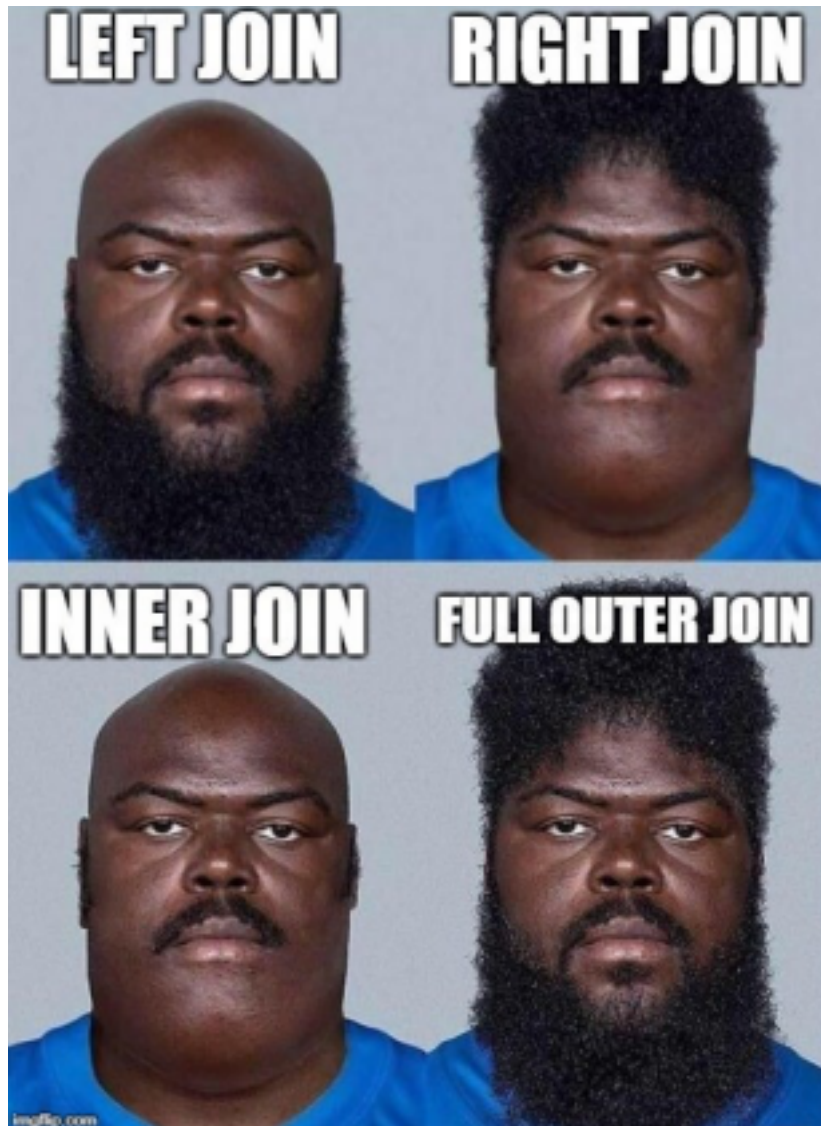




Base de Données (BDD)



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

Objectifs du module

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.

Pourquoi apprendre les SGBDR ?

Méthode de stockage privilégiée dans le monde industriel

Il est à 90% probable que les systèmes informatiques avec lesquels vous interagissez ou interagirez 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.

Compétences

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

- Comprendre une partie des généralités et la terminologie associées aux domaines de l'architecture et l'ingénierie Data
- 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 :
 - ○ *1NF, 2NF, 3NF*
 - ○ *4NF*

Organisation

Itération #1 (½ jour PM)

- Travail en équipe, par îlot, réflexion sur les mots :
 - Données
 - Modèle de données
 - Structuration de données
 - Base de données
 - SQL / noSQL

Itération #2 (1 jour)

- Synthèse des notions évoquées lors de l'itération 1
- En autonomie : Manipulation des données en SQL (site web sqlzoo)

Itération #3 (1 jour)

- AM: S'interfacer avec une BDD
- PM: Travail en équipe, par îlot
 - 💡👉 Pour le jour 3, pensez à prendre des écouteurs et des crayons!
 - Lecture d'un diagramme UML
 - Complétion d'un diagramme UML

Itération #4 (1 jour)

- Créer un script pour alimenter une BDD

Itération 1 ($\frac{1}{2}$ jour)

Introduction : BDD, SQL et noSQL

Objectif(s):

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

Quelques éléments de langage pour débiter..à vous de compléter!

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* est un système qui permet de stocker l'information sous un modèle de données relationnelles.

Parmi les SGBD très connus et utilisés se trouvent : Oracle, SQL Server, PostgreSQL, MySQL et 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 un langage de haut niveau.
- **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 sont, en règle générale, simples.
 - La norme ISO 9075 définit des fonctionnalités obligatoires et optionnelles du langage. Les SGBD implémentent ces fonctionnalités, *a minima* celles obligatoires, et ajoutent généralement leurs propres spécificités.

SQL est un langage issu de l'algèbre relationnelle.

Consignes

- Répondre aux questions suivantes:
 - Qu'est-ce qu'une donnée?
 - Comment définir une base de données?
 - Existe-t-il différents types de données? Si oui, lesquels connaissez-vous?
 - Pourquoi utiliser un SGBD?
 - Quelles sont les principales différences entre SQL et noSQL? Pourquoi choisir l'un plutôt que l'autre? Des exemples de cas réels ?
- Réaliser un tableau de synthèse des avantages et inconvénients de différents SGBD.
- Petit test de fin de journée: <https://librecours.net/module/bdd0/intro-sgbd/quiz.xhtml>
→ *objectif: > 80%*

Quelques éléments de lecture

Si vous avez envie d'un peu de théorie, vous pouvez commencer à lire ou écouter les ressources suivantes : R1.1 et R1.2.

Références / Ressources

R1.1 Databases

<https://datascientest.com/base-de-donnees-definition>

[Database systems - Cornell University Course](#)

R1.2 Relational Algebra

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

<https://www.youtube.com/watch?v=tii7xcFiIOA>

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