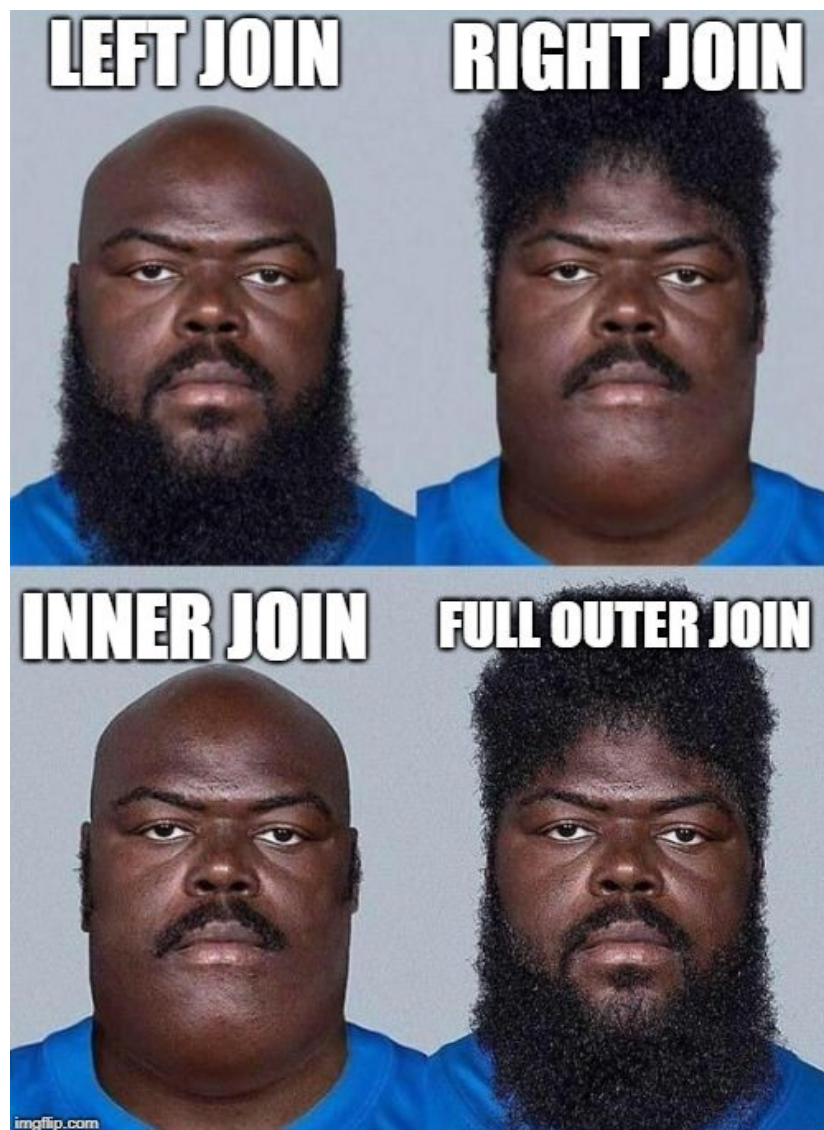




le
campus
numérique



Base de données (BDD)



TYPE OF SQL JOINTS (SOURCE : [HTTPS://WWW.REDDIT.COM/R/DATABASE/](https://www.reddit.com/r/DATABASE/))

Objectifs du module

Maitriser 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 - ou travaillent - à modéliser et optimiser les systèmes de stockage, selon différentes types de contraintes.

Dans l'histoire de l'informatique, très tôt il y a eu besoin d'un système de stockage ayant les caractéristiques suivantes :

- Structuré : avoir un schéma clair et précis des chemins d'accès aux données
- Massive : gérer énormément de données (aujourd'hui on parle de téraoctets /jour)
- Persistant : à contrario des programmes informatiques dont l'état est perdu après la fermeture de l'application.
- Sécuré : contre les failles tant au niveau logiciel (software) que physique (hardware)
- Concurrent : capable de gérer plusieurs utilisateurs en même temps
- Efficient : d'abord la performance, puis la performance et finalement la performance.
- Fiable : opérationnel 99.99999999% du temps.

Les systèmes qui ont permis de répondre à ces besoins sont les RDBMS (*Relational Database Management Systems*) ou SGBD en français. Ils sont issus du modèle de données relationnel décrit par Edgar Codd en 1969.

Pourquoi apprendre les SGBDR ?

Méthode de stockage privilégié dans le monde industriel. Il est à 90% probable que les systèmes informatiques avec lesquels vous interagissez dans votre vie quotidienne utilisent 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 simple : 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 connaissance

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 Definiton Language*)
 - Manipuler la donnée (*Data Manipulation Language*)
- Se connecter à un SGBD :
 - À l'aide d'un client GUI
 - À l'aide du langage de programmation python
- Maitriser les diagrammes de classes du langage de modélisation unifié (UML) , pour:
 - Comprendre l'architecture d'une BDD relationnelle.
 - Concevoir une BDD relationnelle.
- (Bonus) avoir des notions de normalisation :
 - 1NF, 2NF, 3NF
 - 4NF

Modalités

Le timing prévu

Itération #1

- Jour 1 – matin : Travail en équipe, par ilot, réflexion sur les mots :
 - Donnée
 - Modèle de données
 - Structuration de données
 - Base de données
- 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 : début de l'itération 2.

Itération #3

- Jour 3 – matin: travail en équipe, par ilot.
 - 👍 *pour le jour 3, pensez à prendre des écouteurs et des crayons !*
 - Lecture d'un diagramme UML
 - Complétion d'un diagramme UML

Itération 1

Introduction : BDD et SQL – (1 jour)

Objectif proposé :

- Apprendre à utiliser le sous-ensemble du langage SQL pour la manipulation de données.
- (Optionnel) : Avoir des notions d'algèbre relationnelle

Vocabulaire

Le **modèle des données relationnel** peut se décrire comme suit :

- C'est un ensemble de **relations** (*i.e. des tables*)
- Chaque relation contient un ensemble d'**attributs** (*i.e. colonnes*)
- Une relation contient des **tuples** (*i.e. lignes*) avec des valeurs pour chaque attribut
- Chaque attribut est **typé**, et le type est souvent *atomique* (e.g. *entier, réel, etc.*)

Un schéma est la description structurelle d'une base relationnelle

Une clé est un attribut (ou ensemble d'attributs) qui permet d'accéder à un ensemble de tuples.

Une clé primaire est un attribut (ou ensemble d'attributs) qui permet d'identifier un seul et unique tuple.

SGBD - Le *système de gestion de base de données relationnelle* (suivant abrégé tout simplement *SGBD*) est un système qui permet de stocker l'information sous un modèle de données relationnelles.

Quelques SGBD très connus et utilisés sont : Oracle, SQL Server, PostgreSQL, MySQL and SQLite.

SQL - *Structured Query Language* est le langage qui permet d'exploiter un SGBD.

SQL est un langage déclaratif : nous énonçons ce que l'on cherche et non pas le comment ; c'est le travail du SGBD de trouver l'algorithme pour donner le bon résultat de manière optimale. De ce fait, on dit que SQL est langage de haut niveau.

SQL c'est un langage normalisé : en théorie une requête doit pouvoir s'exécuter sur les différents SGBD. Dans la pratique, selon la complexité de la requête, elle sera modifiée pour s'ajuster à l'opinion du SGBD ; les changements restants, en règle générale, simples.

SQL est un langage issu de l'algèbre relationnelle.

Consignes

Aller sur le site <https://sqlzoo.net/> afin de réaliser tous les exercices à l'exception de : 9. Window function, 11. Tutorial Student Records, 12. Tutorial DDL.

Si vous avez besoin d'un peu de théorie avant de vous lancer dans la pratique, vous pouvez commencer par des ressources d'algèbre relationnelle : R1.1 et R1.2

SQL est un langage assez intuitif. Il est cependant possible que vous bloquiez au niveau des jointures. Dans ce cas, vous pouvez consulter la R1.3.

Références / Ressources

R1.1 Relational Algebra - University of Toronto

<http://www.cs.toronto.edu/~ryanjohn/teaching/csc43-s12/lectures/c43-ra-v05.pdf>

R1.2 Relational Algebra – Stanford

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

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

R1.3 Jointures – Université de Lyon

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

Itération 2

S'interfacer à une BDD - (1 jour)

Objectifs proposés :

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 ?

Consignes

Pour cette itération, nous allons utiliser le logiciel SQLite. SQLite est un SGBD, sa 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. Pour l'installer sous votre machine, exécutez l'instruction suivante dans votre terminal :


```
$ sudo apt-get install sqlite3
```

Nous allons aussi installer «DB Browser for SQLite », c'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 l'étape de compréhension d'une BDD. Pour l'installer, les instructions sont simples :

```
Ouvrir l'app Ubuntu Software Application > Chercher DB Browser for SQLite > Installer
```

C2.1 - S'interfacer avec un GUI

1. Téléchargez les données 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
4. *Fact checking* . Vérifiez à l'aide de requêtes SQL les informations de la ressource R2.2.
 - a. Plus grand nombre de semaines numéro 1
 - b. Top 10 albums artists of All-Time

 Attention certains attributs doivent être modifiés afin que vos requêtes s'exécutent correctement.

C2.2 - S'interfacer avec Python

1. Créer un jupyter notebook
2. Connectez-vous à la base de données à l'aide du module sqlite3. Utilisez la ressource R1.4.
3. Effectuez des requêtes SQL depuis python pour répondre aux questions suivantes :
 - a. Effectuez la moyenne par année de la caractéristique « acoustiness ». Quelle est la tendance de cette caractéristique ?
 - b. Quelle est l'année dont le niveau sonore «loudness » a été le plus haut ?
 - c. Quelle est la clé musicale la plus utilisée - en prenant en compte le mode (e.g. majeur, mineur) ?

C2.3 - S'interfacer avec Pandas

Il est possible de créer un *dataframe* à partir d'une requête SQL.

Créez une seule requête SQL, et à l'aide de pandas répondez aux questions proposées en C2. La ressource R2.5 vous sera très utile.

 *Bonus, tracez des courbes*

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://fr.wikipedia.org/wiki/Billboard_200

R 2.3 – Documentation sur les features de chanson par Spotify

<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

Itération 3

Modélisation d'une BDD - (1.5 jour)

Objectifs proposés :

- Lire un diagramme de BDD. Une BDD de type « production » est toujours accompagnée d'un diagramme UML qui permet de comprendre sa structure.
- Traduire un diagramme en schéma (instructions SQL).
- Savoir insérer des données à l'aide d'un fichier SQL (*bulk insert*) et à partir de python.

Consignes

C3.1 – Lire et compléter un diagramme UML

1. Prenez du temps pour étudier les ressources R3.1 et R3.2.
2. À 10h30, vous allez recevoir un diagramme de classes, la suite se déroule en équipe, par ilot.
3. Le diagramme manque d'une information clé : la multiplicité des associations. En équipe, retrouvez les multiplicités des associations suivantes :
 - a. Agency – Routes
 - b. Trips – StopTimes
 - c. StopTimes – Stops
 - d. Stops – Transfer

Pour mieux comprendre la logique métier, vous pouvez vous appuyer de la ressource R3.4 et R3.5 .

C3.2 – Création d'un schéma

Toujours en équipe, travaillez ensemble pour traduire votre diagramme de classes en un fichier SQL qui sera nommé `gtfs_schema.sql` . Les classes que vous allez traduire sont les suivantes :

1. Agency
2. Routes
3. Trips
4. StopTimes
5. Stops

Une fois que votre schéma (i.e. fichier `gtfs_schema.sql`) est finalisé :

1. Créez une BDD nommée `gtfs_tag.db`
2. Exécutez votre schéma sur votre BDD
3. Vérifiez, à l'aide du GUI, que vos instructions ont été bien prises en compte.

Pour réussir cette étape, privilégiez la compréhension de la ressource R3.4. Utilisez R3.6 pour commencer votre schéma , et R3.7 pour l'exécuter.

C3.3 – Insertion des données (*bulk insert*)

En autonomie, vous allez travailler avec les données GTFS du réseau TAG (ressource R3.8). Vous allez insérer l'information concernant les arrêts (fichier stops.txt).

Voici les étapes classiques pour insérer des données:

1. Créer, à l'aide de python, un fichier SQL (e.g. insert_stops.sql)
2. Exécuter le fichier dans la BDD

Dans jupyter notebook, créez une procédure qui crée un fichier SQL, sa signature est la suivante :

```
def gen_insert_file(filename, tablename, df)
```

Les paramètres de cette procédure sont :

- *filename* : est le nom de fichier (e.g. insert_stops.sql)
- *tablename* : le nom de la relation où nous allons insérer nos tuples
- *df* : un dataframe pandas

La fonction `gen_insert_query` de la ressource R3.9 vous sera utile.

C3.4 – Insertion des données via Pandas

En autonomie, vous allez insérer l'information qui correspond au fichier :

- agency
- stops
- route

Pour ce faire, vous allez utiliser la méthode `to_sql` (ressource R3.10) des pandas *dataframe*.

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 3.4 – Référentiel GTFS

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

R 3.5 – GTFS en dessin

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

R 3.6 – Exemple de schéma sql

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

R 3.7 – Introduction à SQLite

<https://cs.stanford.edu/people/widom/cs145/sqlite/SQLiteIntro.html>

R 3.8 – GTFS Réseau TAG

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

R 3.9 – SQL Utils

<https://www.dropbox.com/s/4c7i422c3qiwe4p/sqlutils.py?dl=0>

R 3.10 – Référence de la méthode to_sql

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