

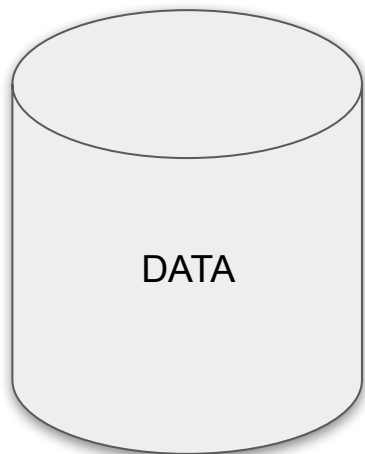
# Introduction aux bases de données relationnelles et au langage SQL

Mathilde Boltenhagen

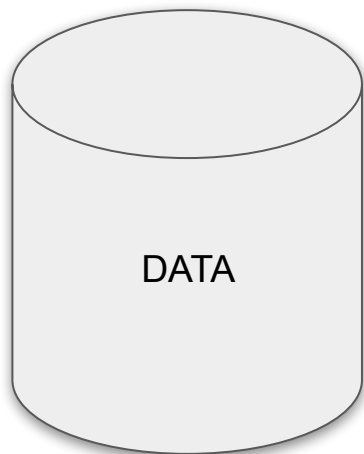
16 avril 2019

LSV, CNRS, ENS Paris-Saclay, Inria, Université Paris-Saclay



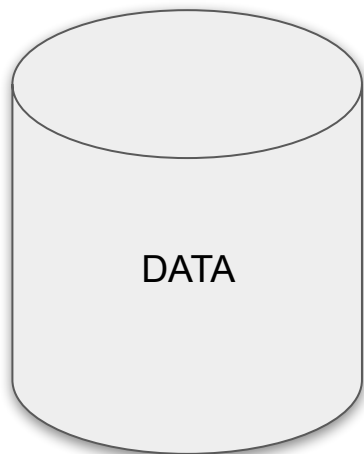


Une base de données est un ensemble de données stockées dans un format plus ou moins défini accessible à l'utilisateur.



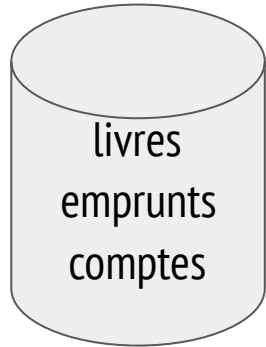
Une base de données est un ensemble de données stockées dans un format plus ou moins défini accessible à l'utilisateur.

> Big Data - NoSQL

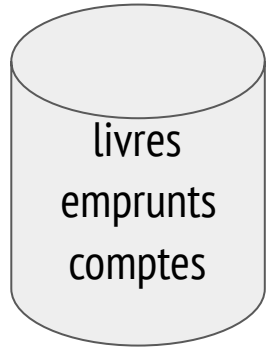


Une base de données est un ensemble de données stockées dans un **format plus ou moins défini** accessible à l'utilisateur.

- > Big Data - NoSQL
- > Base de données relationnelles



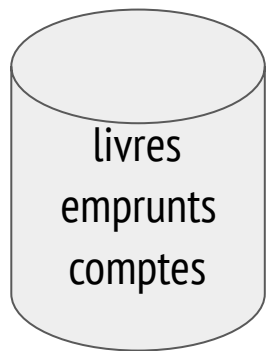
Libraire



Libraire



Magasins / Entreprises



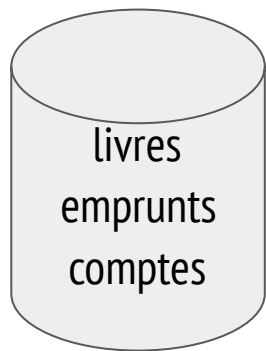
Librairie



Magasins / Entreprises



Applications



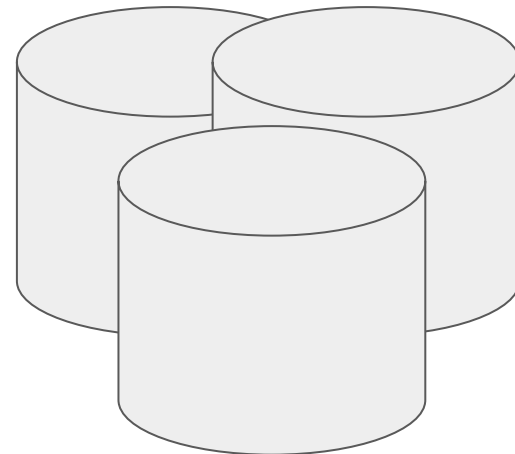
Librairie



Magasins / Entreprises



Applications



Partout



- > gérer les employés, les commandes
- > faire fonctionner une application
- > faire des études statistiques

- > gérer les employés, les commandes
- > faire fonctionner une application
- > faire des études statistiques
  
- > stocker des données structurées et utilisables

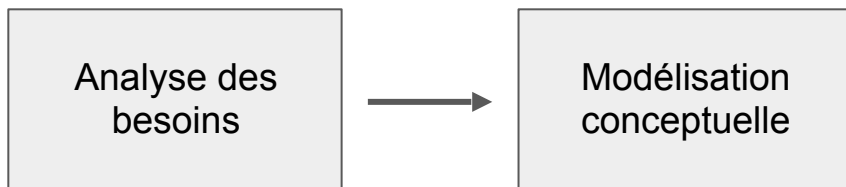


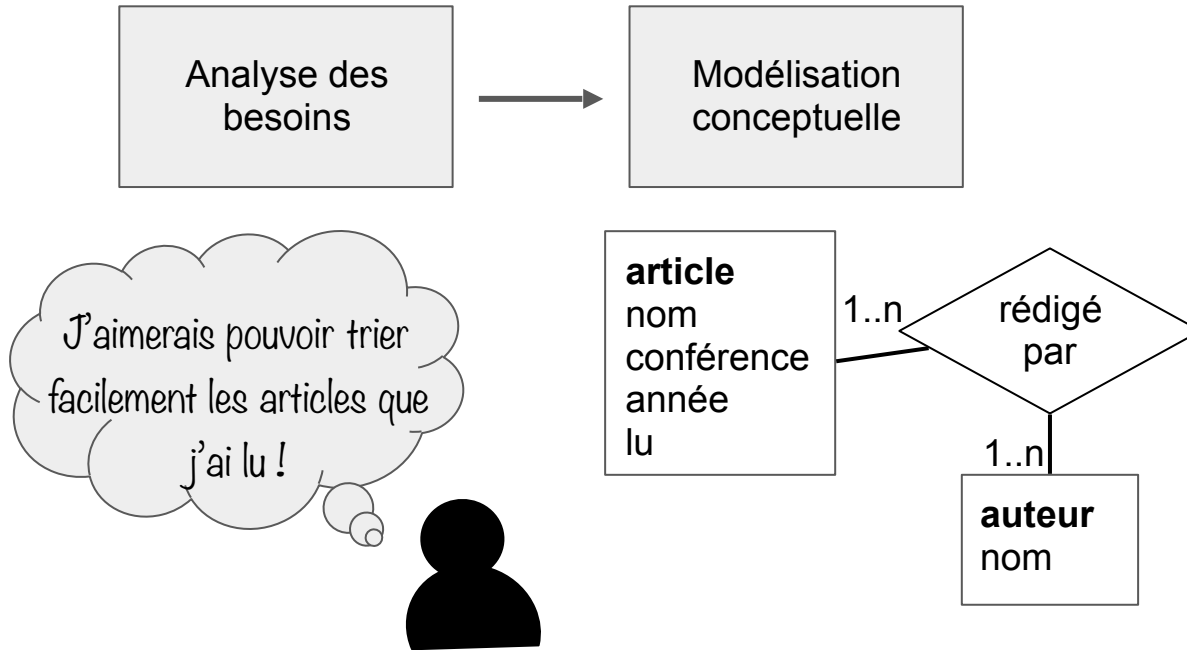
Analyse des  
besoins

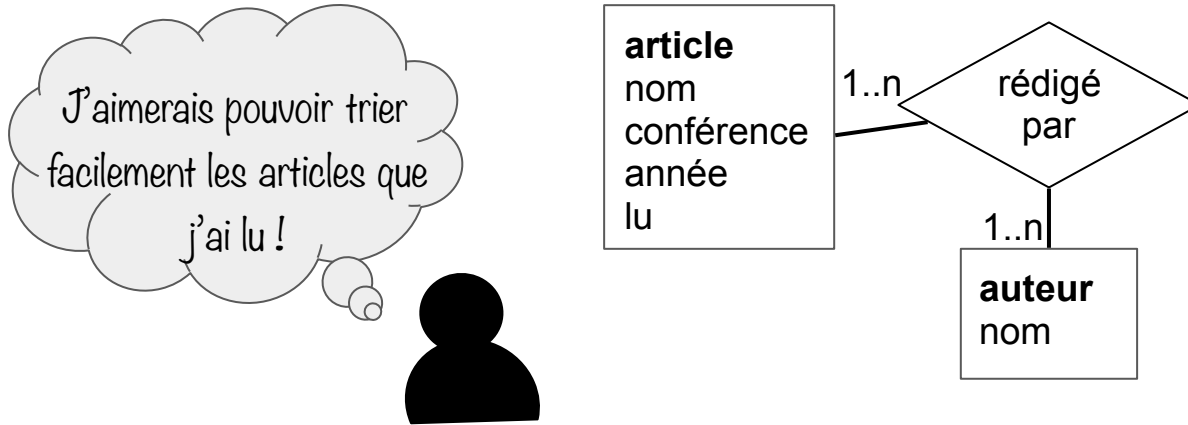
Analyse des  
besoins

J'aimerais pouvoir trier  
facilement les articles que  
j'ai lu !





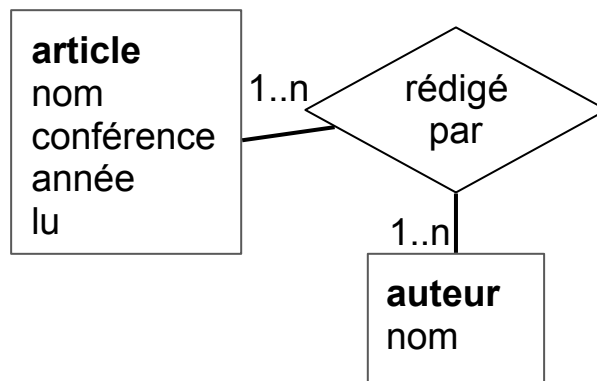








J'aimerais pouvoir trier facilement les articles que j'ai lu !



```

>> create table article
(
  nom varchar,
  conference varchar,
  annee number
)
...
  
```

### **article**

#id number

nom varchar

conference\* number

annee number

lu boolean

les attributs avec # sont  
les clés primaires

### **article**

#id number  
nom varchar  
conference\* number  
annee number  
lu boolean

les attributs avec # sont  
les clés primaires

<b>article</b>
#id number
nom varchar
conference* number
annee number
lu boolean

**id** est la clé primaire de  
la table **article**

les attributs avec **#** sont  
les clés primaires

les attributs avec **\*** sont  
les clés étrangères, ils  
font références à une  
autre table

### **article**

#id number  
nom varchar  
conference\* number  
annee number  
lu boolean

**id** est la clé primaire de  
la table **article**

les attributs avec **#** sont  
les clés primaires

les attributs avec **\*** sont  
les clés étrangères, ils  
font références à une  
autre table

### **article**

#id number  
nom varchar  
conference\* number  
annee number  
lu boolean

**id** est la clé primaire de  
la table **article**

**conference** est une clé étrangère.  
Elle fait référence à la table  
**conference**, et plus précisément à  
l'identifiant d'une conférence

les attributs avec **#** sont  
les clés primaires

les attributs avec **\*** sont  
les clés étrangères, ils  
font références à une  
autre table

le type des données est  
indiqué

### **article**

#id number  
nom varchar  
conference\* number  
annee number  
lu boolean

**id** est la clé primaire de  
la table **article**

**conference** est une clé étrangère.  
Elle fait référence à la table  
**conference**, et plus précisément à  
l'identifiant d'une conférence

les attributs avec **#** sont  
les clés primaires

les attributs avec **\*** sont  
les clés étrangères, ils  
font références à une  
autre table

le type des données est  
indiqué

### **article**

#id number  
nom varchar  
conference\* number  
annee number  
lu boolean

**id** est la clé primaire de  
la table **article**

**conference** est une clé étrangère.  
Elle fait référence à la table  
**conference**, et plus précisément à  
l'identifiant d'une conférence

**id** est un nombre  
**nom** est une chaîne de  
caractères



- > langage de programmation permettant de faire des requêtes sur un  **système de gestion de base de données**
- > MySQL

> langage de programmation permettant de faire des requêtes sur un **système de gestion de base de données**

> MySQL

### Exemple :

Ai-je lu l'article intitulé "[Relational database: a practical foundation for productivity](#)" ?

```
SELECT lu FROM article WHERE nom = 'Relational database: a pr..'
```

- > télécharger le script SQL à l'adresse : <https://boltmaud.github.io/info/TODO>
- > Ouvrir un terminal (cmd sur Windows) et aller à l'emplacement du fichier SQL (commandes **dir** et **cd** <nom\_dossier> pour se déplacer )
- > taper **mysql u root -p** (si ça ne marche pas, ajouter c:/xampp/mysql/bin/ dans la variable d'environnement PATH)
- > lancer @nomDuFichierSQL

Vous avez maintenant chargé une première base de données à l'aide du terminal ! Nous verrons la version graphique dans la suite de la formation.

Afficher les tables :

```
show tables ;
```

Afficher les tables :

```
show tables ;
```

Afficher les valeurs d'une table :

```
SELECT * FROM <nomDeLaTable> ;
```

Afficher les tables :

```
show tables ;
```

Afficher les valeurs d'une table :

```
SELECT * FROM <nomDeLaTable> ;
```

Sélectionner certains attributs d'une table :

```
SELECT <attribut1>, <attribut2> FROM <nomDeLaTable> ;
```

Afficher des valeurs avec une condition sur un attribut :

```
SELECT * FROM <nomDeLaTable> WHERE <attribut1> = <valeur> ;
```

Exercice : Affichez les articles publiés en 2000

Afficher des valeurs avec une condition sur un attribut :

```
SELECT * FROM <nomDeLaTable> WHERE <attribut1> = <valeur> ;
```

Exercice : Affichez les articles publiés en 2000

```
SELECT * FROM article WHERE annee=2000 ;
```



Afficher des valeurs avec une condition sur un attribut :

```
SELECT * FROM <nomDeLaTable> WHERE <attribut1> = <valeur> ;
```

Exercice : Affichez les articles publiés en 2000

```
SELECT * FROM article WHERE annee=2000 ;
```

Exercice : Afficher les articles dont le champ conférence n'est pas renseigné

Afficher des valeurs avec une condition sur un attribut :

```
SELECT * FROM <nomDeLaTable> WHERE <attribut1> = <valeur> ;
```

Exercice : Affichez les articles publiés en 2000

```
SELECT * FROM article WHERE annee=2000 ;
```

Exercice : Afficher les articles dont le champ conférence n'est pas renseigné

```
SELECT * FROM article WHERE conference is NULL ;
```

Que retourne la requête suivante ?

```
SELECT *  
FROM article a, conference c  
WHERE a.conference = c.id ;
```

Afficher les articles avec leurs auteurs

Afficher les articles avec leurs auteurs

```
SELECT a.nom, au.nom  
FROM article a, auteur au, articles_auteurs aa  
WHERE a.id = aa.idArticle AND au.id = aa.idAuteur ;
```

Que retourne la requête suivante ?

```
SELECT c.nom, count(a.id)
FROM article a, conference c
WHERE a.conference = c.id
GROUP BY c.nom;
```

Afficher le nombre d'articles par auteur :

```
SELECT au.nom, count(a.id)
FROM article a, auteur au, articles_auteurs aa
WHERE a.id = aa.idArticle AND au.id = aa.idAuteur
GROUP BY au.nom ;
```

## Insertion

```
INSERT INTO <nomDeLaTable> VALUES (<valeur1>,<valeur2>..);
```

## Suppression

```
DELETE FROM <nomDeLaTable> WHERE <condition> ;
```

## Mise à jour

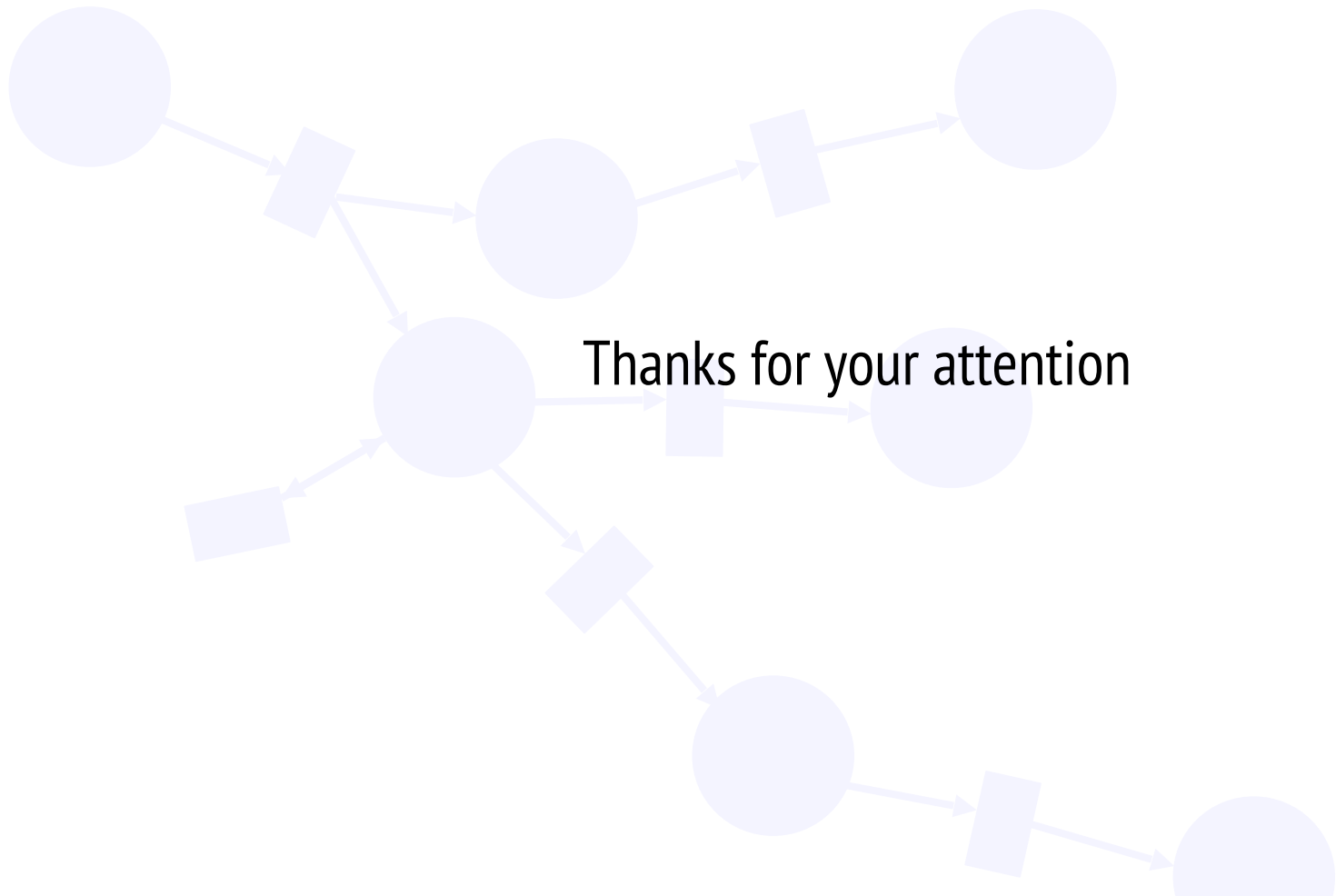
```
UPDATE <nomDeLaTable> SET <attribut1>=<valeur> WHERE <condition>;
```



> <http://localhost/phpmyadmin>

> Manipulation de l'interface

- Ajouter des données
- Supprimer des données
- Importer un fichier script



Thanks for your attention