



# Développement Avancé d'Applications Web

## – Chapitre 1 –

## Rappels sur les Bases de Données

**Dr Bouanaka Chafia**

NTIC

chafia.bouanaka@univ-constantine2.dz

### Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
NTIC	TLSI	Licence 3	Génie Logiciel (GL)

## Prérequis

- Bases de données (S4)
- Le langage SQL

## Objectif du cours

- Rappeler les concepts de bases liés à la création et la manipulation d'une BD relationnelle



# Plan du Cours

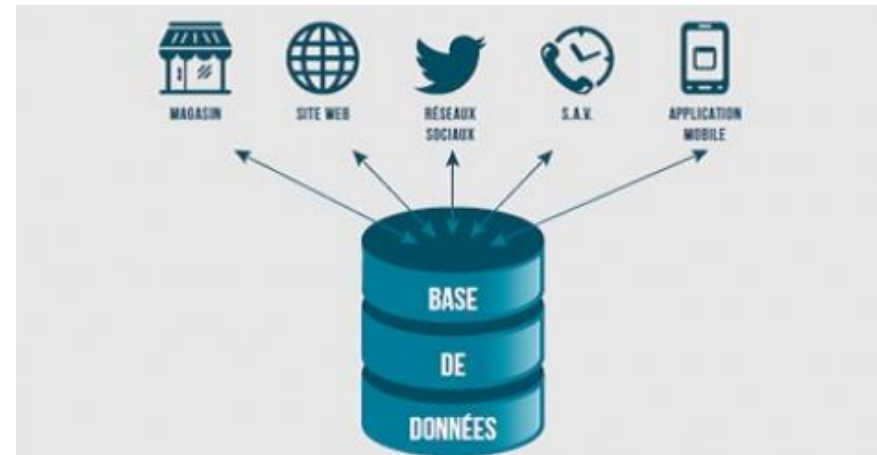
- Définitions
- Conception d'une base de données
- Algèbre relationnelle
- Organisation physique des données

# **Section 1: Définitions**

# Rappels sur les Bases de données

## Définitions : Base de données

- Une **base de données** est un ensemble de **données structurées** modélisant un univers précis et accessible à **plusieurs utilisateurs** en **même temps**.
- permettant le **stockage permanent (sur disque)** de grande quantités d'**informations**
- et faciliter l'exploitation :
  - ajout
  - mise à jour
  - recherche des données
- **Objectifs d'une BD**
  - **Elimination de la redondance des données;**
  - **Indépendance entre les programmes et les données** : on peut isoler le niveau utilisation(application) de l'organisation physique de données ;
  - **Intégration des données** de l'entreprise dans un même espace de stockage.



# Rappels sur les Bases de données

Définitions : SGBD (Système de Gestion d'une BD)

## Définition (Modèle logique de données)

- Logiciel assurant **la structuration, le stockage, la maintenance, la mise à jour et la consultation** des données d'une BD
- Il permet ainsi de gérer une BD et la partager entre plusieurs utilisateurs simultanément.

## Objectifs d'un SGBD

- **Stocker et centraliser des données** dans une BD et les mettre à disposition des utilisateurs;
- **Manipuler les données** de manière transparente pour l'utilisateur.



# Rappels sur les Bases de données

Définitions : SGBD (Système de Gestion d'une BD)

## Le SGBD assure les fonctionnalités suivantes :

- **Gestion du stockage:** tailles énormes de données, éviter les redondances
  - Définition des données via un LDD(Langage de Définition des Données)
  - Manipulation des données via un LMD(Langage de Manipulation des Données)
- **Persistance:** sauvegarde correcte des données
  - Assurer l'intégrité des données
  - Gérer les accès concurrents
- **Fiabilité:** Mécanismes de reprise sur pannes
- **Sécurité-Confidentialité:**
  - Assurer la confidentialité des données
  - Assurer la sécurité de fonctionnement
  - Contrôler les utilisateurs et leurs droits d'accès aux données

# Rappels sur les Bases de données

Définitions : SGBD (Système de Gestion d'une BD)

## Exemples de SGBD :

- **SGBD de bureautique** : Access, Base, Filemaker, Paradox
- **SGBD serveurs** : Oracle, DB2, SQL Server, PostgreSQL, MySQL



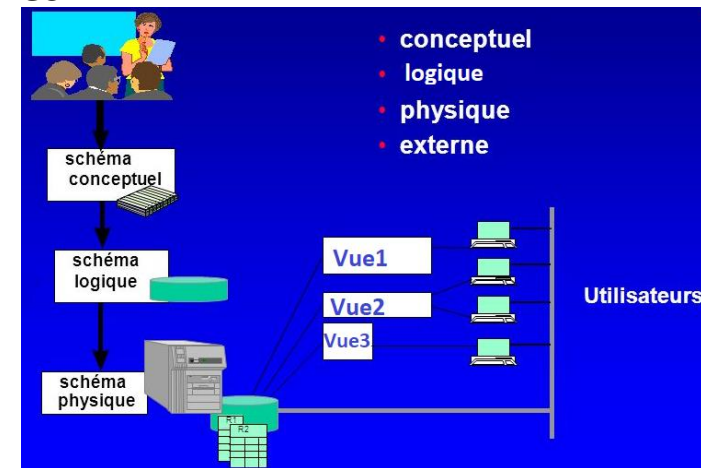
## **Section 2 : Conception d'une base de données**

# Conception d'une BD

## Niveaux de Description

On peut distinguer trois niveaux de description des données :

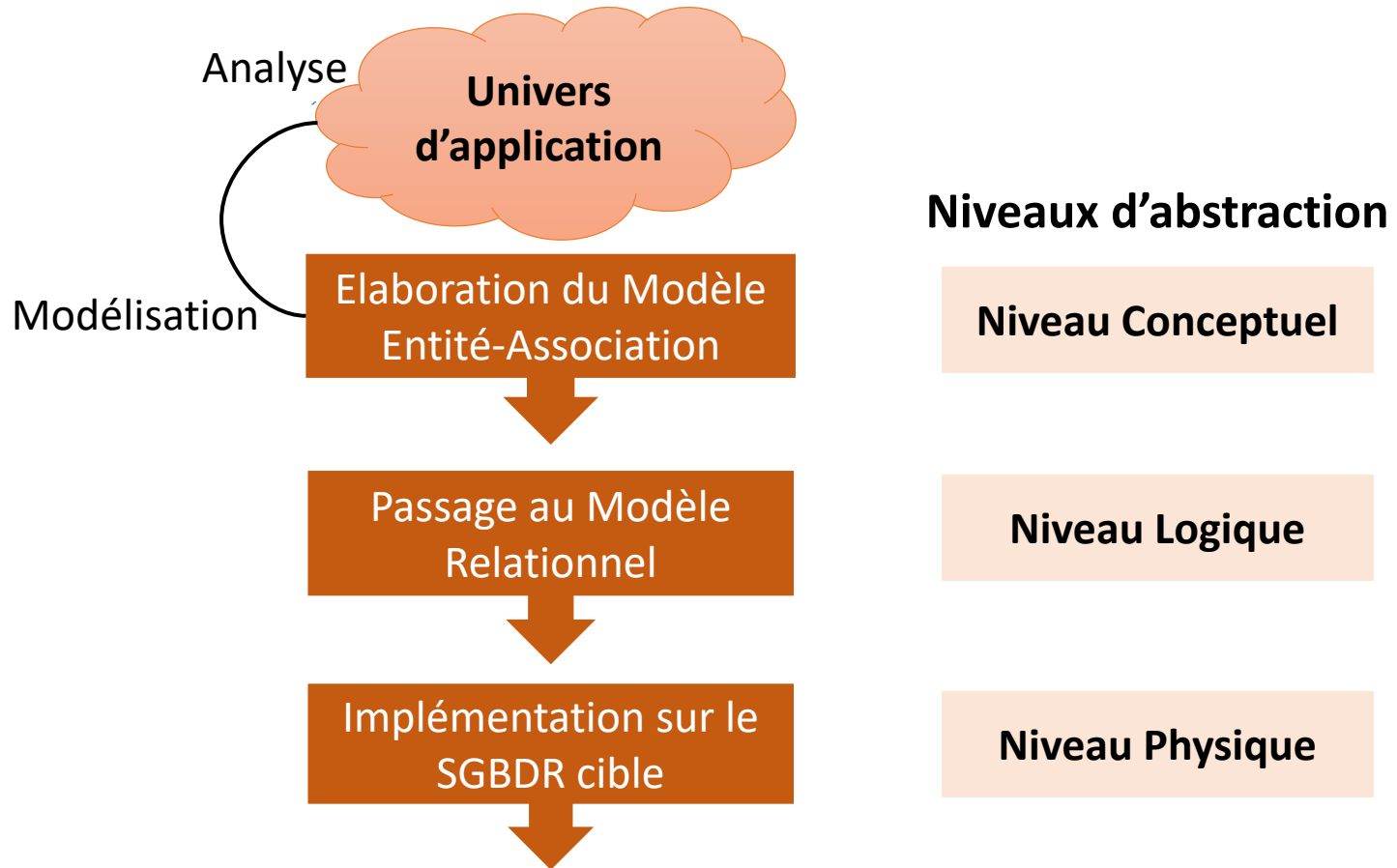
- **Niveau externe (vue) :** correspond aux différentes vues des utilisateurs
  - La vue est propre à chaque application de la base de données
  - Elle est décrite en terme d'objets, propriétés et relations pour une utilisation particulière des informations du domaine.
  - Ce niveau est décrit par un ou plusieurs schéma externes.
- **Niveau conceptuel :** décrit la structure de la BD indépendamment de son implémentation physique (modélisation des entités du monde réel)
  - Exprimé dans un formalisme **Entité-Association (E-A)**
- **Niveau interne ou physique:**
  - Définit la réalisation **de la structure de données (type, taille,...)** selon le SGBD choisi (relationnel par exemple) et **d'objectifs d'optimisation**
  - Exprimé sous forme de fichiers internes du SGBD choisi



# Conception d'une BD

## Etapes de conception d'une BD

- **Etapes de Conception d'une BD**

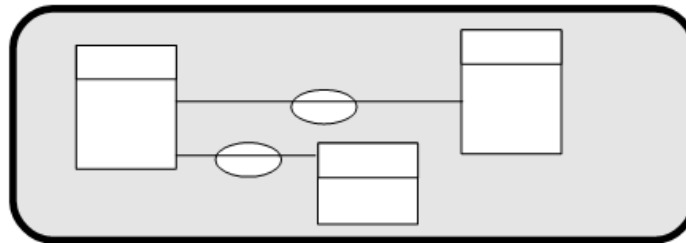


# Conception d'une BD

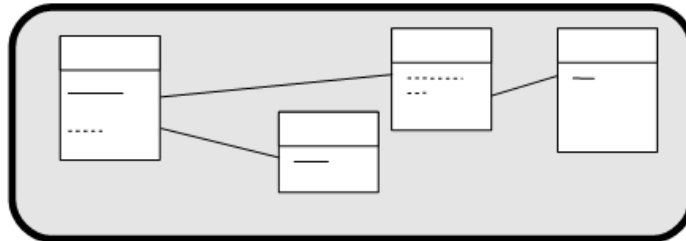
## Modèles de représentation des données

On peut distinguer trois modèles :

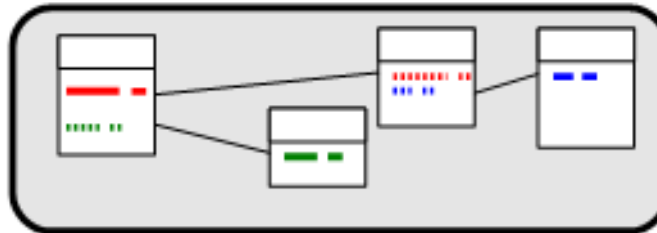
- **Modèle Conceptuel de données :**



- **Modèle Logique de données :**



- **Modèle Physique de données :** Représentation interne des données



# Rappels sur les Bases de données

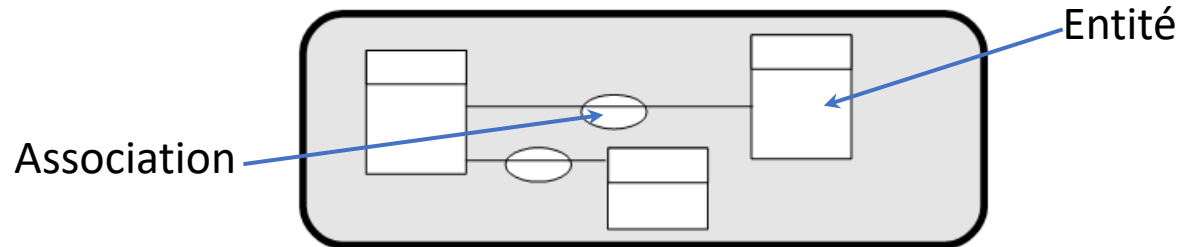
## Modèle Conceptuel de données

**Modèle Conceptuel de données** : correspond à une modélisation assez naturelle du monde réel.

Le modèle se compose de trois concepts élémentaires.

- **Entité**: objet du monde réel que l'on peut identifier et qui possède **un ensemble de propriétés**.
  - Une entité doit être dotée d'un identifiant
  - Sa valeur doit être différente pour chaque occurrence de l'entité.
- **Association**: lien logique entre plusieurs entités.
- **Attribut**: propriété d'une association ou d'une entité.

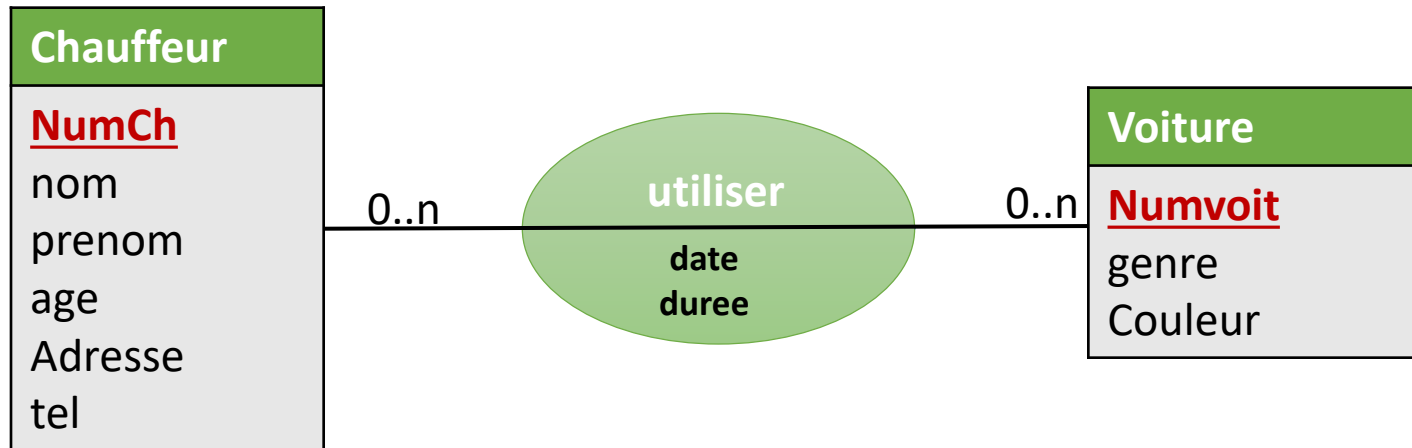
Une propriété est caractérisée par un nom et un type élémentaire (entier, chaîne de caractères, réel ...).



# Conception d'une BD

## Modèle Conceptuel de données

### Exemple



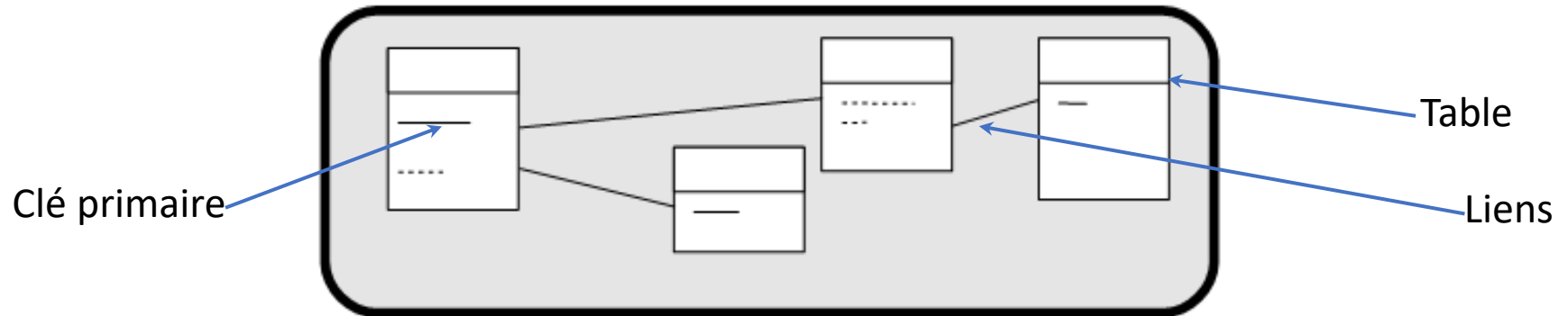
# Conception d'une BD

## Modèle Logique de données

### Définition (Modèle logique de données)

C'est un Intermédiaire entre le **modèle entité-association** et le **modèle physique** des données

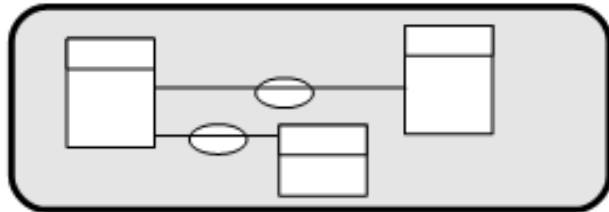
- Il est constitué de **tables relationnelles**
- Chaque relation est constituée de :
  - une **clé primaire** : **identifie de manière unique** chaque occurrence de la table
  - des attributs
  - éventuellement une ou plusieurs **clés étrangères**, clés primaires dans d'autres tables
- Les clés étrangères créent des liens entre les tables



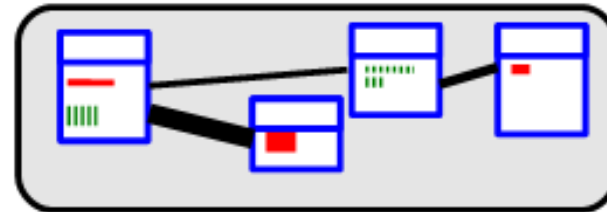
# Conception d'une BD

## Passage du Modèle Conceptuel au Modèle Logique

**Modèle entité association**



**Modèle logique des données**



Entité



Table

Identifiant



Clé primaire

Association 1 à 1



Clés dans la "table à 1"

Association 1 à plusieurs



Clé étrangère dans la "table à 1"

Association plusieurs à plusieurs



Table supplémentaire avec deux clés étrangères

Association *n*-aire



Table supplémentaire avec *n* clés étrangères

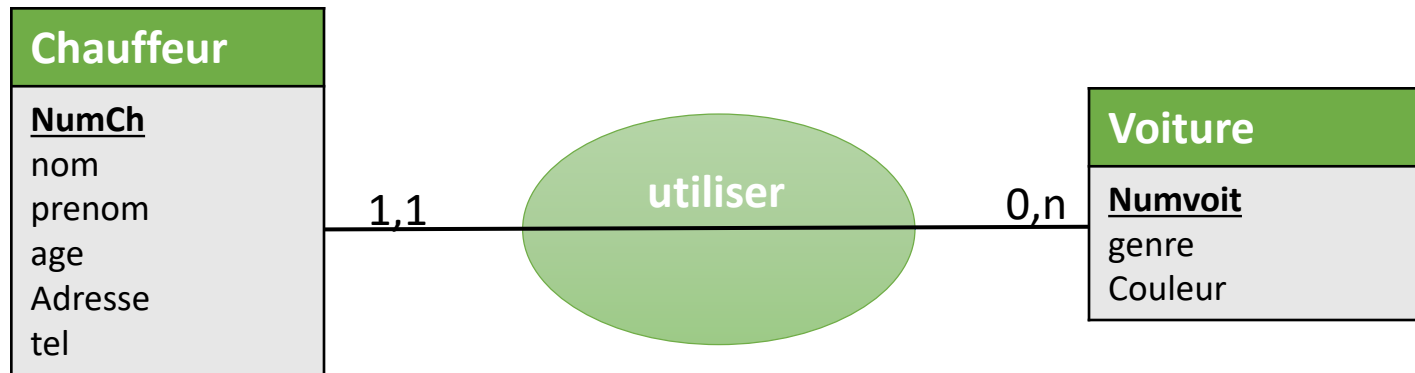


# Rappels sur les Bases de données

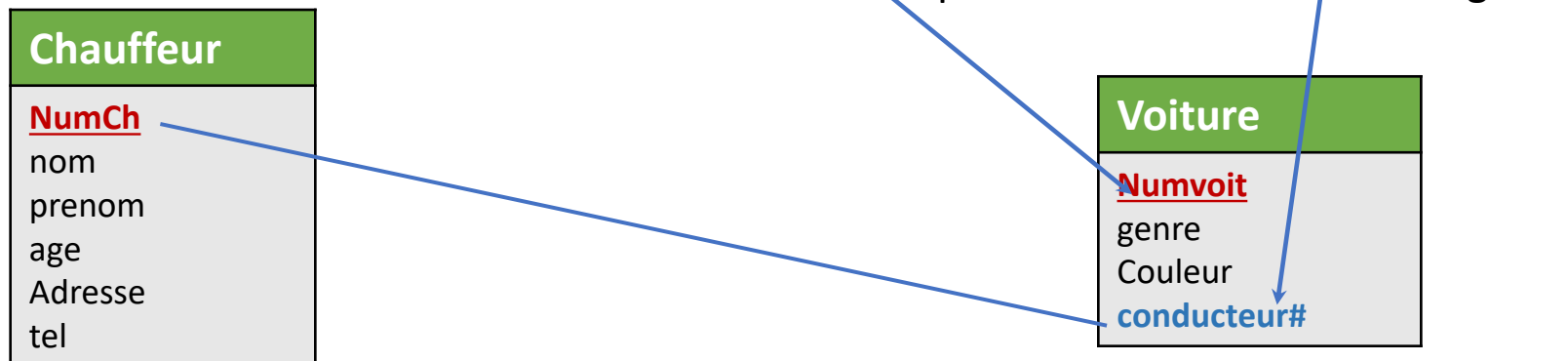
## Passage du Modèle Conceptuel au Modèle Logique

### Exemple 1 : Association 1 à plusieurs

Modèle Conceptuel de données



Modèle Logique de données

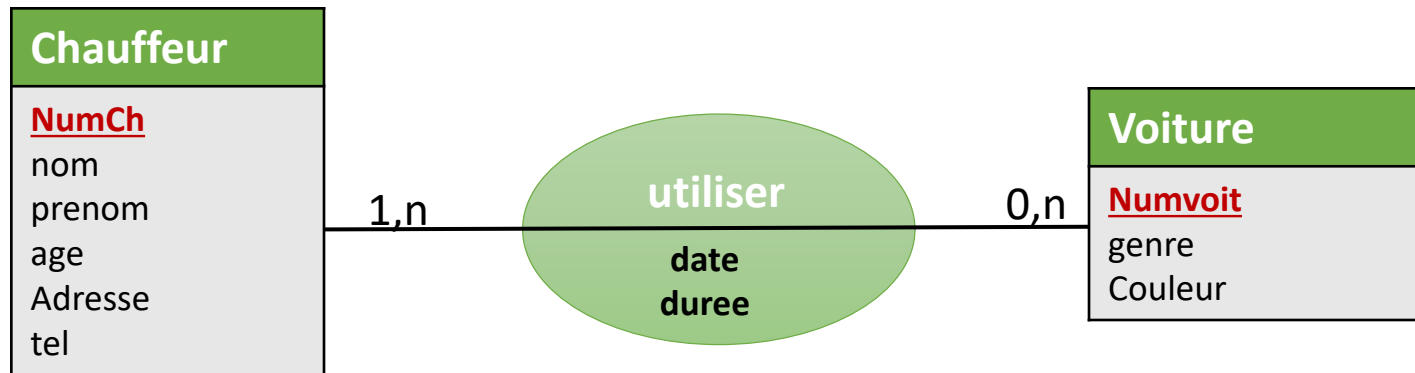


# Rappels sur les Bases de données

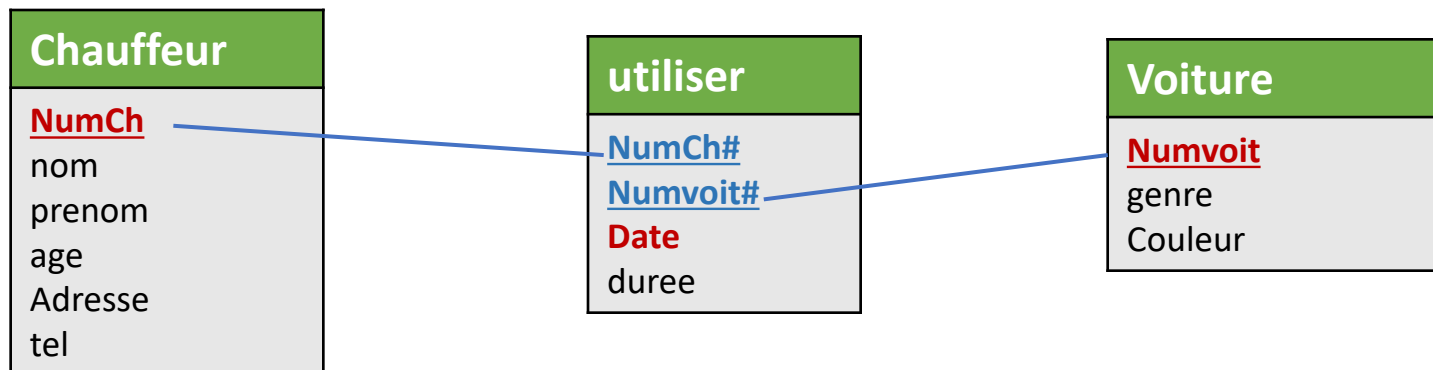
## Passage du Modèle Conceptuel au Modèle Logique

### Exemple 2 : Association plusieurs à plusieurs

Modèle Conceptuel de données



Modèle Logique de données

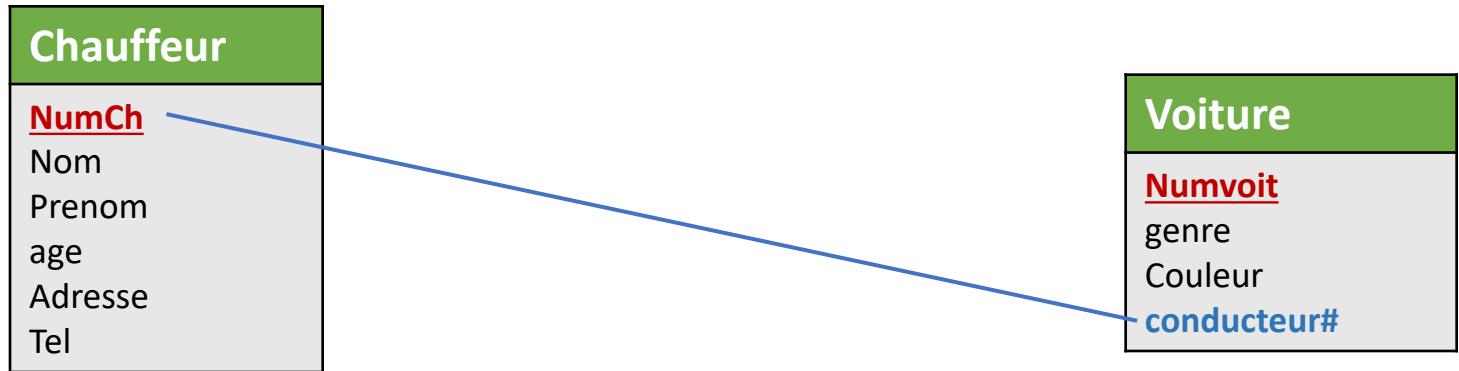


# Conception d'une BD

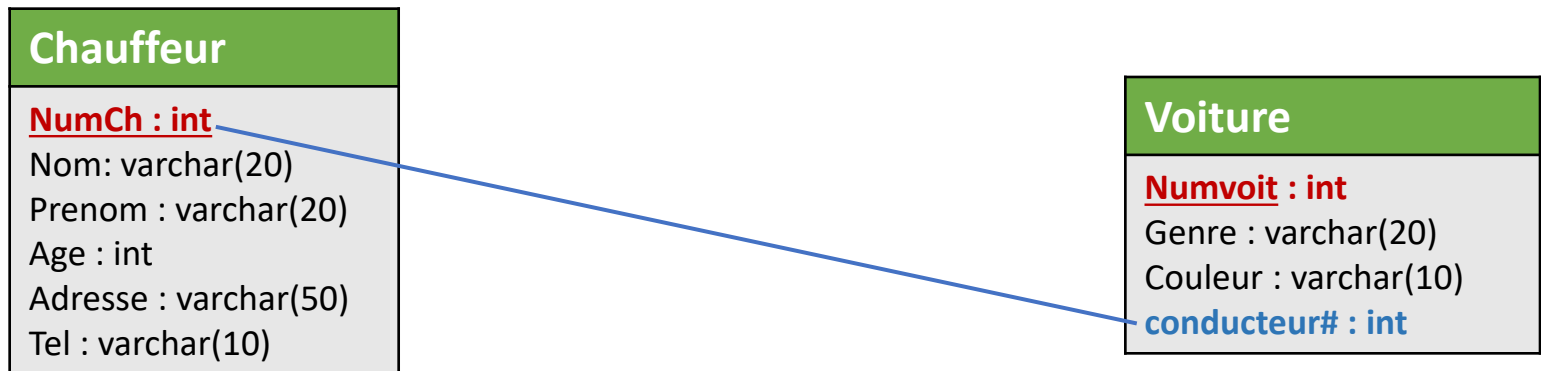
## Passage vers le Modèle Physique

### Exemple

#### Modèle Logique de données



#### Modèle Physique de données



# **Section 3 :**

## **Algèbre Relationnelle**

# Algèbre Relationnelle

## Définition (Algèbre relationnelle)

### Définition (Algèbre Relationnelle)

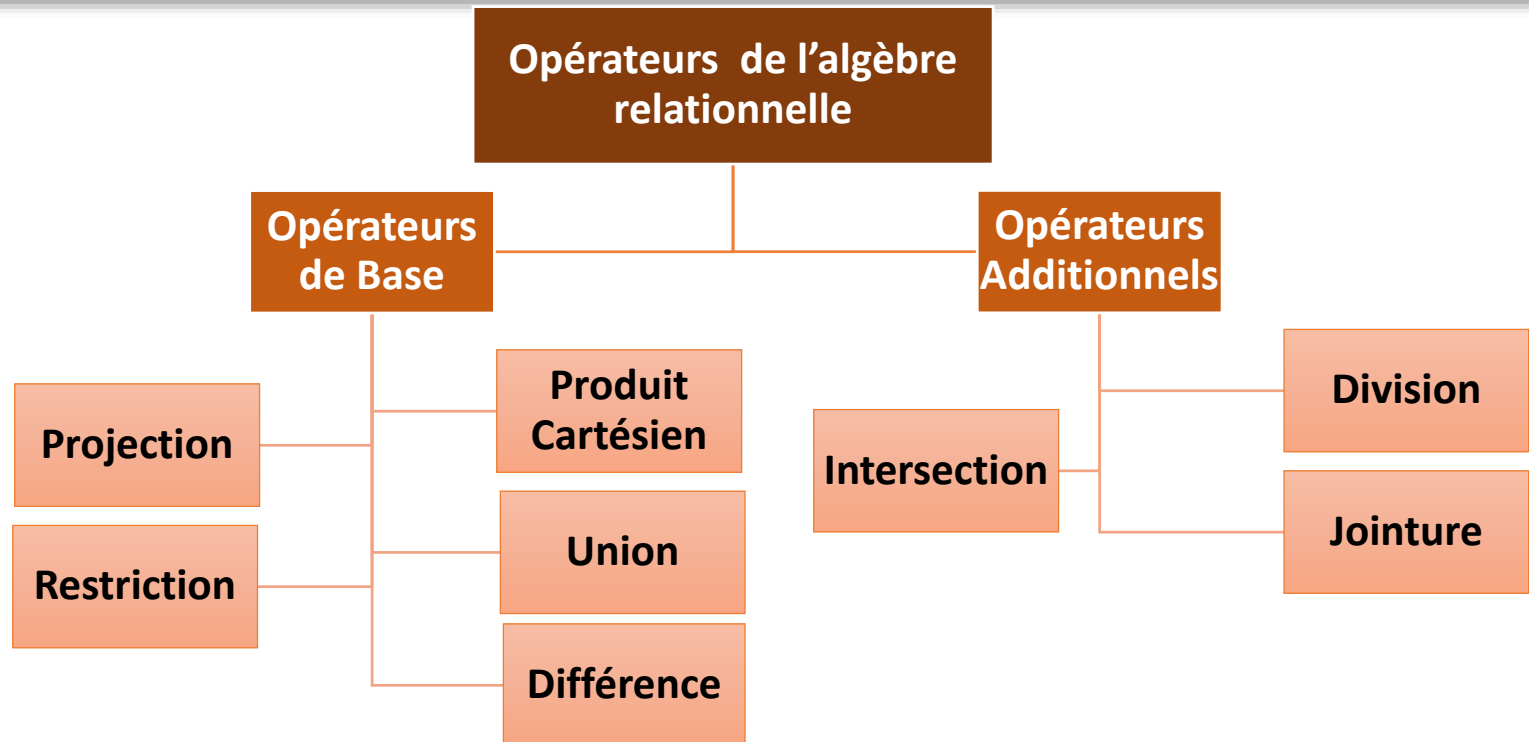
- L'algèbre relation est un langage de requête pour le modèle relationnel ayant des fondements mathématiques solides.
- L'algèbre relationnelle n'est pas visible au niveau de l'interface utilisateur de tout SGBDR,
- Mais, la plupart des SGBDR utilisent l'algèbre relationnelle pour la représentation interne des requêtes (pour l'optimisation et l'exécution des requêtes)

# Algèbre Relationnelle

## Définition (Algèbre relationnelle)

### Définition (Algèbre Relationnelle)

C'est ensemble d'**opérateurs** qui s'appliquent aux **relations** et permet d'exprimer des **requêtes**, posées à une représentation relationnelle, sous forme d'**expressions algébriques**.



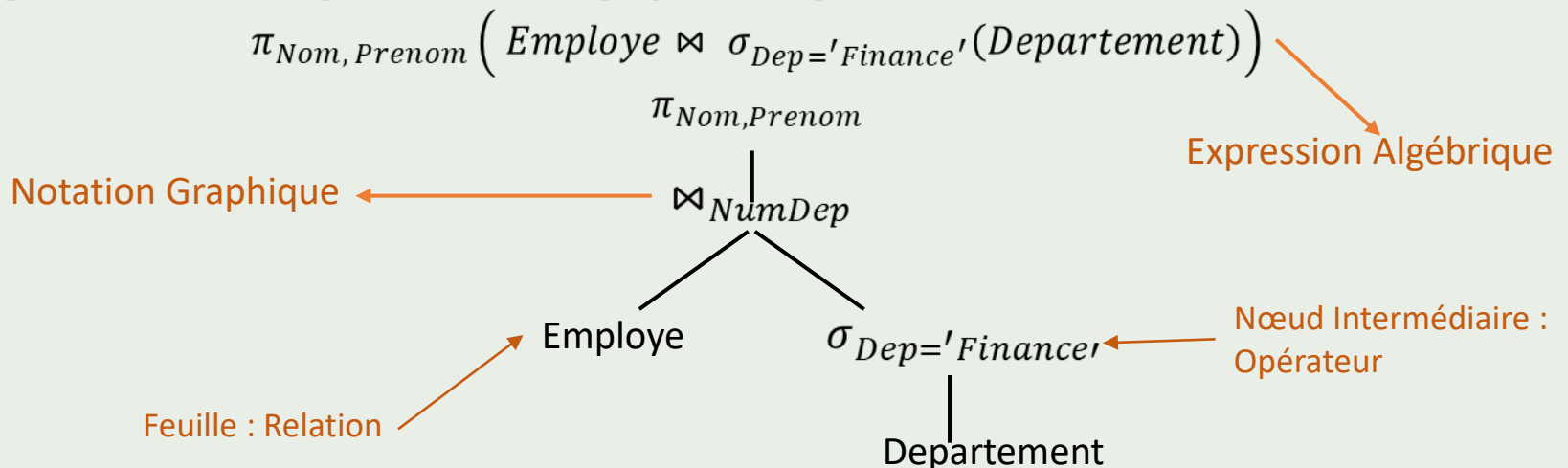
# Algèbre Relationnelle

## Requête dans l'Algèbre relationnelle

- Puisque le résultat de n'importe quelle opération de l'algèbre relationnelle est aussi une relation **R**, **R** peut être le paramètre d'entrée d'une autre opération
- Les opérateurs de l'algèbre relationnelle peuvent être **imbriqués**, le **résultat final** est toujours **une relation**.
- Une requête est un terme ou **une expression** dans l'algèbre relationnelle
- Une expression algébrique est représentée graphiquement par **un arbre d'opérateurs**

### Exemple de Requête

*Requête : les noms et prénoms des employés du département 'Finance'*



# Algèbre Relationnelle

## Requête dans l'Algèbre relationnelle

### Remarques Importantes(Algèbre Relationnelle/SQL)

Certaines différences existent entre les deux langages de requête l'algèbre relationnelle et SQL :

- Les valeurs **NULL** sont exclues dans la définition de l'algèbre relationnelle
- L'algèbre relationnelle considère **les relations** comme **des ensembles**,
- Donc, la duplication des tuples n'apparaît jamais dans les relations d'entrée et de sortie d'un opérateur
- Dans SQL, **les relations** sont des **multi-ensembles** et peuvent contenir des doublons
- L'élimination des doublons est explicite dans SQL(SELECT DISTINCT)



# Algèbre Relationnelle

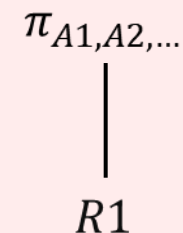
## Opérateurs de l'algèbre relationnelle

### Projection

- La projection est une **opération unaire** (portant sur une seule relation) notée  $\pi$
- La projection de **R1** sur une partie de ses attributs **{A1, A2, ...}** produit :
  - une relation **R2** dont le schéma est restreint aux attributs mentionnés en opérande,
  - comportant les mêmes tuples que **R1**, et dont les doublons sont éliminés.

### Syntaxe & Notation Graphique

$$R2 = \pi_{A1,A2,...}(R1)$$



# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Exemple : Projection

Soit  $R =$

A	B	C
1	4	7
2	5	8
3	6	9

$\pi_{A,C}(R) =$

A	C
1	7
2	8
3	9

# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Important

- $\pi_{A_1, \dots, A_K}(R)$  correspond à la requête SQL suivante:

```
SELECT DISTINCT  $A_1, \dots, A_K$ 
FROM           R
```

# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Remarque

- La projection élimine les doublons
- En général, les cardinalités des relations d'entrée et de sortie ne sont pas égales

### Exemple

Soit  $R =$

A	B
1	4
2	5
3	4

$\pi_B(R) =$

B
4
5

# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Restriction

- La restriction est une opération unaire (c'est à dire portant sur une seule relation).
- Etant donnée une condition **C**, la restriction de **R1** produit :
  - une relation **R2** de même schéma que **R1** et dont les tuples sont les tuples de **R1** vérifiant la condition **C**
  - comportant les mêmes tuples que **R1**, et dont les doublons sont éliminés.

### Syntaxe & Notation Graphique

$$R2 = \sigma_C(R1)$$



# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Important

- $\sigma_{\varphi}(R)$  correspond à la requête SQL suivante:

```
SELECT  *  
FROM    R  
WHERE    $\varphi$ 
```

# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Exemple : Restriction

Soit R =

SID	CAT	ENO	POINTS
101	H	1	10
101	H	2	8
101	M	1	12
102	H	1	9
102	H	2	9
102	M	1	10
103	H	1	5
103	M	1	7

$\sigma_{SID=101} =$

SID	CAT	ENO	POINTS
101	H	1	10
101	H	2	8
101	M	1	12

# Opérateurs de l'algèbre relationnelle

## Restriction

## Filtre certaines Lignes

## Filtre certaines colonnes



# Algèbre Relationnelle

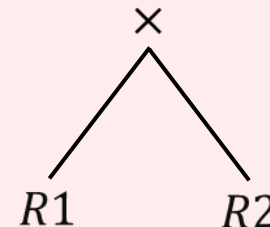
## Opérateurs de l'algèbre relationnelle

### Produit cartésien

- Le produit cartésien est une opération binaire (c'est à dire portant sur deux relations).
- Le produit de **R1** par **R2** produit
  - une relation **R3** ayant pour schéma la juxtaposition de ceux des relations **R1** et **R2**
  - et pour tuples l'ensemble des combinaisons possibles entre les tuples de **R1** et ceux de **R2**.

### Syntaxe & Notation Graphique

$$R3 = R1 \times R2$$



# Algèbre Relationnelle

## Opérateurs de l'algèbre relationnelle

### Exemple: produit Cartésien

Soit  $R =$

A	B
1	2
3	4

Soit  $R' =$

A	B
1	2
3	4

$R \times R' =$

A	B	C	D
1	2	6	7
1	2	8	9
3	4	6	7
3	4	8	9

# Algèbre Relationnelle

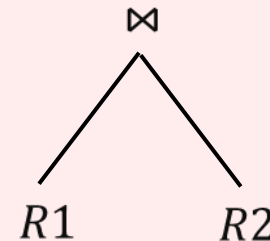
## Opérateurs de l'algèbre relationnelle

### Jointure

- La jointure est une opération binaire notée  $\bowtie$
- La jointure de **R1** et **R2**, étant
- Etant donné une condition **C** portant sur des attributs de **R1** et de **R2**, la jointure produit
  - une relation **R3** ayant pour schéma la juxtaposition de ceux des relations **R1** et **R2**
  - et pour tuples l'ensemble des tuples obtenus par concaténation des tuples de **R1** et de **R2**, et qui vérifient la condition **C**.

### Syntaxe & Notation Graphique

$$R3 = R1 \bowtie R2$$



### Remarque Importante

Les opérateurs ensemblistes sont des relations binaires portant sur des relations de **même schéma** (ayant les même attributs)

# Algèbre Relationnelle

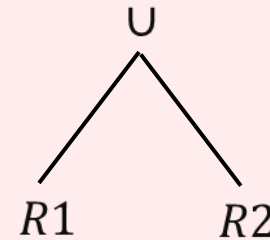
## Opérateurs ensemblistes

### Union

- L'union de deux relations **R1** et **R2** de même schéma produit
  - une relation **R3** de même schéma
  - constituée de l'ensemble des tuples appartenant à R1 **et/ou** à R2.

### Syntaxe & Notation Graphique

$$R3 = R1 \cup R2$$



# Algèbre Relationnelle

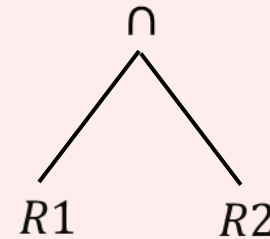
## Opérateurs ensemblistes

### Intersection

- L'intersection de deux relations **R1** et **R2** de même schéma produit
  - une relation **R3** de même schéma
  - constituée de l'ensemble des tuples appartenant à R1 **et** à R2.

### Syntaxe & Notation Graphique

$$R3 = R1 \cap R2$$



# Algèbre Relationnelle

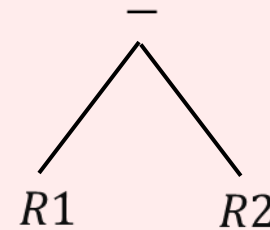
## Opérateurs ensemblistes

### Différence

- La différence entre deux relations **R1** et **R2** de même schéma produit
  - une relation **R3** de même schéma
  - constituée de l'ensemble des tuples de R1 **n'appartenant** pas à R2.

### Syntaxe & Notation Graphique

$$R3 = R1 - R2$$



# Algèbre Relationnelle

## Le renommage

### Renommage

- Le renommage est une opération qui permet de redéfinir ou renommer les noms des attributs d'une relation R.
  - Le renommage est noté  $\rho_{A_1, A_2, \dots}(R)$

### Exemple: renommage

Soit R =

A	B
1	2
3	4

$\rho_{Qte, Prix}(R) =$

Qte	Prix
1	2
3	4



# Algèbre Relationnelle

## Complexité des opérateurs

### **Selection :** $\sigma_{[condition]}R$

- Au plus : balayer la relation + tester la condition pour chaque tuple
- Complexité =  $\text{card}(R)$
- Taille du résultat :  $[0 : \text{card}(R)]$

### **Projection :** $\pi_{[A_1, A_2, \dots, A_k]}R$

- balayer la relation + élimination des doublons
- Complexité =  $\text{card}(R)$ .
- Taille du résultat :  $[1 : \text{card}(R)]$

### **Jointure(naturelle ou thêta) entre R et S**

- Balayer R et pour chaque tuple de R faire :
  - Balayer S et comparer chaque tuple de S avec celui de R
- Complexité =  $\text{card}(R) \times \text{card}(S)$
- Taille du résultat :  $[0 : \text{card}(R) \times \text{card}(S)]$