|  |  |
| --- | --- |
|  | Introduction aux bases de données |

|  |  |
| --- | --- |
| **Auteur** | Frédéric jamoulle et Raphaël Baroni |
| **Date** | 01.01.2021 |
| **Référence** | LESSONS-DB-01 |
| **Version** | 0.3 |
| cc logo https://creativecommons.org/images/deed/attribution_icon_white_x2.png https://creativecommons.org/images/deed/sa_white_x2.png  **Attribution - Partage dans les Mêmes Conditions 4.0 International**  **(CC BY-SA 4.0)** | |

###### Table des matières

[1 Contrat didactique 4](#_Toc62721218)

[1.1 Prérequis et articulation avec le cursus 4](#_Toc62721219)

[1.2 Compétences de formation développées 4](#_Toc62721220)

[1.3 Objectifs de l’activité d’enseignement 4](#_Toc62721221)

[1.4 Présentation du contenu, du contexte 4](#_Toc62721222)

[1.5 Organisation du travail / attentes vis-à-vis des étudiants 5](#_Toc62721223)

[1.6 Modalités et critères d’évaluation 5](#_Toc62721224)

[1.7 Supports de cours 6](#_Toc62721225)

[1.8 Prise de contact avec l’enseignant 6](#_Toc62721226)

[2 Ressources générales 8](#_Toc62721227)

[3 Outils 9](#_Toc62721228)

[3.1 Introduction 9](#_Toc62721229)

[3.2 XAMPP 9](#_Toc62721230)

[3.3 MySQL Workbench 9](#_Toc62721231)

[4 Définition du contexte d’utilisation des bases de données et des technologies associées 10](#_Toc62721232)

[4.1 Définitions générales 10](#_Toc62721233)

[4.2 Contexte d’utilisation des bases de données 10](#_Toc62721234)

[4.3 Les différents systèmes de gestion de bases de données actuels 13](#_Toc62721235)

[5 Mise en œuvre de BD existantes : initiation pratique aux bases de données relationnelles et à leur application sur un réseau 21](#_Toc62721236)

[5.1 Introduction 21](#_Toc62721237)

[5.2 Introduction à la notion de réseaux, d’applications clients / serveur et de sécurisation 21](#_Toc62721238)

[5.3 Introduction aux définitions et manipulations de données 27](#_Toc62721239)

[5.4 Installation de la base de données existante « sakila » 52](#_Toc62721240)

[6 Méthodologie et outils pour le développement d’un modèle de données et sa BD 54](#_Toc62721241)

[6.1 Introduction 54](#_Toc62721242)

[6.2 Spécification des besoins 55](#_Toc62721243)

[6.3 Modèle conceptuel des données 56](#_Toc62721244)

[6.4 Modèle physique des données 59](#_Toc62721245)

[6.5 Implémentation et déploiement de la DB 60](#_Toc62721246)

[7 Création d’une application multimédia comprenant une base de données et son interface web 62](#_Toc62721247)

[7.1 Développement d’une nouvelle BD 62](#_Toc62721248)

[7.2 Création d’une interface web personnalisée pour une BD 63](#_Toc62721249)

[7.3 Projet : création d’une application multimédia personnalisée 66](#_Toc62721250)

[8 Bibliographie 67](#_Toc62721251)

# Contrat didactique

|  |  |
| --- | --- |
| **Activité d’enseignement** | Informatique : base de données |
| **Section** | Bachelier en techniques graphiques |
| **Année** | 1 |
| **ETCS** | 3 |
| **Heures** | 36 |

## Prérequis et articulation avec le cursus

Le cours est placé en 1ère année du cursus. Il fournit les notions de bases de données nécessaires aux cours de 2e année (Langages et techniques de programmation, Applications multimédia).

L’accès au cours est facilité par la maîtrise de l'utilisation d'un PC et de l'environnement Windows.

## Compétences de formation développées

A la fin du cours, l'étudiant(e) sera capable de :

* Configurer et mettre en œuvre des outils informatiques associés à une base de données existante.
* Concevoir, implémenter et opérer une base de données répondant à un contexte opérationnel donné.

## Objectifs de l’activité d’enseignement

* Apprentissage du contexte d’utilisation des bases de données.
* Mise en œuvre d’outils informatiques actuels traitant de bases de données.
* Utilisation de l’anglais pour supporter le développement d’applications multimédia.

## Présentation du contenu, du contexte

* Définition du contexte d’utilisation des bases de données et des technologies associées.
* Mise en œuvre de BD existantes : initiation pratique aux bases de données relationnelles et à leur application sur un réseau.
* Méthodologie et outils pour le développement d’un modèle de données et sa BD.
* Si le temps nous le permet, création d’une application multimédia comprenant une base de données et son interface web.

## Organisation du travail / attentes vis-à-vis des étudiants

Le cours a lieu au laboratoire d’informatique, un PC étant mis à la disposition de chaque étudiant. Il se présente sous forme de sessions théoriques, d’ateliers que l’étudiant réalise à son rythme.

Pour pouvoir suivre le cours, l’étudiant a besoin de l’accès à Microsoft Teams. Grâce à Microsoft Teams, l’étudiant bénéficie du matériel pédagogique (syllabus, solutions d’exercices…) et peut soumettre le résultat de ses travaux.

## Modalités et critères d’évaluation

Les examens ont lieu au laboratoire. Ils comportent une partie théorique (connaissances) et une partie pratique (compétences). La note de la partie théorique est obtenue sur base de questions à choix multiples ou à réponses courtes. La partie pratique concerne la mise en œuvre de BD de type MariaDB / MySQL.

| **Thème** | **Points** | **Sous-thème** |
| --- | --- | --- |
| Compétence N°1 : mise en œuvre d’une BD existante | 9 | Déploiement d’une BD existante sur MariaDB (XAMPP, MySQL Workbench) |
| Gestion d’accès à une BD à partir d’une interface Web (phpMyAdmin) |
| Création d’un modèle de données à partir d’une BD (MySQL Workbench) |
| Modification de la structure d’une BD / synchronisation d’un modèle de données avec la BD (MySQL Workbench) |
| Manipulation de données : requêtes SQL de base sur une table (MySQL Workbench ou phpMyAdmin) |
| Manipulation de données : requêtes SQL de base sur plusieurs tables / jointures |
| Manipulation de données : requêtes SQL / les groupes et les résultats |
| Insertion de données à partir d’un tableau Excel (MySQL Workbench) |
| Compétence N°2 : conception d’une BD et mise en œuvre | 8 | Spécification des besoins (texte et/ou diagramme UML) |
| Création d’un modèle conceptuel des données (diagramme UML sous draw.io) |
| Création d’un modèle physique des données (MySQL Workbench) |
| Implémentation et déploiement de la BD (MySQL Workbench) |
| Sécurisation des accès à la BD (phpMyAdmin) |
| Optionnel : Création d’une interface web personnalisée pour une BD. |
| Connaissances (théorie) | 3 |  |
| **TOTAL** | **20** |  |

## Supports de cours

* Ce syllabus « Cours d’introduction aux bases de données : notions et exercices » [LESSONS-DB-01]
* Une présentation [LESSONS-DB-02]
* DB instructions & exercices.docx
* Microsoft Teams

## Prise de contact avec l’enseignant

* Prioritairement via l’espace de travail collaboratif prévu au sein de Microsoft Teams
* Via email : [frederic.jamoulle@he-ferrer.eu](mailto:frederic.jamoulle@he-ferrer.eu)

# Ressources générales

* Database Management Essentials : University of Colorado System – Michael Mannino (<https://www.coursera.org/learn/database-management/home/info>) [1]
* Modéliser les données avec MySQL Workbench : <https://fsmrel.developpez.com/basesrelationnelles/workbench/>
* MySQL 5 (versions 5.1 à 5.6) - Guide de référence du développeur (Nouvelle édition) – Didier DELÉGLISE – Edition : ENI – Avril 2013 [2]
* MySQL Sakila Sample Database  : <https://dev.mysql.com/doc/sakila/en/>
* Cours de bases de données – Modèles et langages : <http://sql.bdpedia.fr/>
* Cours de bases de données d’ Amélie Gheerbrant, professeur à l’université : <https://www.irif.fr/~amelie/BD.html>
* Cours et tutoriels sur le langage SQL : <https://sql.sh/>
* Structure de BD et tutoriel de conception de Lucidchart : <https://www.lucidchart.com/pages/fr/structure-de-basee-de-donnees-et-tutoriel-de-conception>
* Administrez vos bases de données avec MySQL  : <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql?status=published>
* Tutoriel sur le cycle de vie d’un développement logiciel : <https://www.tutorialspoint.com/sdlc/index.htm>
* Tutoriel sur la maîtrise des bases de données NoSQL : <https://openclassrooms.com/en/courses/4462426-maitrisez-les-bases-de-donnees-nosql?status=published>

# Outils

## Introduction

Ce chapitre fournit la liste de tous les outils qui seront mis en œuvre au sein de ce cours.

Tous les outils choisis sont « open source » et donc gratuits.

## XAMPP

Installation de XAMPP : <https://sourceforge.net/projects/xampp/>

XAMPP contient Apache web server, MySQL, PHP, Perl, un serveur FTP server et phpMyAdmin.

## MySQL Workbench

Installation de MySQL Workbench 8.0 : <https://dev.mysql.com/downloads/workbench/>

La version 8.0.13 a été utilisée pour créer ce syllabus.

# Définition du contexte d’utilisation des bases de données et des technologies associées

## Définitions générales

###### Base de données

Une BD (électronique) est une collection de données, ou d’informations, qui est structurée afin de permettre des opérations rapides de recherche et de modification par des ordinateurs (ou autres composants électroniques).

Les caractéristiques essentielles d’une BD sont :

* la persistance des données ;
* les liens entre les données (données connectées) ;
* le partage des données.

###### Système de gestion de base de données

Un SGBD est un logiciel qui interagit avec les utilisateurs, les applications (clientes de la DB) et la DB afin de capturer et analyser les données.

Un SGBD devra notamment assurer :

* l’accès aux données ;
* la sécurité des données par rapport aux utilisateurs ;
* la disponibilité des données en tenant compte des pannes ;
* la cohérence et l’intégrité des données.

###### Système d’information

C’est un ensemble intégré de composants logiciels et matériels pour collecter, enregistrer et traiter des données dans le but de fournir des informations, de la connaissance et des produits digitaux.

## Contexte d’utilisation des bases de données

Les bases de données sont utilisées dans la majorité des applications nécessitant un partage de données.

Depuis l’avènement de l’ère numérique, le nombre de secteurs mettant en œuvre des bases de données est croissant : eCommerce, site de streaming, site de jeux…

Les bases de données sont généralement mises en œuvre :

* soit localement, soit dans un réseau interne (à une société par exemple), soit sur internet.
* soit interfacées directement par une application web (via un browser), soit par une application de bureau (EXE), soit par une interface de programmation (API ou service web).

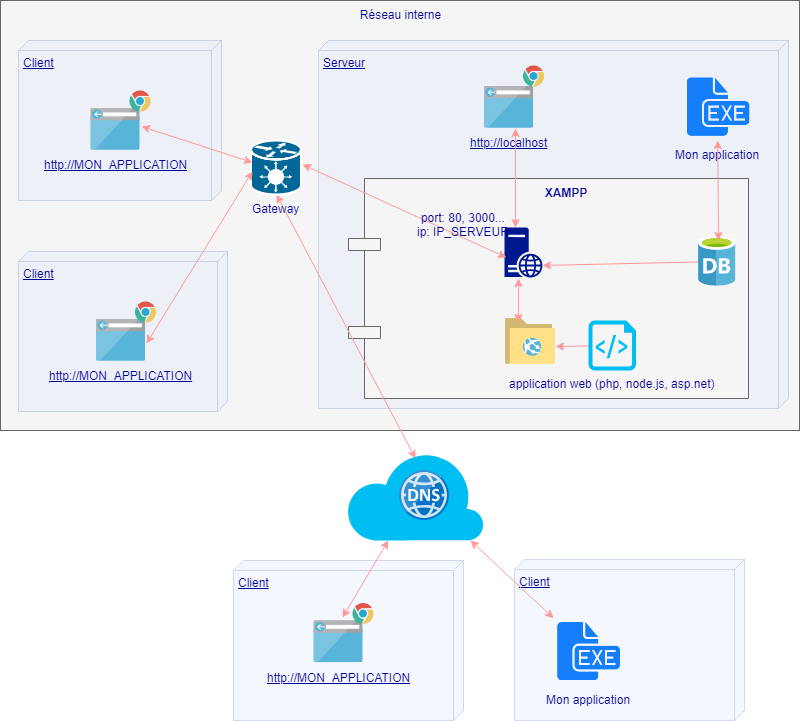


Figure 1 : exemple d’applications web et de bureau utilisant une BD directement

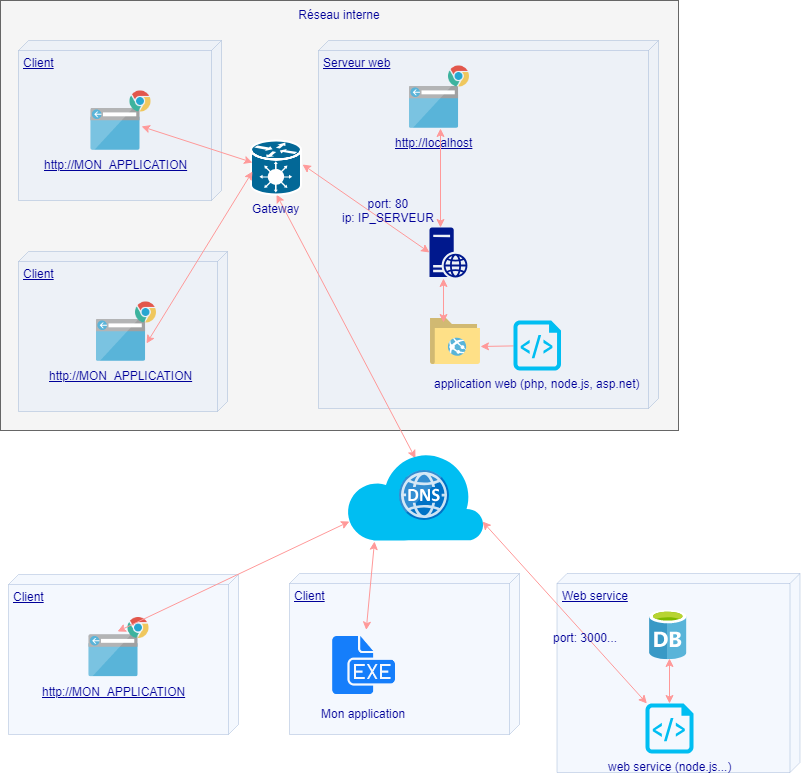


Figure 2 : exemple d’applications web et de bureau utilisant une BD par l’intermédiaire d’un web service

## Les différents systèmes de gestion de bases de données actuels

### Popularité des systèmes de bases de données

En 2019, voici comment DB-Engines Ranking [3], un portail créé en 2012 et maintenu par la compagnie autrichienne Solid IT, a listé les systèmes de BD par popularité (selon des critères définis sur le site), en couvrant environ 340 systèmes :

Table 1 : classement des SGBD par DB-Engines en 2019

| **Rank** | | | **DBMS** | **Database Model** | **Score** | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Jan 2019** | **Dec 2018** | **Jan 2018** | **Jan 2019** | **Dec 2018** | **Jan 2018** |
| 1. | 1. | 1. | [Oracle detailed information](https://db-engines.com/en/system/Oracle) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1268.84 | -14.39 | -73.11 |
| 2. | 2. | 2. | [MySQL detailed information](https://db-engines.com/en/system/MySQL) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1154.27 | -6.98 | -145.44 |
| 3. | 3. | 3. | [Microsoft SQL Server detailed information](https://db-engines.com/en/system/Microsoft+SQL+Server) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1040.26 | -0.08 | -107.81 |
| 4. | 4. | 4. | [PostgreSQL detailed information](https://db-engines.com/en/system/PostgreSQL) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 466.11 | +5.48 | +79.93 |
| 5. | 5. | 5. | [MongoDB detailed information](https://db-engines.com/en/system/MongoDB) | [Document store](https://db-engines.com/en/article/Document+Stores) | 387.18 | +8.57 | +56.24 |
| 6. | 6. | 6. | [IBM Db2 detailed information](https://db-engines.com/en/system/IBM+Db2) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 179.85 | -0.90 | -10.43 |
| 7. | 7. | up arrow 9. | [Redis detailed information](https://db-engines.com/en/system/Redis) | [Key-value store](https://db-engines.com/en/article/Key-value+Stores) | 149.01 | +2.19 | +25.88 |
| 8. | 8. | up arrow 10. | [Elasticsearch detailed information](https://db-engines.com/en/system/Elasticsearch) | [Search engine](https://db-engines.com/en/article/Search+Engines) | 143.44 | -1.26 | +20.89 |
| 9. | 9. | down arrow 7. | [Microsoft Access](https://db-engines.com/en/system/Microsoft+Access) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 141.62 | +2.10 | +14.92 |
| 10. | 10. | up arrow 11. | [SQLite detailed information](https://db-engines.com/en/system/SQLite) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 126.80 | +3.78 | +12.54 |
| 11. | 11. | down arrow 8. | [Cassandra detailed information](https://db-engines.com/en/system/Cassandra) | [Wide column store](https://db-engines.com/en/article/Wide+Column+Stores) | 122.98 | +1.17 | -0.89 |
| 12. | 12. | up arrow 15. | [Splunk](https://db-engines.com/en/system/Splunk) | [Search engine](https://db-engines.com/en/article/Search+Engines) | 81.43 | -0.76 | +17.42 |
| 13. | up arrow 14. | up arrow 17. | [MariaDB detailed information](https://db-engines.com/en/system/MariaDB) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 78.82 | +1.56 | +20.52 |
| 14. | down arrow 13. | down arrow 12. | [Teradata detailed information](https://db-engines.com/en/system/Teradata) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 76.19 | -2.98 | +3.56 |
| 15. | 15. | up arrow 18. | [Hive detailed information](https://db-engines.com/en/system/Hive) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 69.91 | +2.52 | +14.42 |
| 16. | 16. | down arrow 14. | [Solr](https://db-engines.com/en/system/Solr) | [Search engine](https://db-engines.com/en/article/Search+Engines) | 61.48 | +0.13 | -2.89 |
| 17. | 17. | down arrow 16. | [HBase detailed information](https://db-engines.com/en/system/HBase) | [Wide column store](https://db-engines.com/en/article/Wide+Column+Stores) | 60.39 | +0.38 | -1.24 |
| 18. | 18. | up arrow 19. | [FileMaker](https://db-engines.com/en/system/FileMaker) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 57.15 | +0.50 | +1.95 |
| 19. | 19. | up arrow 20. | [SAP HANA detailed information](https://db-engines.com/en/system/SAP+HANA) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 56.64 | +0.33 | +10.47 |
| 20. | up arrow 21. | up arrow 22. | [Amazon DynamoDB detailed information](https://db-engines.com/en/system/Amazon+DynamoDB) | Multi-model https://db-engines.com/info.png | 55.09 | +0.80 | +17.17 |

Suite à un sondage réalisé en 2018 sur StackOverflow [4], une communauté online de développeurs, les bases de données les plus utilisées sont :

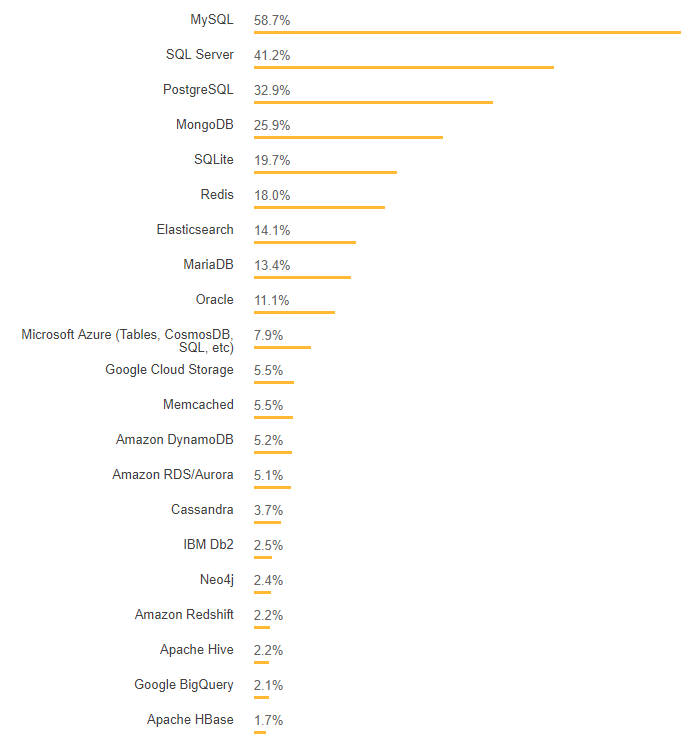


Figure 3 : résultats d’un sondage sur l’utilisation des différents SGBD par les développeurs en 2018

### SQL vs noSQL

Tous les systèmes de gestion de bases de données actuels peuvent être catégorisés selon deux grandes familles : SQL ou NoSQL.

#### SQL / Système de Gestion de Bases de Données Relationnelle

Les Systèmes de Base de Données Relationnelles (SGBDR) ou RDBMS en EN (Relational Database Management System) utilisent SQL (Structured Query Language) pour interroger et maintenir les DB.

Le SGBDR est basé sur le modèle relationnel introduit dans les années 70 (par Edgar F. Codd), où l’information est organisée dans des tableaux à deux dimensions (lignes et colonnes).

Historiquement, les BD relationnelles ont été introduites pour répondre aux faiblesses des BD hiérarchiques et réseaux.

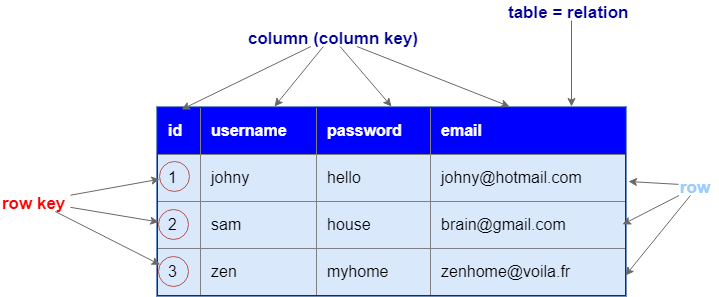


Figure 4 : exemple de BD relationnelle

La manipulation de ces données se fait selon la théorie mathématique des relations.

En 2019, c’est encore les BD relationnelles qui sont le plus utilisées dans le monde.

Ces systèmes sont matures, performants, mais connaissent certaines limitations, notamment une structure et des types de données rigides.

#### NoSQL

La famille des BD « Not Only SQL » (NoSQL), qui a émergé vers 2010, essaie de répondre aux faiblesses des RBMS, notamment afin de permettre de traiter de problèmes de performances et d’extensibilité (traitement de volumes de données très importants).

Elle recouvre principalement 4 types de DBMS :

* BD NoSQL orientée clé-valeur / « Key-value store or DB »
* BD NoSQL orientée graphe / « Graph DB »
* BD NoSQL orientée document / « Document DB »
* BD NoSQL orientée colonne (ou agrégat) / « Wide-column store or DB »

##### BD NoSQL orientée clé-valeur

Parmi les BD NoSQL, la représentation en clé-valeur est la plus simple.

Chaque clé est associée à une valeur.

Cette valeur est considérée comme une collection de données opaque par le DBMS, pouvant avoir différents champs pour chaque enregistrement et n’importe quel type de donnée.

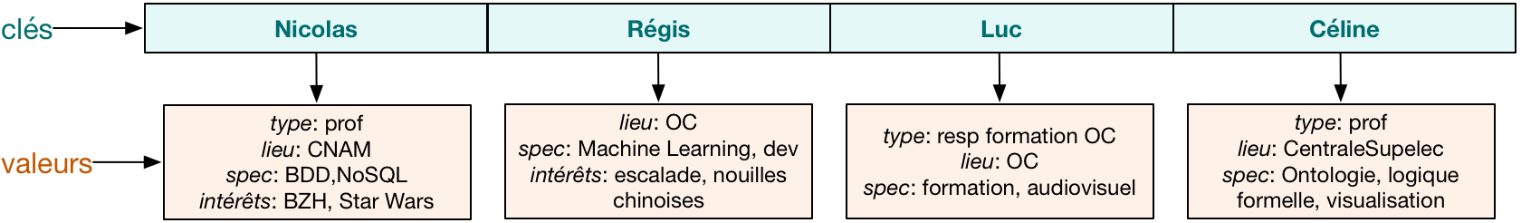


Figure 5 : exemple de BD NoSQL orientée clé-valeur [5]

Ce type de BD est :

* Particulièrement adapté pour la recherche rapide d’une collection de données par un mot clé (la clé), lorsque la valeur est directement manipulée par l’application ;
* Non adapté pour des données connectées et pour des requêtes multicritères (le SGBD ne connait pas la structure des données) ;
* Utilisé pour gérer l’accès rapides aux informations (gestion de cache par exemple), l’enregistrement de profils utilisateurs, les données de shopping…

Les plus populaires en 2019 :

* Redis (« key-value store DBMS », rang au sein des DBMS : 9)
* Amazon DynamoDB (« multi-model DBMS»)
* Datastax Enterprise (« multi-model DBMS»)

##### BD NoSQL orientée document

La représentation de ce type de BD se rapproche de la représentation clé-valeur, mais la valeur est représentée sous forme d’un document (JSON ou XML) reprenant une collection de données structurées, dont la structure est libre et souple, connue du SGBD.

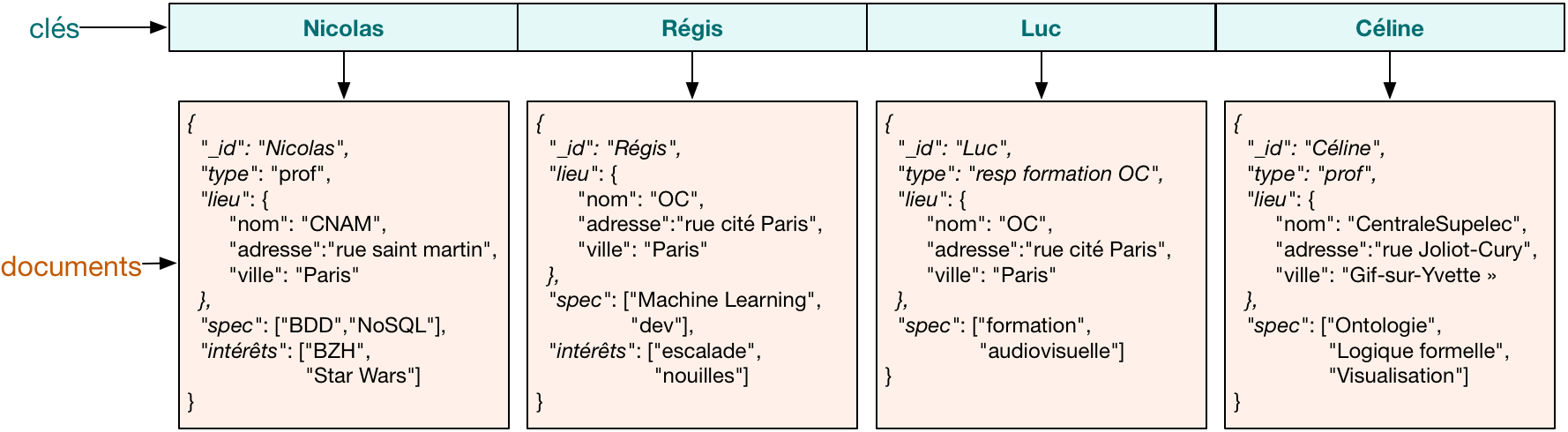


Figure 6 :exemple de BD NoSQL orientée document [5]

Ce type de BD est :

* En terme de fonctionnalité, ce qui se rapproche le plus d’un SGBD relationnel : langage de manipulation de données riches ;
* Particulièrement adaptée pour des systèmes devant être évolutifs (structure souple) dans un contexte distribué (difficulté des SGBD relationnels pour joindre des tables distribuées);
* Potentiellement moins adapté pour des requêtes très complexes ;
* Utilisé pour gérer des contenus (Content Management System), e-commerce, des plateformes d’enregistrement d’objets…

Les plus populaires en 2019 :

* MongoDB (« document store DBMS », rang au sein des DBMS : 5)
* Amazon DynamoDB (« multi-model DBMS»)
* Couchbase (« document store DBMS »)

##### BD NoSQL orientée graphe

C’est un type de BD orientée objet utilisant la théorie des graphes, incluant des nœuds, des liens, et des propriétés sur les nœuds et les liens.

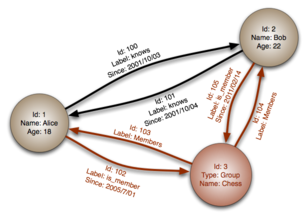


Figure 7 : exemple de BD NoSQL orientée graphe [6]

Ce type de BD est :

* particulièrement utile pour exploiter les relations entre les données, dans le cas de données fortement connectées (performances accrues).
* pas adapté pour des données à faible niveau de connexion, quand les associations n’ont qu’un ou deux niveaux de profondeur ;
* utilisé dans la modélisation des réseaux sociaux (LinkedIn par exemple), données géo spatiales (réseaux ferrés…), catalogues de produits…

Les plus populaires en 2019:

* Neo4j (« Graph DBMS », rang au sein des DBMS : 22)
* Microsoft Azure Cosmos DB (« multi-model DBMS»)
* Datastax Enterprise (« multi-model DBMS»)

##### BD NoSQL orientée colonne

C’est un type de BD qui utilise les colonnes pour enregistrer les données. La représentation de ce type de BD peut être vu comme une BD orientée clé-valeur à deux dimensions.

Il permet de ne pas avoir à traiter d’info inutiles en focalisant les requêtes sur une ou plusieurs colonnes.

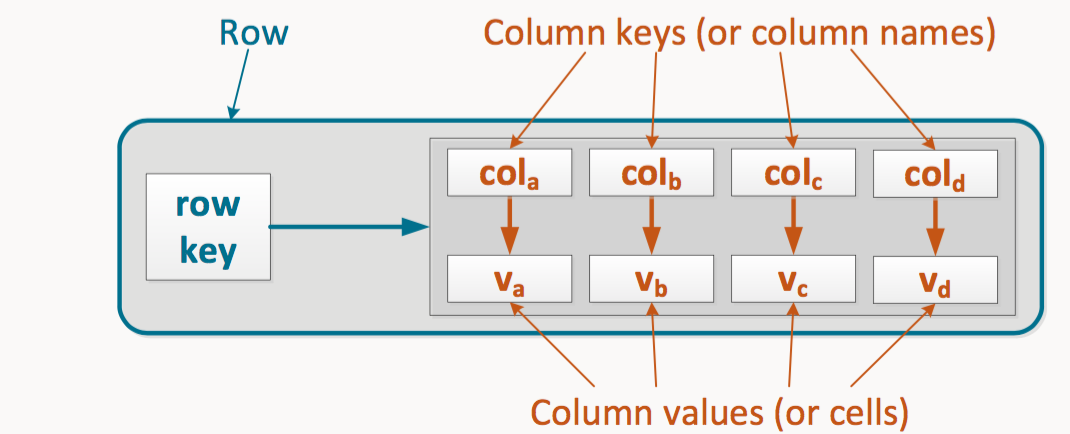


Figure 8 : représentation générique d’une BD NoSQL orientée colonne [7]

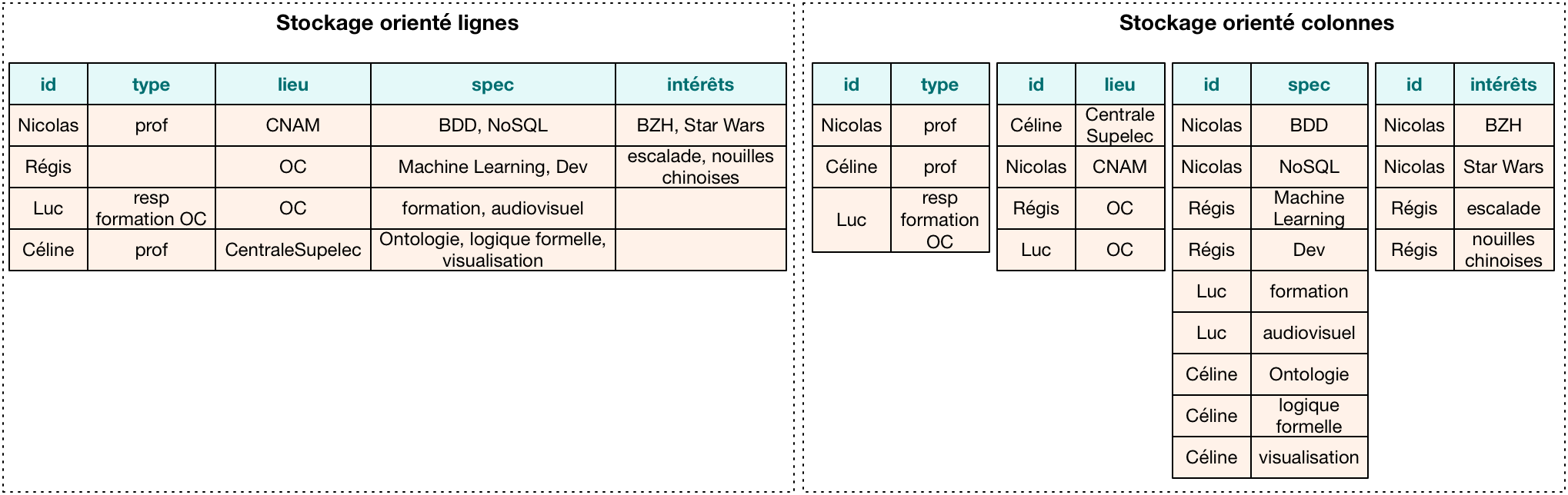


Figure 9 : exemple de BD NoSQL orientée colonne [5]

Ce type de BD est :

* particulièrement adapté pour faire des calculs sur des attributs (ou colonnes) spécifiques, pour faire des calculs analytiques (moyennes, comptage…) ;
* pas adapté pour la simple lecture de données, dans le cadre de recherche par mot clé par exemple, ou dans le cadre de requêtes complexes ;
* utilisée dans les application de comptage, des applications d’écritures lourdes de données (journalisation), reporting à large échelle…

Les plus populaires en 2019:

* Cassandra (« wide column store DBMS », rang au sein des DBMS : 11) ;
* HBase (« wide column store DBMS  »);
* Microsoft Azure Cosmos DB (« multi-model DBMS»).

#### Cas d’utilisation

Comme les SGBD relationnels et chacun des SGBD NoSQL ont leurs points forts, il est donc envisageable que pour supporter des applications de grande envergure, plusieurs types de SGBD soient mis en œuvre.

Il est intéressant de voir l’application de différents SGBD à une application reprenant des millions d’utilisateurs.

En 2017, Twitter, dans son blog [8], a fourni une très belle description des technologies mises en œuvre afin de supporter l’envoi de centaines de millions de Tweets par jour.

Pour gérer le stockage, voici le lien entre les SGBD vu dans ce cours et les technologies mises en œuvre par Twitter :

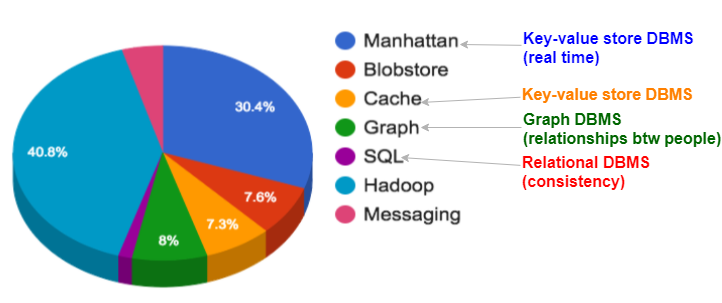


Figure 10 : Lien entre les technologies mises en œuvre pour le stockage des données de Twitter [8] et les SGBD

# Mise en œuvre de BD existantes : initiation pratique aux bases de données relationnelles et à leur application sur un réseau

## Introduction

Comme présenté précédemment, il existe de nombreux systèmes de gestion de BD.

Au sein de ce chapitre, nous allons nous focaliser sur le type de SGBD le plus utilisé actuellement, les SGBD relationnels.

Nous allons apprendre les notions de bases des SGBD relationnels et développer nos compétences dans la mise en œuvre d’application multimédia existantes.

## Introduction à la notion de réseaux, d’applications clients / serveur et de sécurisation

### Déploiement d’une base de données existante sur un serveur de BD

Nous allons déployer une base de données en utilisant des requêtes existantes, sans serveur web.

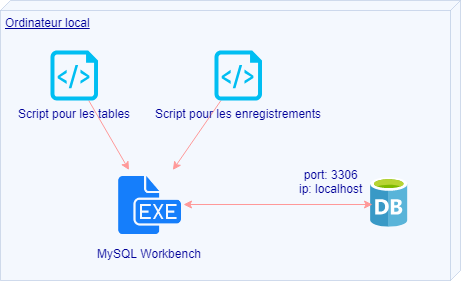


Figure 11 : MySQL Workbench pour création de BD selon des requêtes existantes

**BD-01-SHOW** : Déployer la BD existante «***university*** » au sein d’un serveur de BD de type MySQL, sur base de requêtes données dans le cours Database Management Essentials [1].

**Notions** :

* Installation et mise en route d’un serveur de BD de type MySQL (MariaDB)
  + Installation de XAMPP :

Table 2 : définition de l’acronyme XAMPP

| **Lettre** | **Signification** |
| --- | --- |
| X | Se réfère à un système « cross-platform » |
| A | Apache (serveur web) |
| M | MariaDB (anciennement MySQL) |
| P | PHP (language de programmation) |
| P | Perl (language de programmation) |

* Utilisation de MySQL workbench :
  + pour créer le schéma d’une BD ;
  + pour créer les tables d’une BD à partir d’un script MySQL ;
  + pour ajouter des données à un schéma d’une BD ;
  + pour visualiser les données ;

#### Création du schéma de la BD

**Notions** : Voici la procédure pour créer le schéma « ***university***» de la BD via MySQL Workbench :

1. Vérifier que le serveur MariaDB / MySQL est bien installé et lancé.
2. Créer une connexion via Workbench à votre propre serveur MariaDB, ou utiliser une connexion existante :

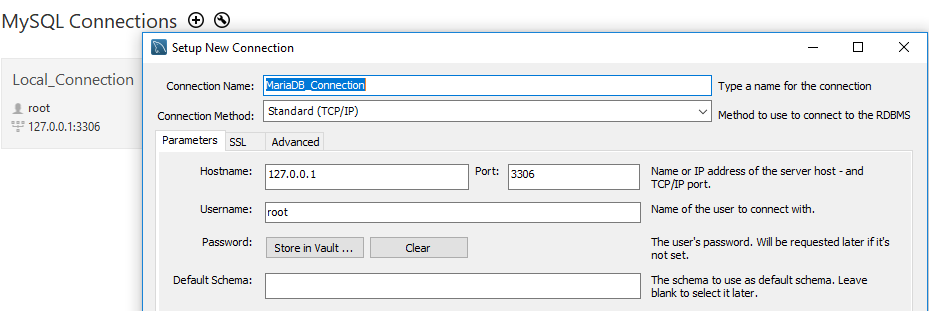


Figure 12 : création d’une connexion sous MySQL Workbench

1. Créer le schéma « ***university***» de la BD :
   1. Clic droit dans le ***« Navigator », « Create Schema… »***
   2. Choisir le bon Charset/Collation : ***utf8, utf8\_general\_ci***

#### Création des tables

**Notions** :

1. Pour créer les tables via MySQL Workbench :
   1. Cliquer sur ***File, « Run SQL Script »***
   2. Choisir : ***BD01A.sql***
   3. Cliquer sur ***Open***
   4. ***Default Schema Name: “university”***

**BD-01A-SHOW** : Montrer la structure des tables au sein de MySQL Workbench, visualiser qu’il n’y a pas d’enregistrement.

#### Ajout des enregistrements

**Notions** :

1. Pour créer les enregistrements via MySQL Workbench :
   1. Cliquer sur ***File, « Run SQL Script »***
   2. Choisir : ***BD-01B.sql***
   3. Cliquer sur ***Open***
   4. ***Default Schema Name: “university”***

**BD-01B-SHOW** : Visualiser les enregistrements.

**BD-02-EX** : Déployer la BD existante « ***university*** » sur votre serveur MariaDB à l’aide de MySQL Workbench.

### Gérer et sécuriser une base de données à partir d’une interface web existante

**BD-03-SHOW**: Explorer la BD “***university***” via une interface web existante. Découvrir le serveur Apache, l’application web phpMyAdmin s’interfaçant avec le serveur de BD MariaDB.

**Notions**:

* Installation et gestion d’un serveur web (Apache par exemple) permettant, à l’aide de scripts PHP et de données se trouvant dans une BD Maria, de créer des pages web dynamiques et de les envoyer à un client :
  + Utilisation de l’outil XAMPP Control Panel.
  + Utilisation de phpMyAdmin.

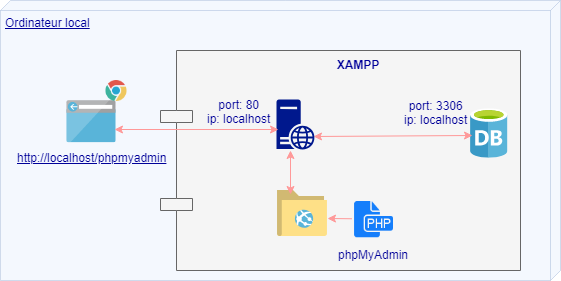


Figure 13 : représentation de l’utilisation local de phpMyAdmin

**BD-03A-SHOW** : 1. Découvrir le serveur http offert par XAMPP : [http://localhost](http://localhost/)

2. Faire en sorte qu’il écoute sur le port 3000 à la place du port 80, et juste localement.  
3. Montrer l’accès au fichier ***welcome.txt*** quand aucun nom de fichier est donné.

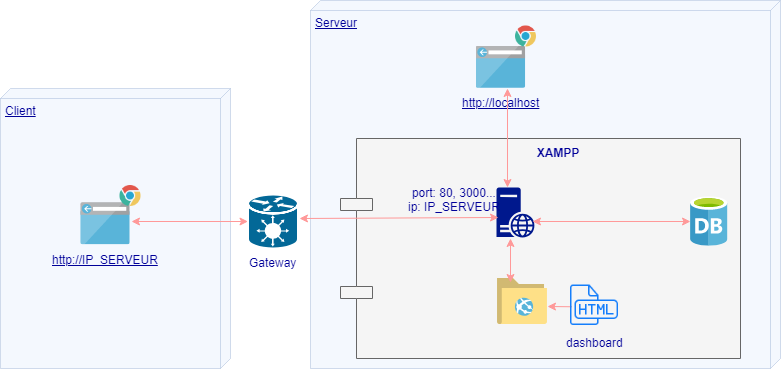


Figure 14 : représentation de l’utilisation distante du « dashboard » publié par le serveur Apache de XAMPP

**Notions** :

* Serveur http
  + Configurer le serveur web via le fichier de configuration : ***XAMPP Control Panel, Config, Apache (httpd.conf)***
    - Ecouter sur une adresse IP locale seulement, sur un port particulier : ***Listen localhost:3000*** (ne pas oublier de redémarrer Apache)
    - Répertoire mettant à disposition les documents : ***DocumentRoot "C:/xampp/htdocs"***
    - Nom du fichier cherché par défaut : ***DirectoryIndex file\_name***
  + Accéder aux fichiers publiés :
    - ***XAMPP Control Panel, Explorer, htdocs***
    - NB : pour phpMyAdmin, le fichier de configuration ***httpd.conf*** spécifie (via un alias), l’accès à un autre répertoire quand l’URL ***/phpmyadmin*** est utilisée
* Adresse IP / Localhost
  + Généralités
  + Découvrir son adresse IP : commande ***ipconfig***
  + Voir si l’accès au serveur est possible : commande ***ping***
* Ports
* Nom de domaine
  + Résolution de nom via DNS
  + Résolution de nom via le fichier hosts de Windows (seulement si droits d’administration) : ***C:\WINDOWS\system32\drivers\etc\hosts***
* Accéder au contenu d’une requête http, de la réponse http : ***Chrome, More tools, Developper tools, Network, Name (localhost par exemple), Headers***

**BD-03B-EX**: Configurer le serveur web afin qu’il écoute sur le port 4000 les requêtes, et, quand l’adresse IP du serveur est donnée, renvoyer le document ***welcome.txt*** (prévoir un message de type « bienvenu sur le serveur de PRENOM NOM. ») mis dans le répertoire publié par votre serveur web.   
Accéder au serveur de votre voisin afin de découvrir son message d’accueil.

**BD-03C-SHOW**: Découvrir l’application phpMyAdmin permettant de gérer des BD se trouvant sur un serveur MySQL / MariaDB.   
Mettre en place une authentification sécurisée à phpMyAdmin via formulaire web de connexion (username et password), pour tout utilisateur existant dans la DB système de MariaDB.

**Notions** :

* Application web sous forme de documents, notamment des scripts PHP (par exemple, voir ***index.php*** au sein de ***C:\xampp\phpMyAdmin***), générant de l’html qui est renvoyé au client
* Configuration de l’authentification pour l’administrateur principal de phpMyAdmin : ***User accounts, root@localhost, Edit privileges, Change password***
* Configuration de phpMyAdmin via le fichier de configuration : ***XAMPP Control Panel, Config, phpMyAdmin (config.inc.php)***
  + Changement du type d’authentification (demande d’un username et password) :

D’une authentification d’un utilisateur existant dans la DB de MariaDB (root), sans password : ***$cfg['Servers'][$i]['auth\_type'] = 'config';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';  
$cfg['Servers'][$i]['AllowNoPassword'] = true;***

Vers l’authentification de n’importe quel utilisateur existant dans la DB de MariaDB, et sécurisation du login avec password:   
***$cfg['Servers'][$i]['auth\_type'] = 'cookie';  
$cfg['Servers'][$i]['AllowNoPassword'] = false;***

**BD-03D-SHOW**: Via phpMyAdmin, création d’un nouvel administrateur pour la base de donnée « ***university*** » (et aucun droit sur les autres BDs) et connexion avec cette utilisateur sur phpMyAdmin.

**Notions** :

* Ajouter des privilèges à une BD : ***cliquer sur le nom de la BD, Privileges, Add user account,*** choisir ***localhost***
* Déconnecter l’utilisateur en cours :

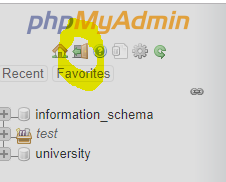


Figure 15 : interface de déconnexion d’un utilisateur au sein de phpMyAdmin

**BD-03E-SHOW**: Visualisation des tables « systèmes » de mysql, notamment la table des « utilisateurs » où se trouve notre nouvel utilisateur. Ajout d’un professeur à la table « ***faculty*** » (ou « ***teacher*** »). Vérifier ce qui se passe si l’on tente d’ajouter deux fois la même personne. Suite à sa promotion, modification du salaire du professeur ***JULIA MILLS*** et suppression de son superviseur.

**Notions** :

* Visualiser les tables systèmes à partir de phpMyAdmin : ***se connecter en root, cliquer sur la table mysql, user, db…***
* Visualiser les enregistrements d’une table à partir de phpMyAdmin : ***cliquer sur la table, (Browse qui est mis par défaut)***
* Ajout d’un enregistrement au sein d’une table à partir de phpMyAdmin : ***cliquer sur la table, Insert***

**BD-03F-EX**: Créer le nouvel administrateur de votre base de donnée « ***university*** » (avec aucun droit sur les autres BDs) portant votre prénom et se connecter avec cette utilisateur sur phpMyAdmin. Ajouter un étudiant à la table « ***student*** » qui vous représente. Modifier la moyenne (***GPA***) de l’étudiant ***TESS DODGE*** pour qui l’on a découvert une tricherie à un examen.

## Introduction aux définitions et manipulations de données

### Le modèle relationnel

Le modèle relationnel est basé sur une organisation des données sous forme de tables à 2 dimensions constituées de lignes et de colonnes. La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle.

Les opérations relationnelles permettent de créer une nouvelle relation (table) à partir d'opérations élémentaires sur d'autres tables (par exemple l'union, l'intersection, ou encore la différence).

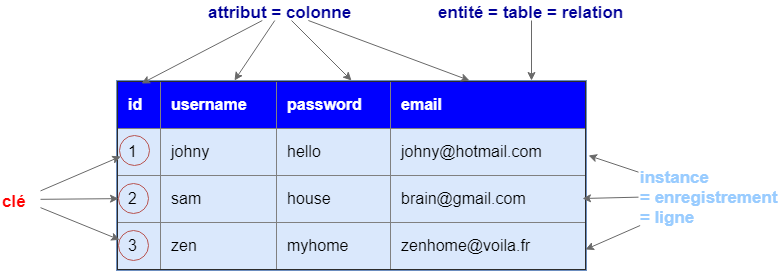


Figure 16 : le modèle relationnel

Chaque table représente une entité, chaque colonne (ou champ) représente un attribut de cette entité et chaque ligne (ou enregistrement) représente une occurrence de cette entité.

### Les tables

#### Définitions

Dans un SGBD relationnel, une relation est un sous-ensemble de données caractérisé par des attributs visualisables sous forme de table.

Une table est l’implémentation physique d’une relation entre différentes colonnes, et une ligne représente une correspondance entre ces colonnes.

Une table est composée :

* de colonnes, les attributs. Chaque colonne est formée au minimum :
  + d’un nom
  + d’un type de données (***CHAR***, ***VARCHAR***, ***INT***…)
  + d’une longueur ou valeur (en cas d’énumération)
* de lignes, les enregistrements (ou tuples).

#### Visualisations d’une table

Une table peut être représentée et visualisée de diverses manières. Si l’on reprend la table « teacher » de la BD « university », voici comment celle-ci se présente au sein de différents outils :

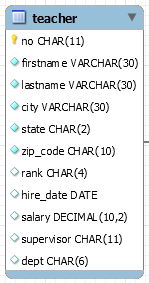


Figure 17: Représentation d’une table au sein d’un modèle EER de MySQL Workbench

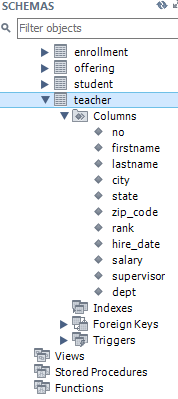


Figure 18 : Représentation d’une table au sein du schéma d’une BD (MySQL Workbench)

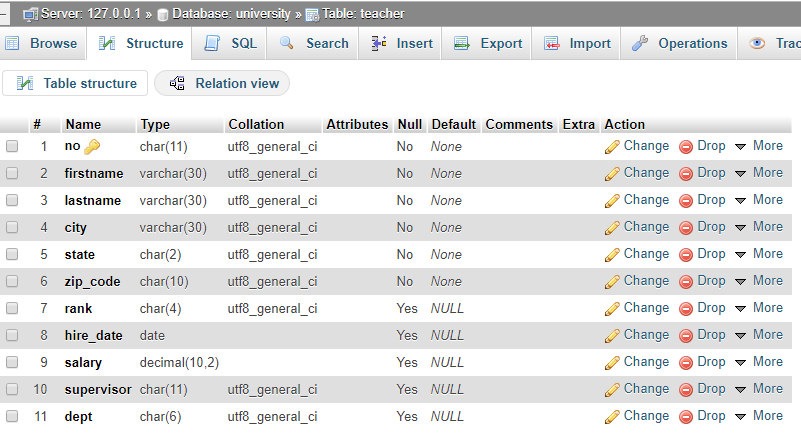


Figure 19 : Représentation de la structure d’une table dans phpMyAdmin

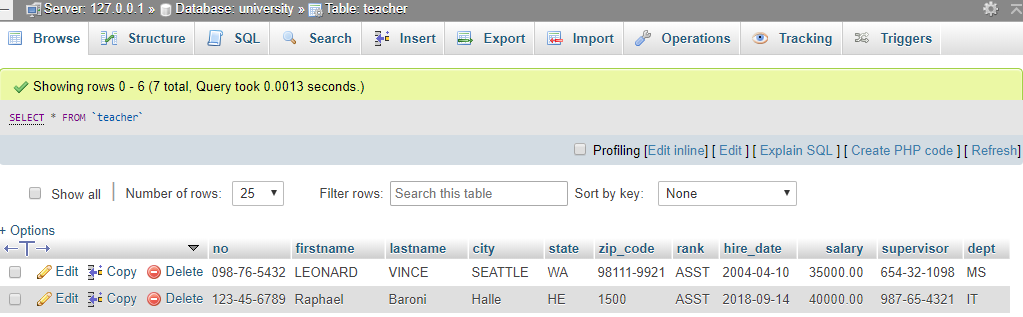


Figure 20 : Représentation des colonnes et des enregistrements d'une table dans phpMyAdmin

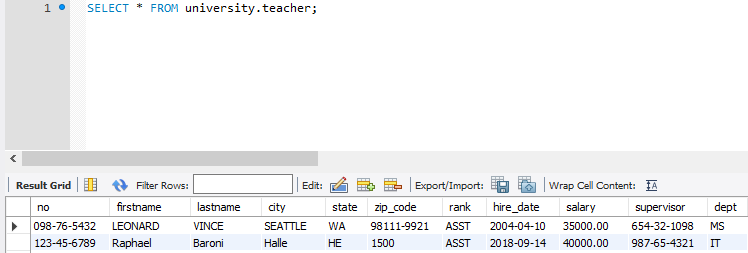


Figure 21 : Représentation des colonnes et des enregistrements d'une table dans MySQL Workbench

NB : On peut distinguer les tables « courantes » qui représente des entités (***student***, ***teacher***…) des tables « associatives » qui assurent la liaison entre au moins deux tables.

#### Conventions

MySQL n’impose pas de normes pour nommer les tables. Néanmoins, les règles de bonnes pratiques sont :

* nom de table et colonne au singulier ;
* frontière des mots en utilisant soit :
  + le style « snake\_case », par exemple : ***hello\_world***, ***public\_relation***, ***e\_mail***…
  + le style « camelCase », par exemple : ***iPhone***, ***eCommerce***, ***publicRelation***…
* nom de l’identifiant : ***id*** ou ***tableName\_id*** ou ***tableNameId***
* nom d’une table d’association : soit ***tableName1\_has\_tableName2***, soit éviter la concaténation en donnant le nom de la relation.

### Les associations

#### Définitions

Les associations permettent de définir comment les tables sont reliées entre-elles.

Une association unit 2 ou n entités.

Au sein de diagramme EER (Enhanced Entity Relationship) de MySQL Workbench, 5 types d’associations sont possibles :

| **Schéma** | **Nom au sein de MySQL Workbench** | **Définition** |
| --- | --- | --- |
|  | One-to-one non-identifying relationships | Une association non-identifiante reliant une occurrence (ou une ligne d’une table) de l’entité A à maximum 1 occurrence de l’entité B. |
|  | One-to-many non-identifying relationships | Une association non-identifiante reliant une occurrence de l’entité A à 1 ou plusieurs occurrence(s) (n) de l’entité B. |
|  | One-to-one identifying relationships | Une association identifiante reliant une occurrence de l’entité A à maximum 1 occurrence de l’entité B. |
|  | One-to-many identifying relationships | Une association identifiante reliant une occurrence de l’entité A à 1 ou plusieurs occurrence(s) (n) de l’entité B. |
|  | Many-to-many identifying relationships | Une association identifiante reliant :   * une occurrence de l’entité A à 1 ou plusieurs occurrence(s) (n) de l’entité B. * une occurrence de l’entité B à 1 ou plusieurs occurrence(s) (m) de l’entité B. |

| **Nom** | **Définition** |
| --- | --- |
| Une association identifiante | Une occurrence d’une entité (ou un enregistrement d’une table) dépend d’une occurrence d’une entité parente.  Une occurrence de la table enfant ne peut pas être identifiée de manière unique sans son parent.  La clé primaire de l’enfant devient donc généralement une clé composite des deux clés primaires des deux tables. |
| Une association non-identifiante | Une occurrence d’une entité (ou un enregistrement d’une table) ne dépend pas d’une occurrence de l’entité associée. |

#### Visualisation des associations

Dans la figure qui suit, la clé étrangère ***category\_id*** fait partie de la clé primaire de la table d’association « ***film\_category*** » :

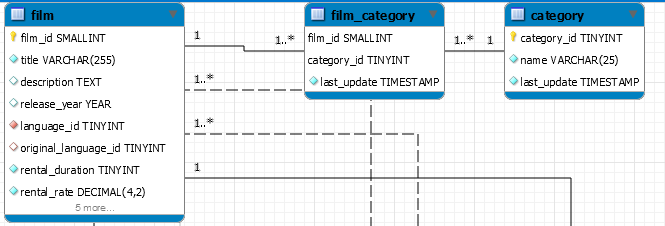


Figure 22 : Exemple d’association identifiante de la BD sakila (film\_category)

Dans cette figure, la clé étrangère ***city\_id*** (provenant de la clé primaire de la table ***country***) ne fait pas partie de la clé primaire de la table ***city***.

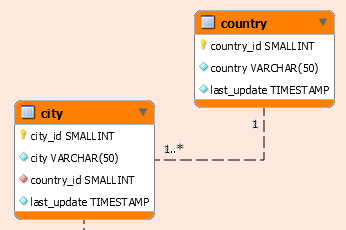


Figure 23 : Exemple d’association non-identifiante de la BD sakila

### Créer le modèle de données de la BD « university »

Un modèle « Enhanced Entity-Relationship » d’une BD est généralement créé lors de la phase de conception d’une BD.

Il permet notamment :

* de documenter la BD ;
* d’identifier la structure des données nécessaires pour implémenter les besoins des utilisateurs de la BD ;
* d’automatiser la maintenance d’un schéma.

Une fois une BD installée, il est intéressant de générer le modèle de celle-ci, soit à partir du script de création des tables, soit à partir de la DB.

**BD-04-SHOW** : Créer le « Enhanced Entity-Relationship » modèle « ***universityModel***» à partir de la BD « ***university***» afin de faciliter une mise à jour ultérieure du schéma.

**Notions** :

* Mise en route du serveur de BD MariaDB ;
* Utilisation de MySQL workbench pour créer un modèle de type « Enhanced Entity-Relationship » à partir d’une BD :
  + Soit via le menu des ***Models***

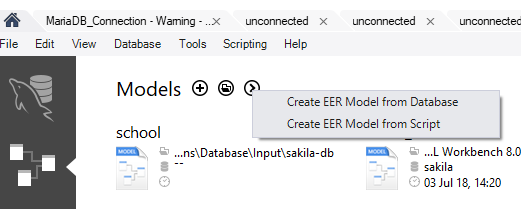


Figure 24 : menu de création d’un modèle EER au sein de MySQL Workbench

* + Soit à partir d’un schéma de DB existant : ***Database, Reverse Engineer…***

### Synchroniser le modèle de données de la BD « university » avec le schéma

**BD-05-SHOW** : Mettre à jour le modèle « ***universityModel***» en renommant la table « ***faculty*** » en table « ***teacher***», en modifiant les propriétés (nom sans « ***fac*** », pas de majuscule, ajout de « underscores »).

Synchroniser le modèle. Faire un refresh du schéma de la BD « ***university***» pour voir les changements.

Observer la clé étrangère « ***FacNo*** » au sein de la table ***offering***.

**Notions** :

* Pour synchroniser (MySQL Workbench) le modèle « ***universityModel***» avec la BD « ***university*** :
  + Ouvrir le modèle
  + ***Database, Synchronize Model…***
* Pour visualiser les clés étrangères dans une table (MySQL Workbench) : ***double clic sur une table, Foreign Keys***

**BD-06-EX** : Créer votre « Enhanced Entity-Relationship » modèle « ***universityModel***» à partir de la BD « ***university*** ». Mettre à jour le modèle en renommant la table « ***faculty*** » en table « ***teacher*** », en modifiant les propriétés (nom sans « ***fac*** », pas de majuscule, ajout de « underscores »).

Synchroniser ce modèle avec la BD.

### Les contraintes d’intégrité

Les contraintes d’intégrité sont les contrôles que l’on peut mettre au niveau des SGBD relationnels, afin d’implémenter des règles de gestion du système d’information.

Cela permet notamment d’assurer qu’il n’y ait pas de doublons au sein d’une entité qui puissent s’insérer, que certaines occurrences doivent être introduites avant d’autres…

Les contraintes d’intégrité sont soit appliquées au niveau d’une table ou au niveau d’une colonne.

Il existe trois types de contraintes d’intégrité :

* Les contraintes de domaine :
  + Contrôle du type ou de la valeur prises par les données, notamment :  
    ***CHECK*** : valeurs doivent vérifier une condition logique.
  + Contrôle de la valeur nulle (***NOT NULL***)  
    ***NOT NULL*** : la valeur est obligatoire.
  + Valeur par défaut (***DEFAULT***)  
    ***DEFAULT*** : lors d’une mise à jour, la valeur dans une colonne est donnée par défaut
* Les contraintes d’entité (ou de relation) :
  + Contrôle de l’unicité (***UNIQUE***)  
    ***UNIQUE*** : la valeur doit être unique dans la colonne
  + Contrôle de clé primaire (***NOT NULL et UNIQUE***)  
    ***PRIMARY KEY*** : la colonne sert d’identifiant unique des occurrences (ou des lignes d’une table)
  + Contrôle de clé primaire composite (***NOT NULL et UNIQUE***)  
    ***PRIMARY KEY*** : les colonnes servant d’identifiant unique des occurrences (ou des lignes d’une table)
* Les contraintes référentielles (ou de référence) :
  + Contrôle de la cohérence et dépendance entre deux tables, notamment pour empêcher la suppression accidentelle de données reliées.  
    ***FOREIGN KEY*** : la clé étrangère d’une table fait le lien (association) avec une autre table en pointant sur la clé primaire de cette dernière. La valeur d’une clé étrangère ne peut pas être différente d’une clé primaire d’une table référencée.

**BD-07-SHOW** : A partir de la BD «***university*** », visualiser le contrôle de la dépendance entre la table « ***offering*** » et la table « ***teacher*** », comprendre les cardinalités :

* Mise à jour de la clé étrangère donnée dans la table « ***offering*** » :
  + Mettre à jour le nom de la clé étrangère « ***FacNo*** » en « ***teacher\_no*** » manuellement.
  + Effacer la relation « ***teacher***» / « ***offering*** » et montrer comment recréer la relation 1:n graphiquement via une association  en faisant attention à ce que un « ***teacher***» ne doive pas obligatoirement exister quand il y a une occurrence d’***offering*** (cardinallité 0…1 du côté « ***offering*** »).
  + Laisser les actions référentielles par défaut (en cas de changement sur la table parent « ***teacher*** »)
* Donner un nom aux rôles associés aux cardinalités.
* Resynchroniser le modèle.
* Visualiser l’échec d’une contrainte de clé primaire : ajout d’une offre avec l’identifiant « 6666 ».
* Visualiser l’échec d’une contrainte de clé étrangère : ajout d’une offre pour un cours ou un prof. non existant.
* Visualiser la réussite d’une contrainte de clé étrangère : ajout dans l’offre (« ***offering*** ») d’un cours pour le nouveau prof.
* Visualiser l’échec d’une contrainte de clé étrangère : Tenter d’effacer le nouveau prof ajouté lors d’une précédente démonstration.
* Mise à jour du modèle afin qu’à l’effacement d’un professeur, la clé étrangère dans la table « ***offering***» soit mise à ***NULL***. Resynchroniser et effacer le nouveau prof.

**Notions** :

* Pour synchroniser (MySQL Workbench) le modèle « ***universityModel***» avec la BD « ***university*** :
  + Ouvrir le modèle
  + ***Database, Synchronize Model…***
* Pour visualiser les clés étrangères dans une table (MySQL Workbench) : ***double clic sur une table, Foreign Keys*** :
  + ***Foreign Key Name***: c’est aussi le nom du « ***caption*** » d’une association.  
    NB : par défaut, ce nom est « ***fk\_source\_table\_destination\_table*** ».  
    Utiliser le menu ***Model***, ***Model Options…*** pour changer ce nom par défaut.
  + ***Referenced Table*** : c’est le lien vers la ou les table(s) contenant la où les clé(s) primaire(s) .
  + ***Column*** : la clé étrangère est indiquée / indexée au sein de « ***Column*** ».
  + ***Referenced Column*** : la clé primaire est indiquée.
* Pour créer une association 1-n entre deux tables :   
  cliquer sur l’association 1:n , cliquer sur la table qui est du côté « beaucoup » (« \* ») de la relation, puis cliquer sur la table qui contient la clé à référencer.  
  NB : ici, clic sur «***offering*** », puis sur « ***teacher*** ».

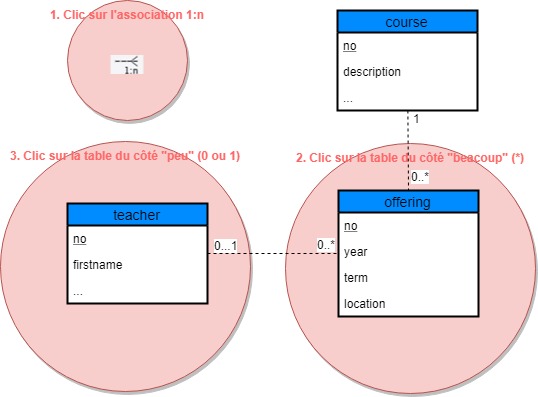


Figure 25 : création d’une association 1-n entre deux tables au sein de MySQL Workbench

* Pour modifier une association 1-n en 0-n entre deux tables afin de rendre la l’association optionnel plutôt qu’obligatoire :
  + Soit via l’association pour la cardinalité 1 et n : ***Double clic sur l’association, Foreign Key, Referencing Table, Mandatory***
  + Soit via la clé étrangère pour la cardinalité 1: ***décocher NN (NOT NULL)***
* Pour spécifier les actions référentielles en cas de mise à jour (***UPDATE*** ou ***DELETE***) d’une clé primaire d’une table parent vers la table enfant (où est donnée la clé étrangère) : ***double clic sur une table, Foreign Keys, Foreign Key Options***
  + ***CASCADE*** : ***DELETE*** ou ***UPDATE*** l’enregistrement de la table parent ainsi que tous les enregistrements associés dans la table enfant
  + ***SET NULL***: ***DELETE*** ou ***UPDATE*** l’enregistrement de la table parent et allouer NULL dans la clé étrangère de la table enfant.
  + ***RESTRICT*** : rejeter l’opération de ***DELETE*** ou ***UPDATE*** de la table parent. Même effet pour ***NO ACTION***.
  + ***NO ACTION*** : même effet que d’omettre de spécifier les clauses ***DELETE*** ou ***UPDATE***, équivalent à ***RESTRICT*** en MySQL.
* Pour donner un nom aux rôles associés aux cardinalités (noms pour l’association) : ***double clic sur l’association, Caption :, 2nd Capt.***  
  NB : généralement, le nom de la clé étrangère est donné par défaut comme nom de l’association. Le nom de l’association peut être changé, il n’apparait pas comme contrainte d’intégrité.
* Problème de synchronisation quand utilisation de MariadDB avec MySQL Workbench. Le code généré pour une index est le suivant : ***ADD INDEX `fk\_offering\_teacher\_idx` (`teacher\_no` ASC) VISIBLE,***  
  Sous MariaDB, ***VISIBLE*** doit être effacé !

**BD-08-EX** : Mettre à jour le modèle de la BD «***university*** » afin que lorsqu’un professeur est effacé de la table « ***teacher*** » (ou anciennement « ***faculty*** »), toutes les offres associées à ce professeur restent, en mettant à ***NULL*** les valeurs de la clé étrangère associée au sein de la table « ***offering*** ».

Synchroniser le modèle avec votre BD « ***university*** » et tester :

* l’ajout d’un nouveau professeur ;
* d’une nouvelle offre associée à ce professeur ;
* l’effacement du nouveau professeur.

### Les requêtes

#### Introduction

La syntaxe de requêtes SQL (ou commandes SQL) couvre ces principaux types de requêtes :

* Requêtes de définition de données (***CREATE***…)
* Requêtes de manipulation de données (***SELECT***…)
* Requêtes de contrôle de données (***GRANT***…)
* Requêtes de contrôle des transactions (***COMMIT***…)

Il existe d’autre types de requêtes au sein de MySQL, notamment pour traiter des commandes de réplication, des commandes d’administration de la BD, des commandes pour écrire des programmes

Seules les requêtes utilisées dans ce cours sont détaillées dans ce chapitre.

Ce cours s’articule autour d’outils qui permettent de définir les données et de les contrôler (contraintes d’intégrité, sécurité…), sans devoir soi-même écrire les requêtes SQL. Celles-ci sont générées par les outils. Cependant, pour pouvoir manipuler ces données, nous allons avoir besoin d’étudier les requêtes de manipulation de données plus en détails.

#### Requête de définition de données

##### Introduction

On distingue typiquement quatre types de requêtes de définition de données :

* ***CREATE*** : création d'une structure de données ;
* ***ALTER*** : modification d'une structure de données ;
* ***DROP*** : suppression d'une structure de données ;
* ***RENAME*** : renommage d'une structure de données.

Ces requêtes peuvent porter sur les structures de données de type suivantes :

* ***TABLE*** : table ;
* ***INDEX*** : indice ;
* ***VIEW*** : table virtuelle ;
* ***SEQUENCE*** : suite de nombres ;
* ***SYNONYM*** : synonyme ;
* ***USER*** : utilisateur.
* …

##### CREATE DATABASE

###### Syntaxe

Selon le manuel de référence MySQL :

***CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db\_name***

***[create\_specification]***

***create\_specification:***

***[DEFAULT] CHARACTER SET [=] charset\_name***

***| [DEFAULT] COLLATE [=] collation\_name***

En résumé :

***CREATE DATABASE db\_name;***

##### CREATE TABLE

###### Syntaxe

Résumé :

***CREATE TABLE table\_name (***

***column1 datatype,***

***column2 datatype,***

***column3 datatype,***

***....***

***);***

Manuel de référence de MySQL pour CREATE : <https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

#### Requête de manipulation de données

##### SELECT

###### Définition

Une requête ***SELECT*** fourni une vue partielle d’une ou plusieurs tables basées sur une série de critères.

Résumé :

***SELECT column1, column2, ...***

***FROM table\_name***

***[WHERE condition]***

***[ORDER BY specification]***

###### Les visualisations de base sur une table

**BD-09-SHOW** : Montrer les possibilités de MySQL Workbench lors de la manipulation des données de la BD « ***university***».

**BD-09A-SHOW** : Lister toutes les propriétés de tous les étudiants :

***SELECT \* FROM student;***

ou

***SELECT \* FROM university.student;***

**BD-09B-SHOW** : Lister seulement les nom et prénoms des étudiants :

***SELECT stdFirstName, stdLastName FROM student ;***

**BD-09C-SHOW** : Trouver tous les profs qui ont été engagés entre 2008 et 2009 :

***SELECT firstname, lastname, hire\_date***

***FROM teacher WHERE hire\_date BETWEEN '2008-01-01' AND '2009-12-31';***

**BD-09D-SHOW** :Trouver tous les profs qui ont un salaire supérieur à 50 000 $ :

***SELECT firstname, lastname, salary FROM teacher WHERE teacher.salary > 50000***

**BD-09E-SHOW** : Créer un nom complet pour les professeurs en combinant les colonnes « firstname » et « lastname » :

***SELECT concat(firstname," ", lastname) as full\_name FROM teacher***

**BD-09F-SHOW** : Créer un nom complet pour un seul professeur en combinant les colonnes « firstname » et « lastname » :

***SELECT concat(firstname," ", lastname) as full\_name FROM teacher where firstname='leonard' AND lastname='vince'***

**Notions** :

* Les expressions: ce sont des combinaisons de colonnes, constantes, opérateurs et fonctions.
* Les expressions logiques : combinaisons d’opérateurs de comparaison et d’opérateurs logiques (cela permet de déterminer les critères de recherche).

Table 3 : Opérateurs de comparaison

| **Opérateur** | **Signification** |
| --- | --- |
| = | Égal |
| > | Inférieur |
| <= | Inférieur ou égal |
| > | Supérieur |
| >= | Supérieur ou égal |
| <> ou != | Différent |

Table 4 : Opérateurs logiques

| **Opérateur** | **Symbole** | **Signification** |
| --- | --- | --- |
| AND | && | ET |
| OR | || | OU |
| XOR |  | OU exclusif |
| NOT | ! | NON |

NB: pour les opérateurs logiques, tant l’opérateur que le symbole peuvent être utilisés.

* Pour chercher un pattern :
  + Opérateur ***LIKE***
  + Meta charactères :
    - ***LIKE ‘VINC\_’*** : « ***\_***» est n’importe quel caractère seul
    - ***LIKE ‘VI%’*** : « ***%*** » est toute chaîne de caractères
* Documentation pour tout savoir sur les dates :
  + Type de données : <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-types.html>
  + Fonctions : <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>
* Création d’alias de colonne (idem pour les tables) : utilisation du mot-clé facultatif ***AS***

**BD-10-EX** : Découvrir la documentation (pour les dates) en créant la requête permettant de visualiser tous les professeurs engagés après 2005.

**BD-11-EX** : Créer la requête permettant de visualiser le salaire des professeurs américains en $, ainsi que dans une autre colonne, le salaire converti en €. Prendre le cours du jour (par exemple : 1$ = 0.88€).

Optionnellement, faire en sorte qu’il n’y ait pas de chiffre après la virgule, en arrondissant.

###### Les visualisations de plusieurs tables : les jointures

Il est très important de pouvoir joindre différentes tables afin de visualiser des données connectées.

En résumé, voici les différents types de jointures [9]:

Figure 26 : représentation des différents types de jointure de tables [9]

Il existe différents types de jointures, principalement :

* Les jointures internes : ***(INNER) JOIN***   
  Il doit y avoir des données de part et d’autre de la jointure, seulement si la condition de jointure est respectée
* Les jointures externes : fournit certaines données, même quand la condition de jointure n’est pas respectée :
  + Jointures par la gauche : ***LEFT (OUTER) JOIN***   
    Cela donne toutes les lignes de la table de gauche (pour la table de droite, seulement celles où la condition de jointure est respectée)
  + Jointure par la droite : ***RIGHT (OUTER) JOIN***  
    Idem, mais pour les lignes de la table de droite.
  + Jointure complète : ***FULL (OUTER) JOIN***  
    **Cette syntaxe n’existe pas en MySQL.**   
    Cela donne toutes les lignes de la table de gauche et de droite, même si la condition de jointure n’est pas respectée.

Il existe différents types de syntaxe, néanmoins, il est conseillé d’utiliser la syntaxe suivante :

***SELECT column\_name(s)***

***FROM table1***

***[LEFT / RIGHT] JOIN table2***

***ON table1.column\_name = table2.column\_name;***

**BD-12-SHOW** : Montrer toutes les offres de cours et de professeurs via MySQL Workbench.

**BD-12A-SHOW** : Montrer toutes les offres de cours, même si un cours n’est pas associé à une offre.

**BD-12B-SHOW** : Montrer toutes les offres de cours qui ont un professeur, avec les infos du cours.

**BD-12C-SHOW** : Montrer toutes les offres de cours avec les infos du cours et du professeur quand elles sont disponibles.

**BD-12D-SHOW** : Montrer toutes les offres de cours avec les infos du cours et du professeur quand elles sont disponibles, pour les cours se donnant en 2017.

**Notions** :

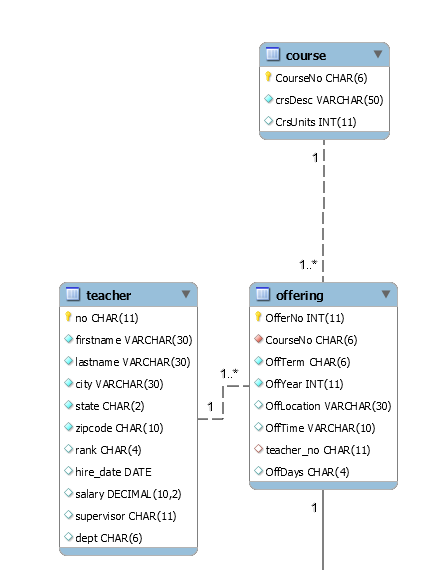
* Reprendre le modèle des données pour visualiser les attributs des requêtes :
* 

Figure 27 : partie du modèle de données de la BD « university »

**BD-13-EX** : Montrer le nom et prénom de tous les étudiants, et quand c’est possible :

* le numéro de cours, le nom du cours, l’année, la période du cours, et les jours de cours ;
* le nom et prénom du prof donnant le cours.

**Notions** :

* Afficher par ordre ascendant ou descendant : ***SELECT column1, column2, ... FROM table\_name ORDER BY column1, column2, ... ASC|DESC;***

###### Les groupes et leurs résultats

Afin de permettre de résumer l’information, par exemple dans le but de fournir des statistiques ou afin de prendre des décisions, il est important de connaître la syntaxe pour résumer des tables.

**BD-14-SHOW** : Afficher, par rang de professeur, le salaire moyen.

**BD-15-SHOW** : Afficher, par matière principal des étudiants (« ***StdMajor*** »), pour les étudiants se trouvant dans les classes de 3ème et 4ème année (« ***StdClass*** » : 1st year = freshman, 2nd year = sophomore, 3rd year = junior, 4th year = senior), le ***GPA*** moyen, si le ***GPA*** moyen est plus grand que 3.

**Notions** :

* Syntaxe d’une requête pour résumer une table (ou grouper les données au sein d’une table) :  
  ***SELECT column\_name(s), aggregate\_function(column\_name)***

***FROM table\_name***

***[WHERE condition]***

***GROUP BY column\_name(s)***

***[HAVING condition]***

***[ORDER BY column\_name(s);]***

* Fonctions d’agrégat : ***AVG, SUM, COUNT, MAX, MIN…***
* Grouper les valeurs de colonne(s) : ***GROUP BY***
* Conditions pour le résumé : ***HAVING***
* Les opérations sur les enregistrements se font avant les opérations sur les groupes : éventuel filtre ***WHERE*** appliqué au niveau des enregistrement, puis éventuel filtre ***HAVING*** au niveau du groupe

**BD-16-EX** : Afficher, par nom de cours et par année, le nombre d’offres. Afficher par ordre des cours les plus populaires (ceux où il y a le plus d’offres).

##### INSERT

###### Définition

Une requête ***INSERT*** ajoute des enregistrements à une table.

Syntaxe résumée :

***INSERT INTO table\_name (column1, column2, column3, ...)***

***VALUES (value1, value2, value3, ...)***

##### UPDATE

###### Définition

Une requête ***UPDATE*** modifie le contenu d’un ou plusieurs champ(s) d’une table basée sur une série de critères.

Syntaxe résumée :

***UPDATE table\_name***

***SET column1 = value1, column2 = value2, ...***

***WHERE condition***

##### DELETE

###### Définition

Une requête ***UPDATE*** supprime un ou plusieurs enregistrement(s) dans une table sur base d’une série de critères.

Syntaxe résumée :

***DELETE FROM table\_name WHERE condition***

### Insérer des données via un tableau

Dans beaucoup de cas, il est utile d’insérer des données à partir de tableaux. Un fichier Excel est une interface d’insertion de données qui peut être considérée, dans certains cas, comme conviviale et rapide.

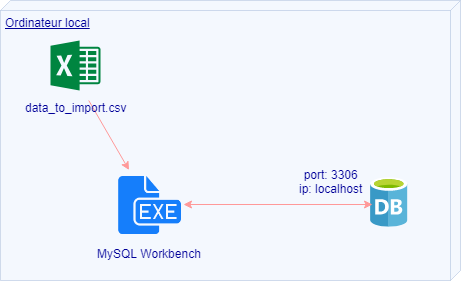


Figure 28 : représentation de l’utilisation d’Excel comme IHM pour introduire des données via MySQL Workbench

**BD-17-SHOW** : Nous souhaitons intégrer les données associées à des films (rank, title, production, director, year, budget, gross) dans une nouvelle base de données « streaming ». Nous allons utiliser les données des plus gros succès mondiaux au box-office [10].

**Notions** :

* Création d’un nouveau schéma de bases de données (MySQL Workbench)
* Création du modèle physique de données pour créer les tables :   
  Il est important de donner le nom de la DB au schéma physique (ici « streaming »): ***Double-clic sur Physical Schemas, Name***
* Génération du schéma de la DB sur base du modèle :   
  Comme c’est la première fois, la fonction ***Database, Synchronize Model…*** ne fonctionne normalement pas, sauf si le nom de la DB existante a été donné au schéma physique (au sein du modèle EER).  
  Attention aussi à utiliser une connexion permettant un accès à la DB nouvellement créée (ou à créer via la synchronisation), sinon la génération de la DB ou la synchronisation ne fonctionnera pas.
  + la 1er fois, on peut utiliser : ***Database, Forward Engineer…***  
    Attention à vérifier/corriger le script afin de générer le bon nom pour le schéma de la DB (« ***streaming*** ») ;
  + la 1ère fois, ou par la suite : ***Ouvrir le modèle EER, Database, Synchronize Model…***
* Création d’un fichier ***.csv*** contenant les données à importer dans la DB :  
  Petits trucs pour nettoyer un fichier Excel :
  + Recherche des éléments vides au sein d’un fichier Excel : ***Sélectionner la colonne où se trouve des éléments vide, Home, Find & Select, Go To Special…, Blanks***
  + Effacer les lignes des cellules sélectionnées : ***Home, Delete, Delete Sheet Rows***
  + Faire une recherche d’un caractère de type « wildcart character » (métacaractère : ***\*, ?, %...)*** dans Excel : mettre un ***~\**** dans votre recherche du caractère ***\****
* Importer les données du fichier ***.csv*** au sein de la table : ***clic droit sur la table, Table Data Import Wizard***

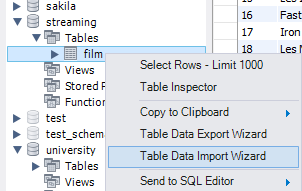
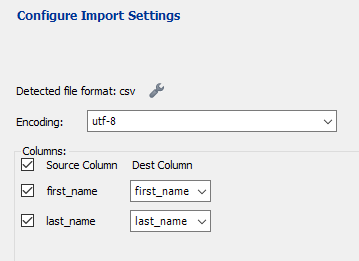


Figure 29 : interface MySQL Workbench d’importation de de données (fichier .csv) au sein d’une table

* + Pour importer les données, il faut vérifier que les colonnes trouvées dans le fichier .csv correspondent bien avec les colonnes de la table :



**BD-18-EX** : 1. Reprendre la démonstration précédente afin de créer la DB « ***streaming*** » avec les 100 films à plus gros succès financier.   
2. Dans notre future application de streaming, nous aurons besoin d’avoir une liste de tous les pays produisant des films. Créer la table «***country*** », et la remplir avec potentiellement la liste de tous les pays du monde.   
Les données sont à trouver sur internet, en anglais (par exemple : <https://en.wikipedia.org/wiki/ISO_3166-1>, <https://datahub.io/core/country-list>). Un pays doit contenir au minimum ces attributs : ***name*** (ex : Belgium), ***code*** (ex : BE).

### Les vues

###### Syntaxe de creation

***CREATE VIEW view\_name AS  
SELECT…***

###### Syntaxe d’appel d’une vue

***SELECT … FROM view\_name;***

### Les routines stockées

#### Les procédures stockées

Ce sont des programmes qui ne renvoient pas de résultats et qui exécutent une série d’instructions lors de leur appel par un call.

Syntaxe : ***CALL nom\_procedure ( [parametre(s)] )***

#### Les fonctions stockées

Ce sont des procédures qui renvoient une valeur comme résultat de leur exécution.

Syntaxe : ***SELECT nom\_fonction ( [parametre(s)] )***

#### Les triggers

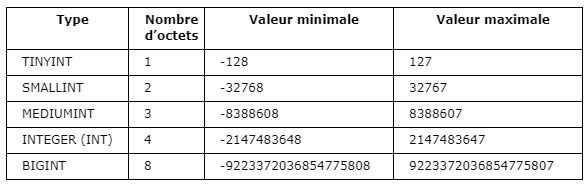
Ce sont des routines stockées exécutées automatiquement lorsque certains événements se produisent sur une table.

Un triggeur s’active avant ou après l’exécution d’une commande ***INSERT***, ***DELETE*** ou ***UPDATE*** sur la table.

### Types de données

#### Les types numériques

Pour MySQL, les types de données de bases sont [2] :

Table 5 : type de données numériques de MySQL

D’autres types intéressants sont :

* DECIMAL(M,D) : utilisé pour des valeurs exactes
  + M : c’est la précision, c’est-à-dire le nombre de chiffres enregistrés pour une valeur
  + D : c’est l’échelle, c’est-à-dire le nombre de décimaux (nombre de chiffres après le point ou la virgule)
  + exemple : DECIMAL (5,2) : les valeurs varient de -999.99 à 999.99
* FLOAT (4 bytes) ou DOUBLE (8 bytes) ou REAL (synonyme of DOUBLE) : utilisé pour des valeurs approximées

#### Les types caractères

Pour MySQL, les types de données de bases sont [2] :

* ***CHAR(M) [BINARY]***Sur M octets, M allant jusqu’à 255. Chaîne de longueur fixe de 255 caractères maximum remplie éventuellement d’espaces, si la valeur est plus courte que la colonne. L’option ***BINARY*** est utilisée pour tenir compte de la casse.
* ***VARCHAR (L) [BINARY]***Sur L+1 octets (où L représente la longueur de la chaîne). Ce type de données stocke des chaînes de **65535** caractères maximum. L’option ***BINARY*** permet de tenir compte de la casse.
* ***TINYBLOB***  
  Sur L+1 octets. Stocke des chaînes de 255 caractères maximum. Ce champ est sensible à la casse.
* ***TINYTEXT***  
  Sur L+1 octets. Stocke des chaînes de 255 caractères maximum. Ce champ est insensible à la casse.
* ***Le BLOB (Binary Large OBject)***Stocké sur L+1 octets. Stocke des chaînes de 65535 caractères maximum. Ce champ est sensible à la casse.
* ***TEXT***  
  Stocké sur L+2 octets. Stocke des chaînes de 65535 caractères maximum. Ce champ est insensible à la casse.
* ***MEDIUMBLOB***  
  Stocké sur L+3 octets. Stocke des chaînes de 16777215 caractères maximum.
* ***MEDIUMTEXT***  
  Stocké sur L+3 octets. Chaîne de 16 777 215 caractères maximum. Ce champ est insensible à la casse.
* ***LONGBLOB***  
  Stocké sur L+4 octets. Stocke des chaînes de 4 294 967 295 caractères maximum. Ce champ est sensible à la casse.
* ***LONGTEXT***  
  Stocké sur L+4 octets. Stocke des chaînes de 4 294 967 295 caractères maximum.
* ***ENUM(valeur\_possible1,valeur\_possible2,valeur\_possible3,…)***  
  Stocké sur 1 ou 2 octets (la place occupée est fonction du nombre de solutions possibles : 65 535 valeurs maximum.
* ***SET(valeur\_possible1,valeur\_possible2,…)***  
  Stocké sur 1, 2, 3, 4 ou 8 octets, selon le nombre de solutions possibles (de 0 à 64 valeurs maximum).

## Installation de la base de données existante « sakila »

La BD sakila a été développée par Mike Hillyer (ex membre de l’équipe de documentation de MySQL AB) afin d’être utilisée comme exemple dans les livres, tutoriels, articles…

Elle peut être utilisée afin de mettre en œuvre les outils liés à des BD MySQL / MariaDB, et notamment afin de découvrir les concepts de tables, de vues, de procédures…

**BD-22-EX** : Installer et explorer la BD sakila offerte par MySQL.

**Notions**:

* Les fichiers pour l’installation de la BD Sakila se trouvent à cette adresse : <https://dev.mysql.com/doc/index-other.html>  
  Trois fichiers sont fournis :
  + ***sakila-schema.sql*** : contient toutes les déclarations « ***CREATE*** » pour créer la structure de la database, incluant des tables, des vues, des procédures et des triggers.
  + ***sakila-data.sql*** : contient toutes les déclarations pour alimenter la structure de la DB, avec la définition de triggers qui doivent être créés après le chargement des données.
  + ***sakila.mwb*** : contient le modèle de données, la conception de la structure de la DB.
* Installation de la BD sakila:

1. Dézipper le fichier ***sakila-db.zip***
2. Créer la structure de la DB via Workbench:
   1. Se connecter au serveur MariaDB (inclus dans le package XAMPP)
   2. Cliquer sur ***File***, « ***Run SQL Script***»
   3. Choisir : ***sakila-schema.sql***
   4. Cliquer sur ***Open***
3. Alimenter la DB via Workbench:
   1. Faire la même chose que ci-dessus, mais pour : ***sakila-data.sql***

* Plus d’informations concernant l’installation de la BD sakila : <https://dev.mysql.com/doc/sakila/en/sakila-installation.html>

# Méthodologie et outils pour le développement d’un modèle de données et sa BD

## Introduction

Tout comme il existe de multitudes de méthodologies pour le développement d’un logiciel, il y a beaucoup de méthodologies et outils spécifiques au développement d’un modèle de données.

Pour ce cours, il vous est proposé une méthodologie qui s’insère dans notre cycle de vie de développement d’un logiciel tel que définit au §6.2.

Figure 35 : méthodologie de développement d’un modèle de données et sa BD

Les activités associées au développement d’un modèle de données et à sa BD sont :

* Spécification des besoins (activité générique nécessaire pour le développement d’une application multimédia) ;
* Modèle conceptuel des données ;
* Modèle physique des données ;
* Implémentation et déploiement de la DB sur le ou les serveurs de développement (ou de production).

## Spécification des besoins

La spécification des besoins (ou spécification des exigences) correspond au cahier des charges de votre application multimédia.

Son but est d’assurer que l’application développée corresponde aux besoins des utilisateurs, tout en tenant compte du contexte opérationnel (technologie, nombre et types d’utilisateurs…).

Pour ce faire, une spécification des besoins est généralement le résultat d’une analyse qui identifie les exigences auxquelles l’application devra répondre, tout au long de son développement.

Dans une spécification des besoins, il est spécifié :

* les besoins fonctionnels associés à l’application à développer, d’un point de vue des utilisateurs finaux et du contexte opérationnel ;
* les besoins techniques associés à l’application à développer ; par exemple, le type de technologie à mettre en œuvre (php, MariaDB)…

La spécification des besoins fonctionnels sera l’élément d’entrée pour concevoir le modèle de données permettant de développer les fonctionnalités de votre application.

Dans le cadre de ce cours, la spécification des besoins techniques sera généralement déjà déterminée pour vous. La technologie à utiliser, le contexte opérationnel vous sera proposé, comme par exemple « L’application devra être accessible sur le web, via un browser. La BD devra être implémentée sur MariaDB, et accessible via phpMyAdmin. ». Ainsi, la spécification des besoins techniques sera un élément d’entrée à votre modèle physique des données, fait pour un SGBD particulier.

Un besoin fonctionnel, ou une exigence fonctionnelle, est représentée textuellement et/ou visuellement.   
Il existe de nombreux outils de description visuels.

Dans le cadre de cours, une représentation graphique très intéressante vous est proposée : la notation UML et ses diagrammes de cas d’utilisation (« use case diagram » en anglais).

Pour ce cours, vous pourrez décrire les besoins fonctionnels soit de manière textuelle, soit en UML, via les diagrammes de cas d’utilisation.

Voici un exemple de cas d’utilisation :

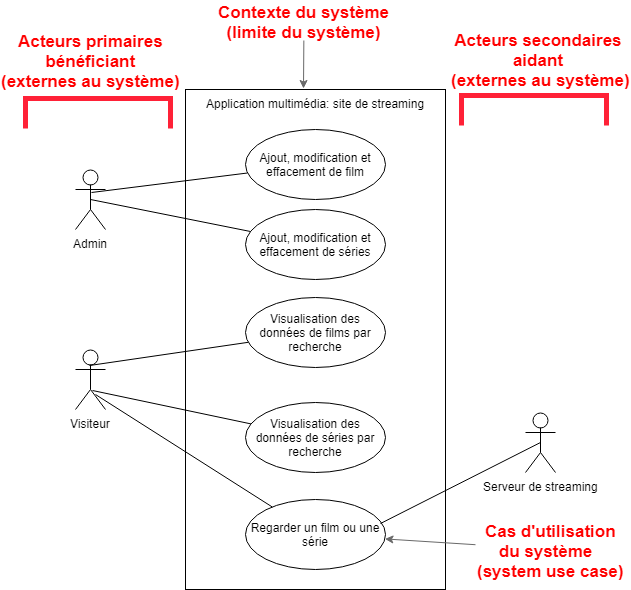


Figure 36 : exemple de diagramme de cas d’utilisation pour représenter les besoins fonctionnels

## Modèle conceptuel des données

Le but de ce modèle est :

* De fournir une description graphique « haut-niveau » des informations à enregistrer dans une BD
* D’identifier les objets, leurs propriétés, et les associations entre ces objets afin de répondre aux besoins des utilisateurs
* De se dégager de l’organisation logique (clé primaires et clés étrangères) et physique des données (type de données qui peut dépendre du SGBD, supports d’information : nombre de BDs, localisation des BDs, capacité des supports…)

Différentes représentations standards existent afin de créer un modèle conceptuel des données, comme le modèle entité-association, ou les diagrammes de classes en UML.

Pour ce cours, nous utiliserons :

* le modèle entité-association (« EER model ») pour représenter les objets et les associations entre ceux-ci
* les notations UML pour représenter les cardinalités des associations, ainsi que les différents types d’association (spécialisation, agrégation…).

L’outil utilisé pour créer nos modèles conceptuels est draw.io.

C’est un logiciel « open source » qui peut s’utiliser soit directement en ligne <https://www.draw.io/> soit en l’installant localement sur sa machine : <https://about.draw.io/integrations/>

Nous utiliserons principalement les familles de formes (« shapes ») « UML » et « Entity Relation » de l’outil draw.io.



Figure 37 : visualisation des « shapes » utilisée au sein du menu de draw.io

Voici les principaux composants d’un modèle conceptuel de données :

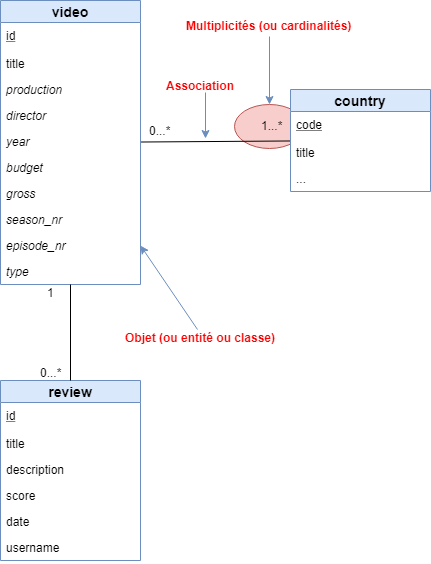


Figure 38 : diagramme conceptuel de données et ses définitions

Au sein d’un de nos modèles conceptuels de données, ces multiplicités (ou cardinalités) sont possibles :

Table 6 : multiplicités pour un modèle conceptuel de données

| **Multiplicités** | **Exemple** |
| --- | --- |
| 0…\* | * multiplicité minimum 0 : 1 pays peut ne pas être associé à une vidéo * multiplicité maximum \* : 1 pays peut être associé à plusieurs vidéos (\*) |
| 1… \* | * multiplicité minimum 1 : 1 vidéo doit être associée au minimum à 1 pays * multiplicité maximum \* : 1 vidéo peut être associée à plusieurs pays |
| 1 | * multiplicité minimum 1 : 1 revue doit être associée au minimum à une vidéo * multiplicité maximum 1 : 1 revue peut être associé à une et une seule vidéo |

## Modèle physique des données

Le but de ce modèle est :

* De fournir une description graphique détaillée des informations à enregistrer dans une BD
* D’implémenter le modèle conceptuel de données en prenant en compte les informations logiques et la technologie mise en œuvre :
  + identification des clés primaires, des clés étrangères, des contraintes d’intégrité…
  + définition du type des données, leur longueur, valeur par défaut…
* D’assurer l’indépendance du modèle des données par rapport à la couche matérielle (type de disque physique)…

L’outil utilisé pour créer nos modèles physiques est MySQL Workbench.

C’est un logiciel « open source » qui peut s’installer localement sur sa machine : <https://dev.mysql.com/downloads/workbench/>

Nous utiliserons la fonctionnalité de création de modèle EER de MySQL Workbench, en utilisant la visualisation des associations selon le langage UML.

Voici les principaux composants d’un de nos modèles physiques de données :

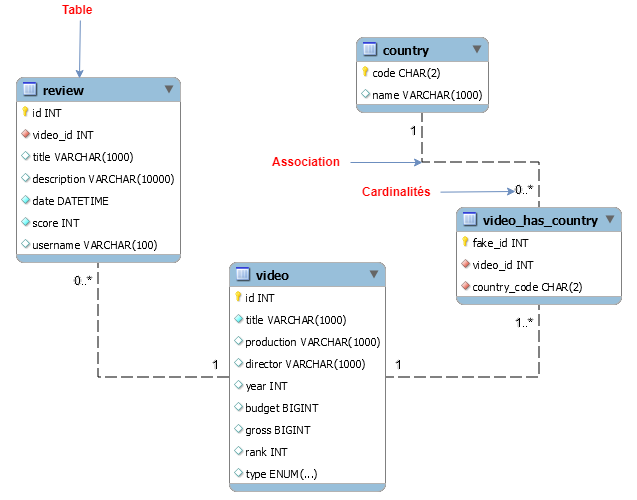


Figure 39 : modèle physique de données et ses définitions

Les notions de table, association et cardinalité sont définies au §5.3.

## Implémentation et déploiement de la DB

Le but de cette étape est :

* D’utiliser le modèle physique de données pour implémenter la BD sur le ou les SGBD à mettre en œuvre ;
* De rendre la BD disponible pour l’IHM.

L’outil utilisé pour générer nos BDs, sur base de modèles physiques de données, est MySQL Workbench.

Pour rappel, c’est un logiciel « open source » qui peut s’installer localement sur sa machine : <https://dev.mysql.com/downloads/workbench/>

Nous utiliserons la fonctionnalité de synchronisation de modèle EER de MySQL Workbench. La fonction ***Database, Synchronize Model…*** fonctionne seulement si :

* le nom de la DB à créer a été donné au schéma physique (au sein du modèle EER).
* on utilise une connexion d’administrateur permettant de créer une BD et d’y accéder
* on enlève ***VISIBLE*** pour les index (l’option ***VISIBLE*** n’existe pas pour MariaDB)

# Création d’une application multimédia comprenant une base de données et son interface web

## Développement d’une nouvelle BD

**BD-19-SHOW** : Nous souhaitons gérer un nouveau service en ligne de streaming de films et séries. Dans un premier temps, nous nous concentrerons simplement sur la revue des vidéos, de manière simpliste, sans prendre en compte les utilisateurs, la sécurité… Appliquons ensemble la méthodologie de développement de son système d’information et principalement, de son modèle de données.

**BD-19A-SHOW** : Identification des besoins / System Requirements Specification :

* Ajout, modification et effacement de films ;
* Visualisation des données de films par recherche : pays de production, année de production, revenus, budget…
* Ajout, modification et effacement de séries ;
* Visualisation des revues de films ;
* Visualisation des données de séries par filtres ;
* …

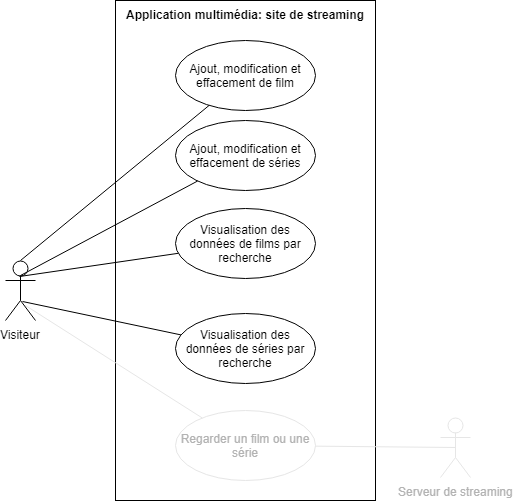


Figure 40 : diagramme de cas d’utilisation pour site de streaming extra simplifié

**BD-19B-SHOW** : Création du modèle conceptuel des données.

**Notions** :

* Utilisation de l’outil ***draw.io*** pour créer un diagramme de type « Entity Relation ».

**BD-19C-SHOW** : Création du modèle physique des données et génération du schéma de la DB.

**Notions** :

* Création de données de type ***ENUM***(MySQL Workbench) : ***ENUM(‘value1’,’value2’,’value3’)***.  
  NB : des listes déroulantes ne sont pas disponibles dans MySQL Workbench pour les ***ENUM***. Néanmoins, celles-ci sont disponibles dans phpMyAdmin.
* Date et heure automatique à la création d’un enregistrement :
  + Type de donnée à ajouter : ***DATETIME***
  + Donnée par défaut : ***NOW()***
* Correction des erreurs éventuelles lors de la synchronisation du modèle :
  + Enlever ***VISIBLE*** pour les index (l’option ***VISIBLE*** n’existe pas pour MariaDB)

## Création d’une interface web personnalisée pour une BD

Quand une BD existe, celle-ci est généralement la source de données pour une interface homme machine (IHM) permettant d’exploiter les informations d’une organisation.

Cette IHM prend de plus en plus souvent la forme d’une application web accédée via un browser. Cette application web peut être développée parmi une multitude de technologie : PHP, node.js, asp.net…

Malgré la mise en œuvre de langages de programmation (php, node.js…), il est possible de créer une application web personnalisée sans connaissance de programmation. Les BD nécessaires à votre application web sont générées pour vous par un outil. C’est par exemple le cas lorsque qu’une application web est créée à partir d’un « Content Management System » comme Wordpress.

Néanmoins, actuellement, il est très difficile de trouver un outil permettant de créer une application web pouvant s’interfacer avec ses propres bases de données, sans connaissance de programmation. En règle générale, il est nécessaire d’écrire des scripts personnalisés afin que votre IHM s’interface avec une BD personnalisée.

Etant donné que ce cours est basé sur l’utilisation de XAMPP, un outil fonctionnant par le biais de php a été investigué dans le but de permettre à un étudiant ne connaissant pas une ligne de code de PHP de créer une interface web de base, et personnalisable, associée à une BD personnelle.

Le meilleur outil trouvé début 2019 était Phreeze.

**Néanmoins, comme celui-ci n’a plus été mis à jour, depuis la mise à niveau de XAMPP et de la version de PHP associée, cet outil ne fonctionne plus. Cette partie du cours n’est donc plus proposée, sauf pour les étudiants qui souhaiteraient malgré tout tester Phreeze avec une ancienne version de XAMPP.**

**BD-20-SHOW** : Générer une application web personnalisée pour la nouvelle DB « ***streaming*** », sur base du framework « Phreeze ».

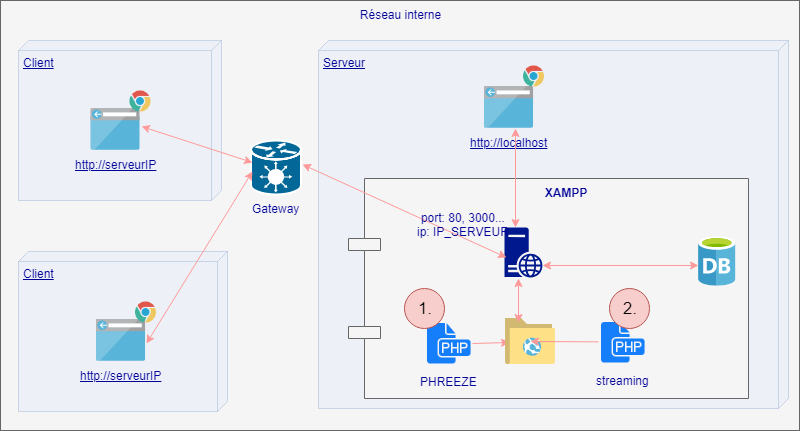


Figure 41 : visualisation de l’architecture nécessaire à la génération d’une IHM web sur base d’une BD existante via PHREEZE

**Notions** :

* Télécharger et installer Phreeze : <http://phreeze.com/phreeze/documentation/installation.php>
* Copier le fichier ***.zip*** d’installation dans le répertoire du serveur web : ***XAMPP Control Panel, Explorer, Sélectionner htdocs,***
* Extraire le ***.zip*** dans le répertoire partagé du serveur web, en renommant ***phreeze-master*** en ***phreeze***, par exemple : ***C:\xampp\htdocs\phreeze***
* Générer les scripts de votre application web en accédant au builder de l’application Phreeze :
  + Accéder via votre browser à [http://localhost/phreeze/builder/](http://localhost/phreeze/builder)
  + Entrer les données pour accéder à votre DB :

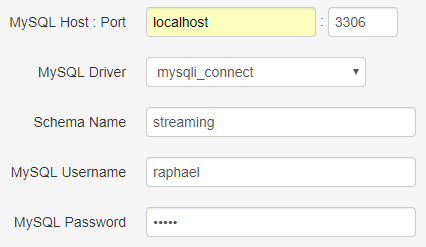


Figure 42 : génération des scripts d’une application web via PHREEZE (partie 1)

* + Générer les scripts de votre application web :

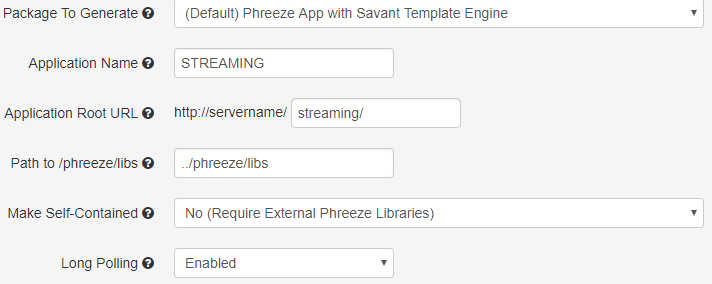


Figure 43 : génération des scripts d’une application web via PHREEZE (partie 2)

* Utilisation des scripts générés :
  + Copier le fichier ***.zip*** généré (***streaming.zip*** dans cet exemple) au sein de xampp/htdocs : ***XAMPP Control Panel, Explorer, Sélectionner htdocs,***
  + Extraire le fichier ***.zip*** généré : ***clic droit sur le fichier .zip, Extract all…***
  + Votre application web personnalisée se trouve maintenant dans le répertoire partagé du serveur web. Par exemple : ***C:\xampp\htdocs\streaming***
* Limitations de l’outil : en cas de clé primaire composite, l’outil ne fournit pas de liste déroulante pour choisir les clés étrangères.

## Projet : création d’une application multimédia personnalisée

**BD-21-EX**: Créer le système d’information pour supporter une organisation qui vous tient à cœur.

###### Consignes

* Un utilisateur devra être créé, portant le nom « admin », le password « heff », et devra avoir tous les privilèges sur la nouvelle DB. Cet utilisateur n’aura aucun droit sur les autres DB.
* Le modèle physique des données devra mettre en œuvre un minimum de 8 tables et des associations entre celles-ci.
* **BD-21A-EX**: Créer la spécification des besoins / Requirements Specification.
* **BD-21B-EX** : Créer le modèle conceptuel des données.
* **BD-21C-EX** : Créer le modèle physique des données et générer le schéma de la DB à partir de celui-ci.
* ***BD-21D-EX-OPTION*** *: Générer une application web personnalisée pour votre nouvelle DB, soit sur base du framework « Phreeze », soit sur base d’un autre outil approprié et sélectionné par vos soins.*

# Bibliographie

|  |  |
| --- | --- |
| [1] | M. -. U. o. C. S. Mannino, «Database Management Essentials,» 2018. [En ligne]. Available: https://www.coursera.org/learn/database-management/home/info. |
| [2] | D. DELÉGLISE, MySQL 5 (versions 5.1 à 5.6) - Guide de référence du développeur (Nouvelle édition), ENI, 2013. |
| [3] | solid IT, «DB-Engines,» [En ligne]. Available: https://db-engines.com/en/. |
| [4] | stackoverflow, «Developer Survey Results 2018,» [En ligne]. Available: https://insights.stackoverflow.com/survey/2018/. |
| [5] | CentraleSupélec, «Choisissez votre famille NoSQL,» [En ligne]. Available: https://openclassrooms.com/en/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462433-choisissez-votre-famille-nosql. |
| [6] | Wikipedia, «Graph database,» [En ligne]. Available: https://en.wikipedia.org/wiki/Graph\_database. |
| [7] | K. Vargas, «The Main NoSQL Database Types,» [En ligne]. Available: https://studio3t.com/whats-new/nosql-database-types/. |
| [8] | M. Hashemi, «The Infrastructure Behind Twitter: Scale,» 2017. [En ligne]. Available: https://blog.twitter.com/engineering/en\_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale.html. |
| [9] | w3schools, «SQL joins,» [En ligne]. Available: https://www.w3schools.com/sql/sql\_join.asp. |
| [10] | Wikipedia, «Liste des plus gros succès du box-office mondial,» 2018. [En ligne]. Available: https://fr.wikipedia.org/wiki/Liste\_des\_plus\_gros\_succ%C3%A8s\_du\_box-office\_mondial. |
| [11] | ISO/IEC/IEEE, «ISO/IEC/IEEE 12207:2017 - Systems and software engineering -- Software life cycle processes,» ISO/IEC/IEEE, 2017. |
| [12] | EN, «EN 50657:2017 - Railways Applications - Rolling stock applications - Software on Board Rolling Stock,» EN, 2017. |
| [13] | Wikipedia, «Iterative and incremental development,» [En ligne]. Available: https://en.wikipedia.org/wiki/Iterative\_and\_incremental\_development. |
| [14] | A. De Groote, «Informatique (2) - Bases de données,» 2015-2016. |
| [15] | Scott W. Ambler, «The Agile System Development Life Cycle (SDLC),» [En ligne]. Available: http://www.ambysoft.com/essays/agileLifecycle.html. |