BY` (voir Annexe).
5. Déterminer la liste des communes (nom et population) dont le nom commence par `Pa` et se termine par `is`. On pourra pour cela utiliser l'opérateur `LIKE` (voir Annexe).
6. Déterminer les communes dont le nom a strictement plus de lettres que leur nombre d'habitants. On pourra utiliser la fonction `LENGTH` (voir Annexe).

IV Création et alimentation d'une base de données

IV.1 Organisation d'une base de données

1. Ouvrir la base de données nommée `eleves`.
2. Créer une table dont le schéma relationnel est le suivant :
   `films_nom_prenom (titre:text, realisateur:text, annee:int)` (en remplaçant nom et prenom par **votre** nom et **votre** prénom)
3. Ajouter des entrées à la table, à partir des informations fournies en Annexe. Consignes : l'attribut `realisateur` ne doit contenir qu'un seul nom et toutes les informations doivent être présentes dans la table.
4. La table précédente contient plusieurs fois des informations strictement identiques. Rechercher par exemple l'année de sortie du film *Sacré Graal*.
5. En vous inspirant de la deuxième partie du TP, déterminer comment modifier la table `films_nom_prenom` et quelle autre table créer (toujours avec vos nom et prénom comme suffixes) pour éviter les informations en doublons.
6. Mettre en œuvre une solution et effectuer une requête sur une jonction des deux tables permettant d'obtenir les noms des réalisateurs du film *Sacré Graal*.

2. Une des « actions » au dessus des résultats de la requête est faite pour cela.

IV.2 Exploitation d'une base de données

Le contenu de la table `cafes` vient de la liste[3] des noms, des adresses et des coordonnées géographiques des endroits de Paris où le café était servi à 1 euro au 15 mai 2014.

1. Combien de cafés sont situés dans le XI^{e} arrondissement ?
2. Les coordonnées géographiques du lycée Dorian sont en degrés décimaux : (latitude) 48,854563° et (longitude) 2,392678°. Pour des lieux tous situés dans une zone géographique aussi petite que Paris, on peut approximer que la distance entre deux points est directement proportionnelle à la racine carrée de la somme de la différence des longitudes au carré et de la différence des latitudes au carré : $\sqrt{(lat1 - lat2)^2 + (long1 - long2)^2}$. Quel est le café à un euro le plus proche du lycée ?

V Utilisation de bases de données avec Python (Optionnel)

1. Ouvrir Spyder et un nouveau fichier de script. Enregistrer votre fichier de script dans votre répertoire personnel. Taper dans le script les lignes suivantes :
2. Quels types d'objets Python sont `res` et `foo` ? Que contient `foo` ?
3. À l'aide d'une requête effectuée depuis le script Python, déterminer les triangles tels que $\frac{AB+AC+BC}{3} = 42$ et placer les résultats dans un fichier. Comparer avec la méthode n'utilisant que SQLite Manager (dernière question de l'exercice 1).

3. Récupérée sur http ://opendata.paris.fr

VI Annexe

VI.1 ORDER BY

La clause `ORDER BY attribut` permet de trier les résultats de la requête selon un attribut donné. Par défaut le tri est ascendant. L'option `DESC` permet de trier par ordre descendant.
`ORDER BY nom` trie la requête par ordre ascendant des valeurs de l'attribut `nom`
`ORDER BY annee DESC` trie la requête par ordre descendant des valeurs de l'attribut `annee`
Elle est toujours située à la fin de la requête.

VI.2 LIKE

L'opérateur `LIKE`, utilisé en complément de la clause `WHERE` permet de sélectionner sur la base d'un modèle (par exemple, la valeur de l'attribut commence/se termine ou comprend une séquence de caractères particulières, le « modèle »).
Syntaxe : `SELECT * FROM table WHERE attribut LIKE modele`
Modèle : le modèle est une chaîne de caractères, contenue entre guillemets, avec comme caractère joker le caractère « % ».
Exemples : `LIKE "%a"` : permet de rechercher toutes les chaînes de caractère qui se termine par un « a ».
`LIKE "a%"` : permet de rechercher celles qui commencent par un « a ».
`LIKE "%a%"` : pour rechercher celles qui utilisent le caractère « a ».
`LIKE "pa%on"` : permet de rechercher les chaînes qui commencent par « pa » et qui se terminent par « on », comme « pantalon » ou « pardon ».

VI.3 LENGTH

La fonction `LENGTH` renvoie la longueur d'une chaîne de caractères.
`LENGTH("total")` renvoie `5`.
Utilisée sur un attribut avec la clause `WHERE`, elle permet de faire des recherches avec comme critère la longueur d'un attribut de type texte ou chaîne de caractères.

VI.4 Liste de films

Voici trois films, suivi du ou des réalisateurs et de l'année de sortie.
- *Sacré Graal* de Terry Jones et Terry Gilliam sorti en 1975.
- *La Vie de Brian* de Terry Jones sorti en 1979.
- *Brazil* de Terry Gilliam sorti en 1985.