

# Introduction aux bases de données

## 1. Introduction

[max.silberztein@univ-fcomte.fr](mailto:max.silberztein@univ-fcomte.fr)

# Références

- Notes de cours : Sylvie Damy, Julien Henriet (UFC)
- Database Design for Mere Mortals: 25th Anniversary Edition, M. Hernandez
- Mastering 3NF, A Comprehensive Guide to Efficient Relational Database Design, M. Kirshteyn
- Course Introduction & Relational Model (CMU Intro to Database Systems / Fall 2021, Carnegie Mellon University):  
<https://www.youtube.com/watch?v=v4bU6n97Vr8>

# Plan du cours

## 1. Introduction

1. Historique
2. Définitions

## 2. Bases de données simples

1. Intégrité
2. Normalisation et dénormalisation
3. Analyse statistique
4. Fonctionnalités des bases de données

## 3. Bases de données SQL

1. Modélisation : modèle conceptuel et logique
2. Implémentation : type des données, modèle physique
3. Gestion d'une base de données : phpMyAdmin
4. Langage de requête SQL

# 1. Historique

# Pourquoi des bases de données ?

- Stockage permanent des données
  - Mise à jour des données
  - Accès facile
  - Présentation des données
- 
- Qualité des données
  - Sécurité
  - Partage

- Dans les années 1960, chaque programmeur gérait ses données comme il le voulait. Les données étaient traitées par son programme
- Mais les données ne pouvaient pas être utilisées par d'autres programmes. Le programmeur était la seule personne capable à comprendre comment les données étaient représentées, et la seule personne capable d'accéder aux données de son programme
- Si le logiciel devenait obsolète, toutes ses données étaient perdues

# Bases de données SQL



- Dans les années 1970, Trois programmeurs chez **IBM** révolutionnent la façon dont on pense et traite les données.
- Ted Codd invente une nouvelle façon de représenter et de structurer les données qui deviendra le standard : le modèle de base de données relationnelle (RDBMS)
- Donald Chamberlin et Raymond Boyce conçoivent le langage de programmation **SQL** (Structured Query Language) pour gérer les données représentées dans une base de donnée relationnelle.

# Bases de données SQL



- En 1979, la société ORACLE vend un serveur qui implémente une base de donnée relationnelle
- La société Rational Software vend la première version d'un interpréteur SQL
- ORACLE achète Rational Software
- ORACLE deviendra le numéro 1 mondial des systèmes de bases de données pendant plusieurs dizaines d'années, et le numéro 1 mondial des vendeurs de logiciel.





# Bases de données SQL

- Entre 1970 et 1980, la méthode conceptuelle MERISE devient un standard utilisé en France pour formaliser les données et concevoir des bases de données relationnelles.
- Mais la méthode MERISE a des concurrents ; elle ne deviendra donc pas le standard mondial.
- Nous suivrons les recommandations de la méthode MERISE

# Bases de données NoSQL

- Les systèmes de bases de données qui n'utilisent pas SQL
- Beaucoup plus flexibles que les bases de données SQL
- Les tableurs, ex. **Excel** ne nécessitent pas de programmation
- Les tables (rigides) sont remplacées par des documents (flexibles), cf. **FireBase**
- Les données sont représentées par des objets structurés (OOP)
- Les données sont sauvegardées au format **json**, ce qui facilite la synchronisation en ligne (« cloud computing »)

C11 <L> TOTAL 25

| A         | B   | C     | D        |
|-----------|-----|-------|----------|
| ITEM      | NO. | UNIT  | COST     |
| MUCK RAKE | 43  | 12.95 | 556.85   |
| BUZZ CUT  | 158 | 6.75  | 1066.50  |
| TOE TONER | 250 | 49.95 | 12487.50 |
| EYE SNUFF | 2   | 4.95  | 9.90     |
| SUBTOTAL  |     |       | 13155.50 |
| 9.75% TAX |     |       | 1282.66  |
| TOTAL     |     |       | 14438.16 |

# Les tableurs

- En 1978, Dan Bricklin a développé la première application de tableau de calculs automatique : **VisiCalc**
- Mitch Kapor, ancien employé de VisiCalc, créé en 1983 la société Lotus Software qui licencie le tableur **Lotus 1-2-3**
- Microsoft lance en 1982 le tableur **Multiplan**, puis en 1984 **Excel**
- Très vite, les tableurs sont utilisés pas seulement comme feuilles de calcul, mais aussi comme **bases de données**
- Aujourd'hui, Excel est le standard, il existe des versions compatibles libres et gratuits





# Les bases de données aujourd'hui

- Les bases de données simples sont rangées dans des **tableurs**
- La plupart des bases de données relationnelles sont rangées dans des bases de données **SQL**
- La plupart des bases de données dans le « cloud » sont des bases noSQL (ex. FireBase, Google Bigtable, Amazon DocumentDB)

=> Nous étudierons les bases de données tableurs et SQL

## 2. Définitions

# Une base de données est :

- un ensemble d'informations modélisant les objets d'une partie du monde réel et servant de support à une application informatique
- une collection organisée, interrogeable et persistante de données ou d'informations, concernant un thème donné pour l'entreprise

# Bases de données : définitions

Du point de vue plus informatique :

- Ensemble de **données** et de **métadonnées**, (description des types de données) rassemblées dans des **tables**
- Ensemble de fichiers contenant les **entités** (ex. produits), associés à des **attributs** (propriétés), avec des **relations** (liens) potentiels entre entités de différentes tables

# SGBD : Système de Gestion des Bases de Données

Ensemble de logiciels fournissant des fonctionnalités pour :

- décrire,
  - mémoriser,
  - manipuler,
  - filtrer, trier, traiter des ensembles de données,
- 
- en contrôler l'accès,
  - partager ces données,
  - gérer des pannes, etc.



# Les différents rôles dans les systèmes de gestion de bases de données

1. Concepteurs
2. Administrateurs
3. Utilisateurs
4. Développeurs

# 1. Concepteurs

- **Concepteurs logiques**

s'intéressent à la modélisation du domaine. Quelles sont les informations à décrire ? Quelles sont les relations entre informations ? conception de la base de données

- **Concepteurs physiques**

projetent le modèle conceptuel vers le modèle physique, ex. combien de bits a-t-on besoin pour décrire telle ou telle information

## 2. Administrateurs

- garantissent l'intégrité des données
- garantissent la sécurité des données
- améliorent la performance
- gèrent les pannes
- gèrent les mises à jour et migrations

# 3. Utilisateurs

- accèdent à la base de données rangée sur un ordinateur « serveur » à partir d'ordinateurs « Clients »
- ne travaillent qu'à travers le biais d'interfaces et n'ont pas forcément connaissance des problèmes techniques liés à la gestion du serveur et/ou de la base de données
- les utilisateurs plus expérimentés peuvent écrire leurs propres requêtes SQL pour interroger la base de données, mais ne la modifient pas

## 4. Développeurs d'applications logicielles

- réalisent des applications logicielles utilisées comme interfaces entre les utilisateurs et les bases de données
- Ces interfaces permettent aux utilisateurs finaux d'accéder à la base de données
  - => soit pour extraire des informations
  - => soit pour la gérer, ex. ajouter ou supprimer des données