



# Bases de données - Modèle relationnel

## Introduction

SITE :<http://www.univ-orleans.fr/lifo/Members/Mirian.Halfeld/>



## Les bases de données - Bibliographie

---

- Ullman and Widom, *A first course in database systems*, Prentice-Hall International, 1997
- R. Ramakrishnan, *Database Management Systems*, McGraw-Hill, 1998
- Abraham Silberschatz Henry F. Korth S. Sudarshan, *Database System Concepts*, Fourth Edition, McGraw-Hill
- Greg Riccardi, *Principles of Database Systems with Internet and Java Applications*, Addison Wesley, 2001.

**LIVRES disponibles à la BU**



# Les bases de données - un aperçu

---

## Les sujets

1. Introduction au modèle relationnel
2. Langages de requêtes (algèbre relationnel et SQL)
3. Conception des bases de données (contraintes)
4. TPs Oracle

## Évaluation

- Devoirs sur table (3 dans le semestre)
- 1 projet en binôme ou trinôme
- Coefficient du projet légèrement inférieur



# Les bases de données

---

- Grande quantité de données stockées (dans un ordinateur).
- Normalement ces données sont inter-reliées et la BD est une entité cohérente logiquement et véhiculant une certaine sémantique.



# Les SGDB

- **SGBD**: Logiciel responsable pour la gestion de ces données.
- Ensemble de programmes qui permettent à des utilisateurs de créer et maintenir une base de données.
- SGBD commerciaux les plus connus sont [Oracle](#), [Sybase](#), [Ingres](#), [Informix](#) et [DB2](#)
- Capacités basics d'un SGBD:
  - Stockage d'un **grand volume** de données pendant **longtemps** et avec sécurité.
  - Accès **efficace**.
  - **Support d'un modèle de données**.
  - Permettre à l'utilisateur de créer des nouvelles bases de données ainsi que de spécifier leurs schémas (utilisation de DDL)
  - Permettre à l'utilisateur d'interroger et de modifier les données de la base (utilisation de DML)
  - Contrôler l'accès aux données par **plusieurs utilisateurs, en même temps**. L'action d'un utilisateur ne doit pas affecter un autre.



## Les SGDB - que souhaitons nous de plus?

- Assurer le respect des règles de cohérence définies sur les données; **vérifier les contraintes d'intégrité.**
- Rendre transparent le partage des données entre différents utilisateurs.
- Gérer les autorisations d'accès.
- Assurer la sécurité et la reprise après panne.
- Offrir des interfaces d'accès multiples.



# Modèle de données

- Abstraction mathématique selon laquelle l'utilisateur voit les données  
Exemples: relationnel, réseaux, hiérarchique, etc
- Possède deux parties:
  - Un langage qui permet la description des données.

```
CREATE TABLE STUDENT
( Num Integer,
  FirstName Char(100),
  LastName Char(100),
  BirthYear Integer)
```

- Un langage avec un ensemble d'opérations pour manipuler les données.

```
SELECT LastName
FROM STUDENT
WHERE BirthYear =1980
ORDER BY LastName
```

# Architecture logique d'un SGBD

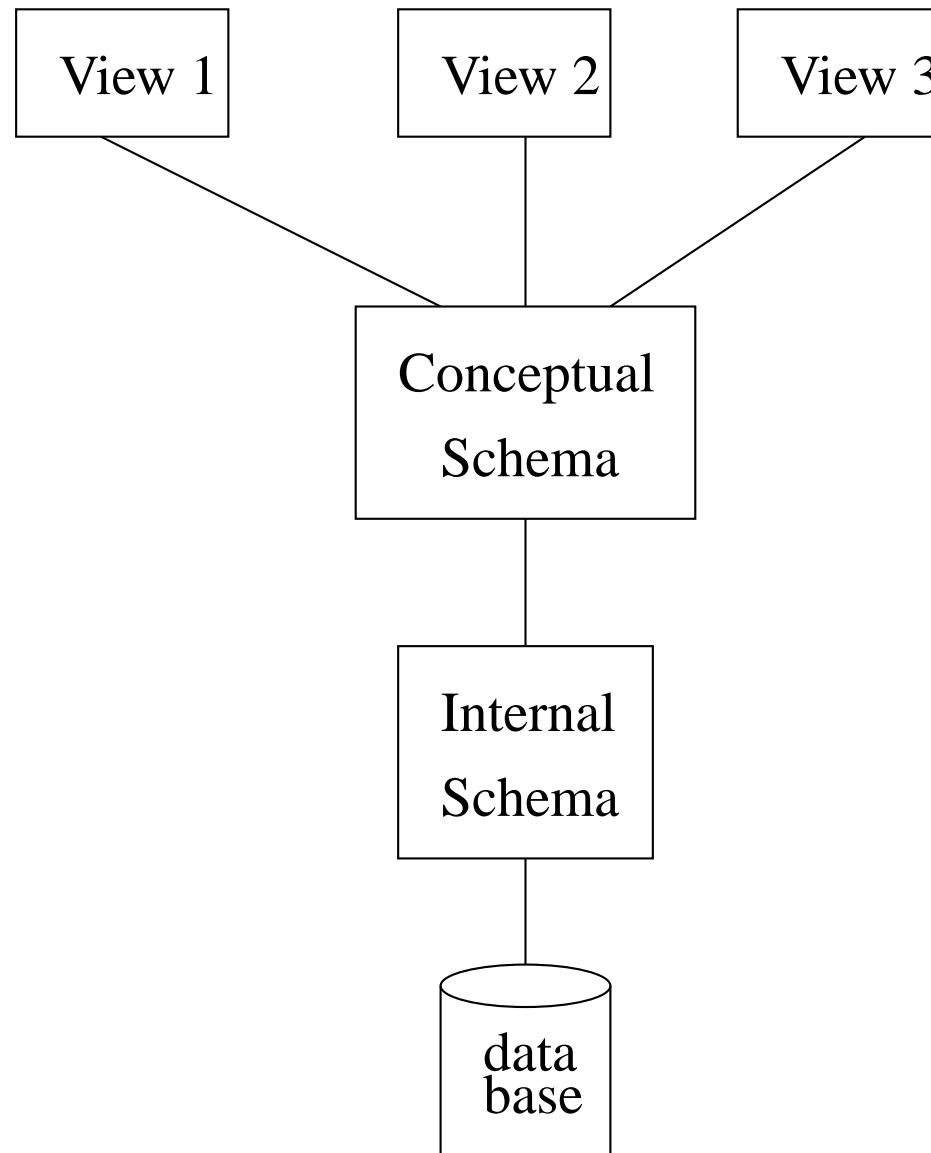
La plupart des SGBD suivent l'architecture standard Ansi/Sparc qui permet d'isoler les différents niveaux d'abstraction nécessaires pour un SGBD.

- **Niveau interne ou physique:** décrit le modèle de stockage des données et les fonctions d'accès.
- **Niveau conceptuel ou logique:** décrit la structure de la base de données globalement à tous les utilisateurs.
  - Le schéma conceptuel est produit par une analyse de l'application à modéliser et par intégration des différentes vues utilisateurs.
  - Ce schéma décrit la structure de la base indépendamment de son implantation.
- **Niveau externe:** correspond aux différentes vues des utilisateurs. Chaque schéma externe donne une vue sur le schéma conceptuel à une classe d'utilisateurs.

Le SGBD doit être capable de faire des transformations entre chaque niveau, de manière à transformer une requête exprimée en terme du niveau externe en requête du niveau conceptuel puis du niveau physique.



# Architecture logique d'un SGBD



# Le modèle relationnel

- Les données sont organisées en relations
  - Tables: relations
  - Colonnes: attributs
  - Lignes: n-uplets (ou tuples)

STUDENT	Num	FirstName	LastName	BirthYear
	2008120	Dumont	Marie	1980
	2008122	Dubois	Paul	1980
	2008125	Martin	Jean	1981
	...	...	...	

# Le modèle relationnel

- **Schéma d'une base de données relationnel**

- Ensemble de noms de tables
- Ensemble d'attributs pour chaque table

STUDENT [Num, FirstName, LastName, BirthYear]

INSCRIPTION[Num, CourseCode, Year]

- **Instance d'une bases de données**

- Ensemble de valeurs dans une table (ensemble de n-uplets)

$\{\langle 2008120, \textit{Dumont}, \textit{Marie}, 1980 \rangle, \langle 2008122, \textit{Dubois}, \textit{Paul}, 1980 \rangle\}$

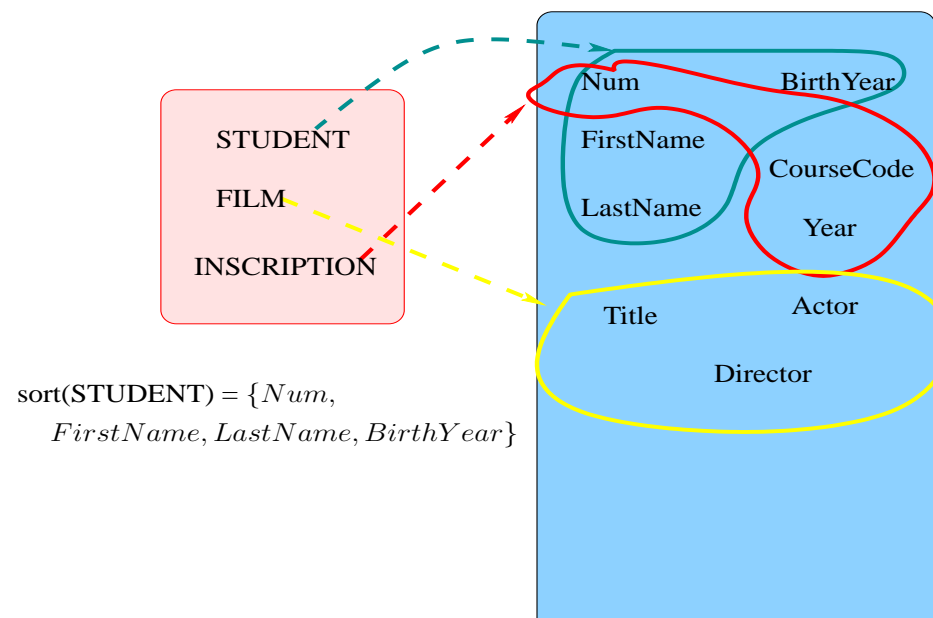
## Exemple d'une base relationnelle

Film	title	director	actor
	The Cameraman	Buster Keaton	Buster Keaton
	Rear Window	Hitchcock	James Stewart
	Rear Window	Hitchcock	Grace Kelly
	To Be or Not to Be	Lubitsch	Carole Lombard
	To Be or Not to Be	Lubitsch	Jack Benny

Schedule	theater	title	director
	le Champo	Buster Keaton	Buster Keaton
	le Champo	Rear Window	Hitchcock
	Action Christine	To Be or Not to Be	Lubitsch

# Modèle relationnel - Définitions formelles

- Nous considérons trois ensembles (infinis et dénombrables) **disjoints**:
  - **att**: attributs
  - **dom**: domaine
  - **relname**: noms de relations
- Un **schéma de relation** est un nom de relation (dans **relname**).
- Le **sort** d'une relation est une **fonction** qui associe à chaque **nom de relation** un **ensemble fini d'attributs** (un sous ensemble de **att**)





## Exemple

- **dom** = {The Cameraman, Hitchcock ,Grace Kelly, Lubitsch, 10, tata ...}
- **att** = {title, director, actor, ...}
- **rename** = {Film, Schedule, ...}
- Le *sort*  
 $sort(Film) = \{title, director, actor\}$   
 $sort(Schedule) = \{theater, title, director\}$



## Schéma de bases de données

- Un ensemble **fini non vide** **R** de noms de relations
- $\mathbf{R}[R_1[U_1], \dots, R_n[U_n]]$   
indique les schémas de relations de **R**.
- Schéma de la base de données **Cinéma**:  
**Cinéma**[*Film*[*title, director, actor*], *Schedule*[*theater, title, director*]]



# Points de vue étiqueté et point de vue non étiqueté

Nous pouvons nous placer dans différents contextes:

1. Nous connaissons les noms des attributs
2. Nous ne connaissons pas les noms des attributs, mais nous connaissons leur ordre



# Points de vue étiqueté et point de vue non étiqueté

## Approche étiqueté

title	year	length	inColor	studioName	producerC#
StarWars	1977	124	true	Fox	12345
Mighty Ducks	1991	104	true	Disney	67890
Wayne's World	1992	95	true	Paramount	99999

## Approche non étiqueté

1	2	3	4	5	6
StarWars	1977	124	true	Fox	12345
Mighty Ducks	1991	104	true	Disney	67890
Wayne's World	1992	95	true	Paramount	99999

## N-uplet: Point de vue étiqueté

- Les noms des attributs sont considérés
- Un n-uplet est une **fonction totale** qui associe à chaque attribut une valeur dans le domaine **dom**.
- Exemple d'un tuple  $u$ :

$u(\text{title}) = \text{To Be or Not to Be}$

$u(\text{director}) = \text{Lubitsch}$

$u(\text{actor}) = \text{Carole Lombard}$

Représentation d'un n-uplet:

$\langle \text{title: To Be or Not to Be, director: Lubitsch, actor: Carole Lombard} \rangle$

Film	title	director	actor
	To Be or Not to Be	Lubitsch	Carole Lombard

## N-uplet: Point de vue non étiqueté

- Seulement l'arité est considérée
- Un tuple est un **élément du produit cartésien** de  $\text{dom}^n$  ( $n \geq 0$ )

$u(1) = \text{To Be or Not to Be}$

$u(2) = \text{Lubitsch}$

$u(3) = \text{Carole Lombard}$

Représentation d'un n-uplet:

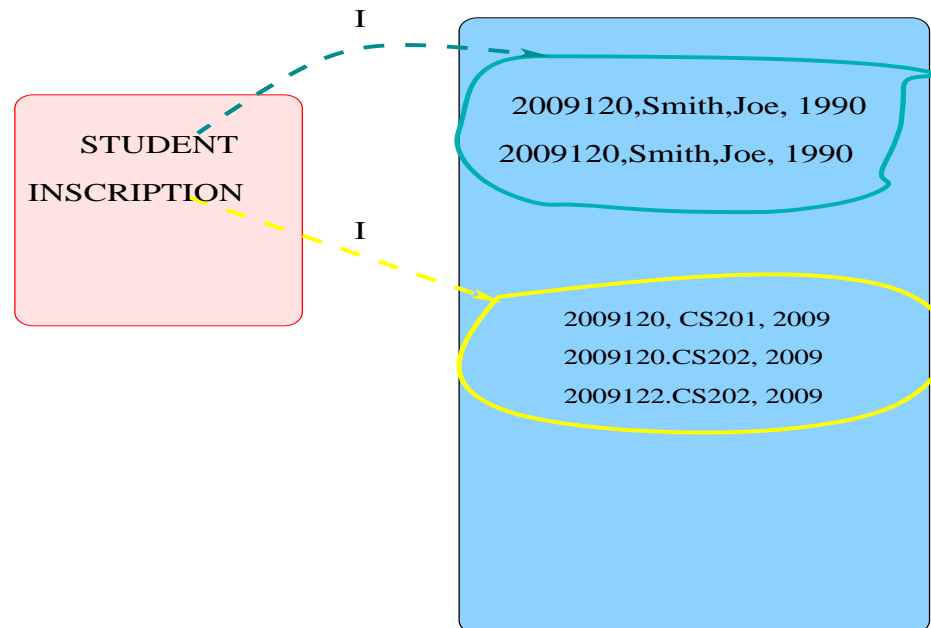
$\langle \text{To Be or Not to Be}, \text{Lubitsch}, \text{Carole Lombard} \rangle$

Film	1	2	3
	To Be or Not to Be	Lubitsch	Carole Lombard

Il existe une correspondance naturelle entre les deux points de vues

# Instance de relation

- **Instance de relation** d'un schéma de relation  $R[U]$  :
  - Ensemble fini de n-uplet  $I$  dont le *sort* est  $U$ .
  - Ensemble fini de n-uplet  $I$  dont l'arité est  $|U|$ .
- **Instance de base de données** dont le schéma est  $\mathbf{R}$ : application  $\mathbf{I}$  (dont le domaine est  $\mathbf{R}$ ) telle que  $\mathbf{I}(R)$  est une relation sur  $R$  pour tout  $R \in \mathbf{R}$ .



## Exemple

$$I(\text{Film}) = \{f_1, f_2\}$$

$$f_1(\text{title}) = \text{The Cameraman}$$

$$f_1(\text{director}) = \text{Buster Keaton}$$

$$f_1(\text{actor}) = \text{Buster Keaton}$$

$$f_2(\text{title}) = \text{To Be or Not To Be}$$

$$f_2(\text{director}) = \text{Lubitsch}$$

$$f_2(\text{actor}) = \text{Carole Lombard}$$

Film	title	director	actor
	The Cameraman	Buster Keaton	Buster Keaton
	To Be or Not to Be	Lubitsch	Carole Lombard

## Exemple

$I(\text{Film}) = \{\langle \text{TheCameraman}, \text{BusterKeaton}, \text{BusterKeaton} \rangle, \langle \text{To Be or Not To Be}, \text{Lubitsch}, \text{CaroleLombard} \rangle\}$

Film	1	2	3
	The Cameraman	Buster Keaton	Buster Keaton
	To Be or Not to Be	Lubitsch	Carole Lombard



# Étapes de la mise en place d'une BD relationnelle

1. Faire la conception de la base: analyse de l'application, liste de contraintes.
2. Implantation du schéma de la base avec les contraintes
3. Insertion des données dans la base
4. Consultation de la base via des langages de requêtes

**La cohérence de la base de données dépend de la qualité de sa conception!!!**

# Comment assurer la qualité d'une base?

## Qualité des schémas BD

- Quelques critères (informels) de qualité d'un schéma BD
  - Sémantique simple des attributs et de chaque schéma de relation.
  - Réduction des valeurs redondantes.
  - Réduction des valeurs nuls dans les relations.
  - Interdiction des n-uplets farfelus.
- Problème important de la redondance : Gaspillage de place et **anomalies des mise à jour**.



## Exemple

<b>videold</b>	<b>dateAcquired</b>	<b>title</b>	<b>genre</b>	<b>length</b>	<b>rating</b>
101	1/25/98	The Third-Nine Steps	mystery	120	R
90987	2/5/97	Elisabeth	drama	105	PG13
145	12/31/95	Lady and the Tramp	comics	93	PG
8034	4/5/98	Lady and the Tramp	comics	93	PG
90988	4/5/98	Elisabeth	drama	105	PG13
90989	3/25/86	Elisabeth	drama	105	PG13
543	5/12/95	The Third-Nine Steps	mystery	120	R
123	4/29/91	Annie Hall	comedy	120	R

## Exemple

- Redondance : À chaque fois qu'un film (titre) apparaît, les valeurs pour le *genre*, *length* et *rating* apparaissent aussi.
- Mélange de la sémantique des attributs : attributs concernant la vidéo avec des attributs d'un film.

- **Anomalie de mise à jour**

Que se passe t-il si la longueur (*length*) du film 90987 est mise à jour et passe de 105 à 107?

Incohérence ou besoin de modification de plusieurs n-uplet!

- **Anomalie d'insertion**

Que se passe t-il si avec l'insertion du n-uplet :

$\langle 102, 1/1/99, Elisabeth, drama, 110, PG13 \rangle$

Incohérence ou interdiction d'insertion!

- **Anomalie de suppression**

Que se passe t-il si avec la suppression du vidéo numéro 123?

Perte des informations sur le film *Annie Hall*

# Contraintes d'intégrité

- Un schéma de base de données n'impose pas certaines contraintes qu'existent dans le monde réel
- Exemple1:  
`Movie[title, year, director, actor]`  
⇒ Nous savons que chaque film est associé à un seul directeur. Cette information n'est pas visible à partir du schéma.
- Exemple2:  
`STUDENT[Num, LastName, FirstName, BirthYear]`  
`INSCRIPTION[Num, CourseCode, Year]`  
⇒ Nous voulons que l'inscription d'un étudiant dans un cours soit possible seulement si cet étudiant est enregistré comme un étudiant dans la base.

# Contraintes d'intégrité

Différents types de contraintes: comment les exprimer?

Les plus courantes (et utiles)

- **Dépendances fonctionnelle**

title, year → director

- **Dépendances d'inclusion**

INSCRIPTION(Num)  $\subseteq$  STUDENT (Num)

Les SGDB ne nous donnent pas les moyens d'implémenter DIRECTEMENT les dépendances fonctionnelles et d'inclusion. Mais ils nous permettent d'implémenter des **clés** et des **clés étrangères**.

Exemple1: **Pour que notre DF corresponde à la définition d'une clé dans notre exemple IL FAUT DÉCOMPOSER notre grande table!**



## Conclusion

- Les bases de données permettent la réalisation de diverses applications, **néanmoins, elles doivent rester cohérentes au fur et à mesure que ces données sont mises à jour.**
- La cohérence d'une base dépend de la qualité de son schéma et de la mise en place des contraintes.
- **La vérification des contraintes dans une base de données est efficace si elle profite des mécanismes (déclaratifs) que les SGBD laissent à notre disposition.**