

Cours 5 : Organiser des données

Virgile Reigner

Pratiques Numériques (L2 et L3) – 2022-2023

Université de Picardie Jules Verne

Introduction

- Les données organisées de manière systématique peuvent prendre la forme d'une table pour faciliter leur traitement
- La répétition des données peut être évitée par l'ajout d'autres tables liées à la première par des identifiants
- Eviter les répétition permet de limiter les fautes de saisie et de faciliter la mise à jour des données
- On appelle base de données un ensemble de données liées entre elles (notamment dans un cadre informatique)

I/ Les principes des bases de données

- 1956 : disque dur par IBM
- 1965 : Architecture Ansi/Sparc par Charles Bachman
- 1966 : *Information Management System* par IBM
- 1970 : Modèle relationnel par Edgar Frank Codd
- 1974 : System R par IBM
- 1975 : Diagramme entité-association par Peter Chen
- 1987 : Le SQL devient une norme ISO
- 2004 : BigTable par Google et Dynamo par Amazon

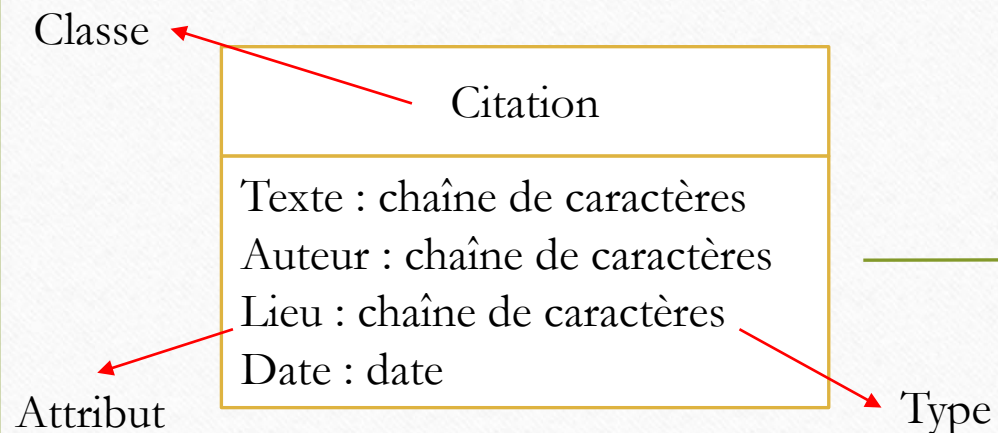


I/ Les principes des bases de données

- Les SGBD relationnels sont les plus couramment utilisés
- Des modèles concurrents apparaissent (mouvement NoSQL) :
 - Les SGBD document (XML, JSON) utilisent Xpath, Xquery et JSONpath
 - Les SGBD graphe (RDF) utilisent SPARQL
- Adaptation des modèles relationnels au Big Data (NewSQL)
- Créations de modèles complémentaires (théorème CAP)

II/ Fonctionnement d'un SGBDR

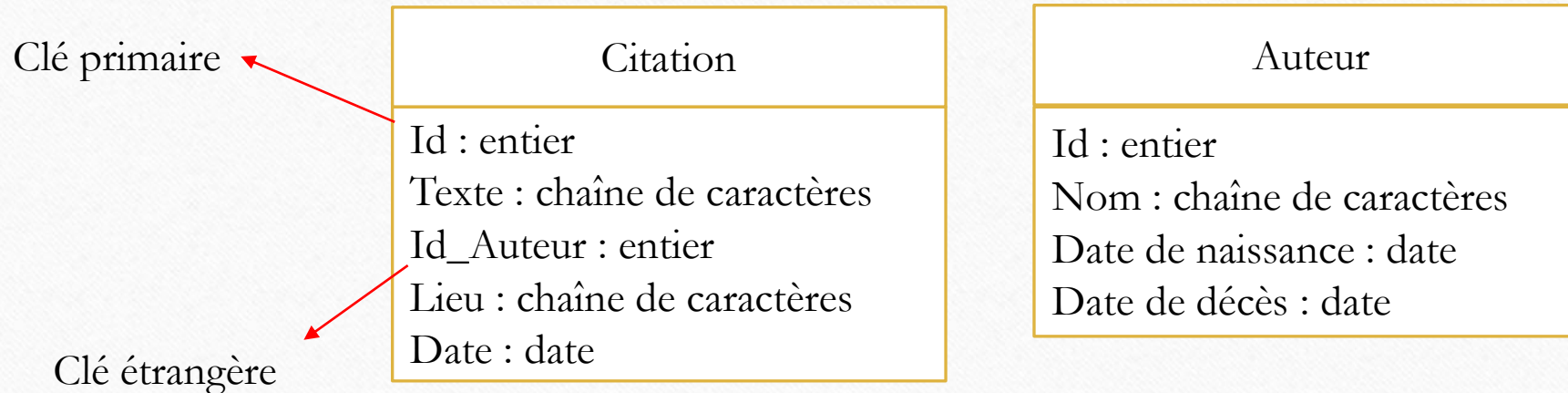
- Les données sont organisées en **tables**
- Les lignes représentent les **entités** et les colonnes les **attributs**
- Un groupe d'entités ayant des attributs communs s'appelle une **classe**



- Chaque enregistrement de la base de données correspondant à cette classe forme une entité.
- Cette entité dispose d'attributs conformes à sa classe, ils forment les champs de l'enregistrement
- L'ensemble des entités et des attributs de cette classe forment une table

II/ Fonctionnement d'un SGBDR

- On appelle **clé primaire** l'identifiant unique de chaque entité
- On appelle **clé étrangère** la mention d'une clé primaire parmi les attributs d'un enregistrement



II/ Fonctionnement d'un SGBDR

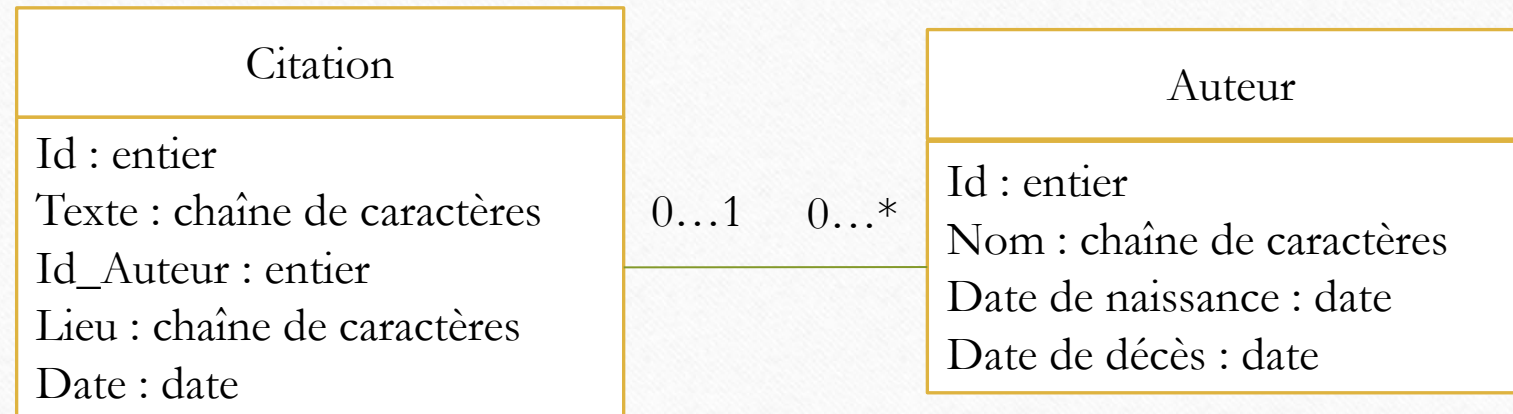
- On appelle **relation** le lien entre deux classes
- Elle peut avoir plusieurs cardinalités :
 - 1-1 si chaque entité ne peut être associé qu'à une seule autre et réciproquement
 - Par exemple la carte d'identité : elle est liée à une seule personne et on ne peut pas en avoir plusieurs en cours de validité en même temps
 - 1-N si chaque entité ne peut être associé qu'à une seule autre, mais pas réciproquement
 - Par exemple un diplôme de licence : il est attribué à une seule personne mais on peut en avoir plusieurs
 - N-N si chaque entité peut être associée à plusieurs autres (on crée alors une **table de relation**)
 - Par exemple une maison : vous pouvez en posséder plusieurs et leur propriété peut être partagée

II/ Fonctionnement d'un SGBDR

- Le langage utilisé est le *Structured Query Language* (**SQL**)
- Il est composé de trois sous-ensemble :
 - **Langage de Définition de Données** pour manipuler la structure des données
 - **Langage de Manipulation de Données** pour ajouter, supprimer ou sélectionner des données
 - **Langage de Contrôle de Données** pour gérer les mises à jour et les autorisations
- On appelle **transaction** le fait de modifier un élément de la base de données et **commit** le fait de valider cette modification
- Une **requête** est l'interrogation d'une base de données pour obtenir des informations

III/ Que faire avec des données ?

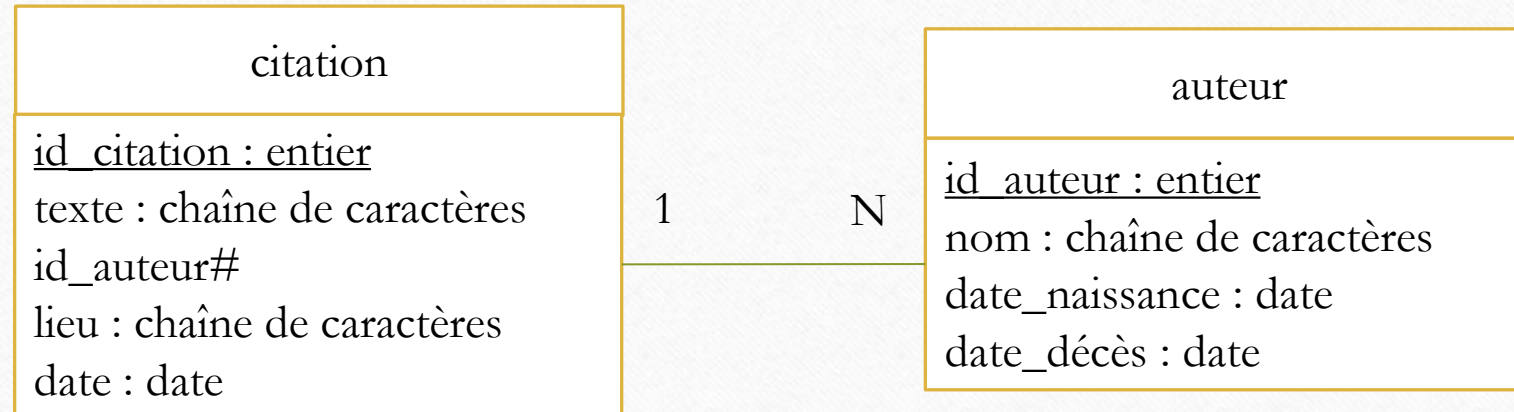
- A partir d'un échantillon, créer un **modèle conceptuel** :



- Ce modèle doit représenter les classes, les attributs, leur type et le degré de relation entre les classes.

III/ Que faire avec des données ?

- Ce modèle permet de créer un **modèle logique relationnel** :



- Ce modèle met en valeur les clés primaires et clés étrangères, ainsi que la cardinalité des relations (1-1, 1-N ou N-N)

III/ Que faire avec des données ?

- Une dernière étape : le **modèle physique** implémenté dans un SGBD
- Exemples de logiciels :
 - Par Oracle : Oracle Database, MySQL
 - Par IBM : DB2
 - Libre de droits : PostgreSQL, MariaDB, SQLite
 - NoSQL : MongoDB, BaseX, eXistDB
 - Interface intégrée : Microsoft Access, FileMaker Pro, Paradox
- Des combinaisons avec d'autres logiciels : architecture LAMP

Quelques conseils

- Utiliser un SGBD pour optimiser la gestion de données
- Créer un modèle UML à partir de <https://app.diagrams.net> (logiciel libre)
- Les attributs d'une même entité ne doivent pas contenir deux fois une même information
- Si la valeur d'un attribut dépend logiquement de celle d'un autre attribut, créer une classe supplémentaire
- Un format interopérable pour les données tabulaires : le CSV