Data Analyst # 2

2024

OS & BDD

KIT APPRENANT

Sujet 2 : Base de Données

(BDD)

TYPE OF SQL JOINTS (SOURCE : HTTPS://WWW.REDDIT.COM/R/DATABASE/)

Objectifs pédagogiques

Maîtriser les concepts et l’utilisation d’un SGBD

Le stockage de l’information fait partie des challenges actuels et historiques de

l’informatique. Plusieurs générations de chercheurs en informatique ont travaillé - et travaillent

encore - à modéliser et optimiser les systèmes de stockage, selon différents types de contraintes.

Dans l’histoire de l’informatique,

le besoin d’un système de stockage performant et

présentant les caractéristiques suivantes s’est fait ressentir très tôt :

● Structuré : Un schéma clair et précis des chemins d’accès aux données est réalisable;

● Massif

: Un grand nombre de données peut être géré. Aujourd’hui, on parle de

téraoctets/jour;

● Persistant : Il reste stable et pérenne, a contrario des programmes informatiques dont

l’état est perdu après la fermeture de l’application;

● Sécurisé : Il présente des sécurités contre les failles tant au niveau logiciel (software) que

physique (hardware) --> sauvegardes fréquentes, …;

● Concurrent : La même base de données reste accessible à plusieurs utilisateurs en même

temps sans risque pour son intégrité;

● Efficient : d’abord la performance, puis la performance et finalement la performance - bons

résultats avec un minimum d’efforts;

● Fiable : Il est opérationnel 99.999999999……% du temps. Avec autant de 9, après la

virgule, que nécessaire.

Les systèmes RDBMS (Relational Database Management Systems) ou SGBD en français

ont permis de répondre à ces besoins. Ils sont issus du modèle de données relationnel décrit par

Edgar Codd en 1969.

Les SGBDR : un incontournable à maîtriser

● C’est une méthode de stockage privilégiée dans le monde industriel.

Il est à 90% probable que les systèmes informatiques avec lesquels vous

interagissez ou interagissez dans votre vie quotidienne utilisent au moins un SGBD pour

sauvegarder l’information.

● Transformation de l’informatique : d’une science de calcul vers une science de la donnée.

Autrefois,

la donnée n’avait qu’un but opérationnel précis : enregistrer une

opération, afin de faire constater un évènement/état. De nos jours, la donnée est plus qu’un

répertoire d’octets, mais une opportunité d’extraction de connaissances.

Démarche pédagogique

Itération #1

Jour 1 (AM) – Travail en équipe, par îlot, réflexion sur les mots :

- Donnée
- Modèle de données
- Structuration de données
- Base de données

Jour 1 (PM) – Une deuxième partie, en autonomie :

apprendre la manipulation des données en SQL

-

dans votre ordinateur à l’aide du site web sqlzoo.

-

Itération #2

Jour 2 - S’interfacer avec une BDD

-

- S’interfacer avec la BDD via Python : “ma première API python pour BDD”

apprendre la manipulation des données en SQL

Itération #3

Pour l’itération 3, pensez à prendre des écouteurs et des crayons!

Jour 3 et 4 – Travail en équipe, par îlot.

◦

Lecture d’un diagramme UML

◦ Complétion d’un diagramme UML

◦ Créer un script pour alimenter une BDD

Compétences

À la fin de ce module vous serez en mesure de :

● Exploiter un SGBD à l’aide du langage de programmation SQL pour :

○ Définir un schéma relationnel (Data Definition Language)

○ Manipuler la donnée (Data Manipulation Language)

● Vous connecter à un SGBD :

○ À l’aide d’un client GUI

○ À l’aide du langage de programmation python

● Maîtriser les diagrammes de classes du langage de modélisation unifié (UML) pour:

○ Comprendre l’architecture d’une BDD relationnelle

○ Concevoir une BDD relationnelle

● (Optionnel) Avoir des notions de normalisation :

o 1NF, 2NF, 3NF

o 4NF

3

Itération 1: Introduction SQL (1 jour)

Objectif(s):

● Apprendre à utiliser le sous-ensemble du langage SQL pour la manipulation de données

● (Optionnel) : Avoir des notions d'algèbre relationnelle

Travaux pratiques

1. Apprenez les bases du langage SQL en allant sur SQLZoo : https://sqlzoo.net/.

Réalisez les exercices jusqu’au numéro 9

Pour éviter de perdre votre travail. Vous pouvez créer un compte avec votre e-mail Campus.

2. Les bases de données (BDD) sont un univers en soi. Vous avez fait vos premiers pas dans

cet univers avec les commandes SQLZoo. Mais avant d’aller plus loin dans l’usage des

BDD, il est nécessaire de comprendre comment elles sont structurées et comment elles

fonctionnent. Plus précisément nous nous intéresserons aux BDD relationnelles.

Créer un mémo au sein duquel vous proposez une définition pour les mots suivants :

-
-
-
-
-
-
-
-
-
-

un modèle des données relationnel

une relation

un attribut

un type d’attribut.

une clé primaire

une clé secondaire

une clé étrangère

un système de gestion de base de données (SGBD)

Langage déclaratif

langage normalisé

3. Une fois ce mémo fini individuel fini, discutez avec votre ilôt / groupe si vous vous accordez

avec les définitions. Faites un mémo commun !

Ressources et lectures

SQL est un langage issu de l’algèbre relationnelle. Si vous avez envie d’un peu de théorie,

vous pouvez commencer à lire ou écouter les ressources suivantes :

R1.1 Databases - University of Toronto

https://slideshowes.com/doc/159317/relational-algebra---university-of-toronto

R1.2 Relational Algebra – Stanford

https://www.youtube.com/watch?v=tii7xcFilOA

https://www.youtube.com/watch?v=GkBf2dZAES0

SQL est un langage assez intuitif. Il est cependant possible que vous bloquiez au niveau

des jointures. Dans ce cas, vous pouvez consulter :

R1.3 Jointures – Université de Lyon

https://perso.liris.cnrs.fr/fabien.duchateau/ens/BDW1/cm/bd-sql-jointures.pdf

https://sql.sh/2401-sql-join-infographie

Itération 2 : S’interfacer à une BDD (1 jour)

Objectifs proposés

- Installer une base de données relationnelle
- Se servir d’une BDD à l’aide des différents outils :
  - à l’aide d’une interface visuelle (GUI),
  - à l’aide du langage Python et avec la librairie Pandas.
  - (optionnel) quels outils pour quel besoin ?

Travaux pratiques avec SQLite

Vous avez découvert le SQL via SQLZoo. Derrière SQLZoo se cachent plusieurs outils.

Lorsque que vous avez réalisé des requêtes SQL, vous les avez réalisées dans un interpréteur de

requête, et cet interpréteur s’est ensuite adressé à une BDD relationnelle SQL. Le tout fonctionne

sur un serveur commun. Nous allons découvrir comment installer et faire fonctionner ces différents

outils.

Dans un premier temps, nous allons utiliser le logiciel SQLite. SQLite est un SGBD dont la

qualité principale est la légèreté du système. Il est très probable que certaines des applications

mobiles installées dans votre téléphone utilisent SQLite pour stocker l'information.

Installation de SQLite

Selon le type de machine sur lequel vous êtes, Ubuntu ou Windows, les façons

d'installer sont différentes. Ci-dessous les méthodes d'installation pour les deux systèmes : Ubuntu

puis Windows.

Ubuntu (Linux / WSL) :

Sur Ubuntu, il y a deux solutions. Soit utiliser le terminal et les lignes de commande,

ou installer le logiciel via l'application graphique Ubuntu. Il est recommandé d'utiliser la première

approche pour sa simplicité.

● Ouvrez un terminal puis réaliser les commandes suivantes (recommandé):

$ sudo apt-get install sqlite3

$ sudo apt-get install sqlitebrowser

OU

● ouvrez l'app Ubuntu Software Application > Chercher DB Browser for SQLite > Installer

Windows :

Pour l’installer dans une machine Windows :

1. Téléchargez le ZIP DB Browser via la page suivante : sqlitebrowser.org

2. Décompresser le fichier Zip

5

3. Cliquer sur l’exécutable DB Browser for SQLite

Ici, vous avez installé SQLite en même temps que «DB Browser for SQLite ». «DB Browser

for SQLite » est un client visuel (i.e. GUI) pour les BDD de type SQLite. Un client visuel est un

programme capable de se connecter à un SGBD, et de simplifier notamment

de

compréhension d’une BDD.

l’étape

S’interfacer avec la BDD

1. Téléchargez les données BillBoard 200 qui se trouvent dans la ressource R2.1

2. Connectez-vous à la BDD à l’aide de votre client (i.e. DB Browser for SQLite)

3. Pour comprendre les données, référez-vous aux ressources R2.2 et R2.3

Travaux pratiques sur la BDD

Maintenant que vous avez installé la BDD, et que vous êtes connecté à cette

dernière, créez un mémo et répondez aux questions suivantes :

1. Trouvez le top 10 albums des 20 dernières années.

2. Trouvez l'album qui est resté le plus longtemps numéro 1. Refaire cette même recherche

mais pour les 9 autres albums.

3. Réalisez un Fact checking sur 3 informations de votre choix de la section "All-Time Billboard

200 achievements (1963-2015)" de la ressource R2.2. Vérifiez à l'aide de requêtes SQL que

les informations de la section sont correctes.

4. Optionnel : reproduire à la fin du mémo le schéma de la BDD.

Remarques :

1. Les requêtes sont sensibles à la casse.

2. Certaines requêtes peuvent générer des doublons. Attention de les supprimer avant de

3.

concaténer des résultats.

Il est recommandé de toujours réaliser vos requêtes avec une limite avant de vous lancer

dans une requête complète.

Aides et ressources

Quelques pistes de réflexion question code (sans nettoyage initial):

6

7

Travaux pratiques avec Python

Vous avez réalisé un travail avec SQLite et l’interfaçage via DB Browser fourni avec.

Maintenant, réalisons le même travail, mais en utilisant Python. Cette étape vous permettra

d’envisager le développement d’application ou d’exploration basé sur une BDD SQL avec Python.

S’interfacer avec Python

1. Créer un jupyter notebook

2. Connectez-vous à la base de données à l’aide du module sqlite3.

a. Utilisez la documentation officielle de SQLite pour comprendre le module sqlite sur

Python.

b. Créez une fonction ou une classe qui vous permet de facilement de :

i. Se connecter à votre BD

ii. Effectuer une requête

iii. Récupérer le résultat

3. Effectuez des requêtes SQL depuis Python.

a. Pour tester le bon fonctionnement

b. Puis pour répondre aux questions proposées précédemment avec DB Browser (voir

Travaux pratiques avec SQLite).

Remarque : si tout se passe bien, ce sera presque du copier / coller.

S’interfacer avec DataFrame

Il est possible de créer un dataframe à partir d’une requête SQL. Voir la documentation de

Pandas. Une fois que vous avez pris en main la documentation, réalisez les tâches suivantes :

Pour chacune des questions, vous pouvez soit réaliser une seule requête SQL et utiliser

ensuite Pandas pour manipuler les données, ou réaliser pour chaque question une requête fine et

précise SQL et utiliser ensuite Pandas pour manipuler les sous datasets. La seconde approche est

la table

recommandée. Les données sur

acoustic_features.

les caractéristiques des chansons sont sur

1. Effectuez la moyenne par année de toutes les caractéristiques. Quelle est la tendance que

vous constatez ?

2. Quelle est l’année dont le niveau sonore «loudness » a été le plus haut ?

3. Quelle est la clé musicale la plus populaire - en prenant en compte le mode (e.g. majeur,

mineur) ?

4. Optionnel

: Créer un ou plusieurs

schéma via le paquet de votre choix

(matplotlib/seaborn/plotly) pour représenter ces tendances et vos conclusions.

a) Quel graphique vous permet de mieux comprendre la popularité des tonalités, en faisant

la différence entre majeur et mineur par note ?

Références / Ressources

R 2.1 – Données Billboard 200

https://www.dropbox.com/s/z6bcckb74k0d97f/billboard.zip?dl=0

R 2.2 – C’est quoi Billboard 200 ?

https://en.wikipedia.org/wiki/Billboard_200

R 2.3 – Documentation sur les features de chanson par Spotify

8

https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

R 2.4 – Module Sqlite sur Python 3.6

https://docs.python.org/3.6/library/sqlite3.html

R 2.5 – Dataframe from SQL

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_sql.html

9

Itération 3 Créer une BDD (1 jour)

Objectifs proposés

- Savoir lire un diagramme de BDD
- Traduire un diagramme en schémas (instructions SQL)
- Créer une BDD SQL
- Savoir insérer des données:

  - à l'aide de Python et Pandas
  - d'un fichier SQL (bulk insert)

Travaux pratiques

Pour cette itération, nous allons toujours utiliser SQLite.

Savoir lire et écrire un diagramme UML

Lisez les ressources suivantes :

1. Diagrammes des classes (UML) – Stanford :

vidéo 1 : https://www.youtube.com/watch?v=LmS4Y99fNaQ

vidéo 2 : https://www.youtube.com/watch?v=X89KLfrNOPo

2. Référence diagrammes de classes – Microsoft. Le document commence à dater mais reste

simple pour rentrer dans le sujet.

Concevoir un diagramme UML (travaux en groupe)

Regardez le diagramme de classes UML (voir figure 3.1). Ce diagramme correspond à la

structuration de la donnée pour le système de transport d’une ville.

Remarque : Système de transport != Système GPS.

Il ne s’agit donc pas d’un diagramme concernant les résultats d’un calculateur d’itinéraire tels que

Google Maps.

Figure 3.1 - Structuring transport systems

Le diagramme 3.1 manque l'information sur la multiplicité des associations. C'est une

information capitale pour le diagramme UML. Avant de créer le diagramme, créer un mémo dans

lequel vous définissez et expliquez les multiplicités des associations suivantes :

1. Agency – Routes
2. Trips – StopTimes
3. StopTimes – Stops
4. Trips – Calendar
5. Trips - Frequencies

Vous pouvez vous aider des deux ressources suivantes :

1. Référentiel GTFS : https://developers.google.com/transit/gtfs/reference
2. GTFS en dessin : https://xang1234.github.io/isochrone/

Création d'un schéma pour BDD (en groupe)

Toujours en groupe, traduisez votre diagramme de classes UML en un fichier SQL.

1. Créez votre schéma SQL et nommez-le : "gtf_schema.sql".

2. Traduisez les classes suivantes sans oublier les clés primaires et secondaires:

a) Agency

b) Routes

c) Trips

d) StopTimes

e) Stops

Pour vous aider, voici un exemple :

i. Pour créer votre schéma, inspirez vous du référentiel GTFS de google

ii. Pour créer votre fichier .sql, voici deux exemples :

https://gist.github.com/denysvitali/cf33fb42c3cfd26c0aabc8e849f8252d

https://www.dropbox.com/s/t4s7fuo0fxynjqk/schema.sql?dl=0

Création de votre BDD (en individuel)

1. Une fois votre schéma finalisé, construisez la BDD.

a) Créer une BDD nommée “gtfs_tag.db”

b) Exécuter votre schéma sur votre BDD

c) Vérifier, à l’aide du GUI, que vos instructions ont bien été prises en compte

Pour vous aider à exécuter, voici deux ressources sur l’exécution avec SQLite :

i.

https://www.youtube.com/watch?v=giAMt8Tj-84

ii. https://www.youtube.com/watch?v=xyCxLKEQPAs

Insertion des données via Pandas (en autonomie individuelle)

Maintenant que vous avez réalisé le schéma et créé la base de données. Vous allez

travailler de nouveau individuellement. Réalisez les tâches suivantes :

1. Récupérer les données GTFS du réseau TAG

2. Utilisez Pandas pour insérerez l'information correspondant aux classes :

a) agency

b) stops

Ressources: pandas to_ sql

ATTENTION: IL EST INTERDIT D'UTILISER L'OPTION if_exists='replace'. Si vous avez un

problème de nom de colonne ou d'index, ne choisissez pas la facilité qui vous mènera à

l'échec.

Félicitations !!! Vous venez de créer votre première BDD SQL. Elle n'est pas complète

mais vous avez dorénavant l'ensemble des outils en main pour créer une BDD. Voyons un cas plus

concret avec le réseau TAG.

Insertion des données via les outils SQL (bulk insert)

Votre mission sera d’insérer les données pour les classes suivantes des déplacements

(trips) et des horaires d’arrêt (stoptimes). Le diagramme TAG utilise le format GTFS de Google.

Pour réaliser ces tâches nous allons vous guider pas à pas. Nous allons aborder comment

créer le fichier SQL et exécuter le fichier dans la BDD, ainsi que comment nous allons alimenter la

BDD.

Réalisez les tâches suivantes :

1. Créer un notebook

2.

Implémenter une fonction qui génère UNE commande d’insertion SQL.

1. La signature de la fonction est la suivante :

def gen_insert_query(table_name:str, a_dict:dict) -> str

2. Les paramètres sont :

1. Tablename : le nom de la table (e.g. gtfs_stops)

2. a_dict : dictionnaire Python

3. La fonction doit retourner une chaine de caractère qui représente le code SQL

d’insertion

3.

Implémenter une fonction qui génère DES commandes d’insertion SQL

1. La signature de la fonction est la suivante :

def get_insert_queries(tablename:str, df: pd.DataFrame) -> list

2. Les paramètres sont :

1. Tablename : le nom de la table (e.g. gtfs_stops)
2. df : le DataFrame pandas

3. La fonction doit retourner une chaine de caractère qui représente le code SQL

d’insertion

4.

implémenter une procédure qui crée un fichier SQL

1. La signature de la fonction est la suivante :

def gen_insert_file(filename, tablename, df)

2. Les paramètres sont :

filename : le nom du fichier (e.g. insert_stops.sql)

tablename : le nom de la table (e.g. gtfs_stops)

1.

2.

3. df : le DataFrame pandas

3. La fonction doit créer un fichier .sql sur le disque dur.

Remarque: afin de rendre la transaction de votre fichier SQL efficace, regardez les mots

clés BEGIN et COMMIT du langage SQL.

5. Exécutez votre fichier SQL sur la BDD en utilisant la commande CLI .read de sqlite.

Félicitation !! Vous venez de créer vos premiers outils pour alimenter votre BDD.

13

Réflexions sur les travaux réalisés (mémo)

Créer un mémo à l’aide des questions ci-dessous. Vous avez inséré des données avec

deux méthodes différentes : Pandas et Bulk insert. Selon vous :

1. Quelle méthode est plus rapide et facile à implémenter ?

2. Quelle méthode est plus rapide pour insérer l'information ?

3. Dans quel scénario préconisez-vous l'utilisation d'une méthode ou l'autre ?

(Optionnel) Créer un calculateur d'itinéraire

Lorsque vous utilisez Google Map entre deux points, ce dernier vous propose des itinéraires

"à pied", "à vélo" ou "en transport en commun". Vous allez construire un exemple où à partir de la

position GPS de départ et d'arrivée, vous allez proposer un itinéraire pour le transport en commun.

Donnée de départ :

• A - Un point GPS de départ à Grenoble, en format tuples de float;

• B – Un point GPS d'arrivée à Grenoble, en format tuples de float;

• H – une heure de part en format datetime;

Calculez l'itinéraire des lignes TAG à prendre !

(Optionnel) Pour aller plus loin

Si fini, suivez les liens…:

● https://drive.google.com/drive/folders/1uAVFUAS-tnDwRrbMlxjBmUP8ONlYzudu?usp=sharin

g

● https://www.youtube.com/watch?v=ruz-vK8IesE

● https://towardsdatascience.com/arm-yourself-to-select-your-first-database-8bc9008bf8ec

● https://www.ionos.fr/digitalguide/sites-internet/developpement-web/diagramme-de-classes-um

l/

Références / Ressources

R 3.1 – Diagrammes des classes (UML) - Stanford

https://www.youtube.com/watch?v=LmS4Y99fNaQ

https://www.youtube.com/watch?v=X89KLfrNOPo

R 3.2 – Référence diagrammes de classes – Microsoft

https://docs.microsoft.com/en-us/visualstudio/modeling/uml-class-diagrams-reference?view=vs-

2015

R 3.3 – Référentiel GTFS

https://developers.google.com/transit/gtfs/reference

R 3.4 – GTFS en dessin

https://xang1234.github.io/isochrone/

R 3.5 – Exemple de schéma sql

https://www.dropbox.com/s/t4s7fuo0fxynjqk/schema.sql?dl=0

R 3.6 – Introduction à SQLite

https://www.youtube.com/watch?v=giAMt8Tj-84

14

https://www.youtube.com/watch?v=xyCxLKEQPAs

R 3.7 – GTFS Réseau TAG

https://www.data.gouv.fr/fr/datasets/horaires-theoriques-du-reseau-tag

15