

Bases de Données

Cours de SRC I

Mathieu MANGEOT

mathieu.mangeot@univ-savoie.fr

Objectifs du cours

- Analyser les besoins et modéliser les données d'un système d'information
- Mettre en œuvre des bases de données
 - en utilisant un Système de Gestion de BD

Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Introduction

- Évolution
- Définitions
- Modèles de données

Les bases de données

- Notion intuitive
- Rôle essentiel dans le domaine informatique
 - utilisation très répandue (banques, entreprises...)
- Demande croissante et exigeante
 - Nouveaux besoins (multimédia, volume,...)
 - Domaine en évolution
- Culture informatique
 - Réseaux, programmation, Bases de données, ...

Les ancêtres

- *Fichiers de données:*
 - séquence d'enregistrements.
 - accès séquentiel, indexé
- *Fichiers de traitements*
 - ensemble d'instructions pour manipuler les données des fichiers (Assembleur, Cobol).
- *Fichiers données + fichiers traitements*
 - fortement liés, non indépendant

Données et traitements liés

- *Contenu des fichiers de traitement :*
 - accès au fichier de données (chemin, mode d'ouverture, mode de parcours)
 - description des données du fichier dans des structures
 - parcours du fichier de données pour manipulation/MAJ des données
 - mise en forme des résultats
 - ...

Pas très pratique ...

- *Inconvénients :*
 - redondance des informations : MAJ difficile
 - problèmes d'incohérence, de fiabilité, de sécurité
 - manque de structuration des données
- *Si changement de la structure d'un fichier de données*
 - tous les programmes qui l'utilisent doivent être modifiés !
 - temps perdu en maintenance !



Évolution ...

- Données et traitements indépendants
- Fichiers de données regroupés en une seule entité :

**Les bases de données
sont nées !**

Définitions

- Valeur : 12
- Donnée : 12 de moyenne
- Information : Agnès a 12 de moyenne
- Connaissance : un étudiant refait l'année s'il a une moyenne <10.

Définition : BD

- Base de données (BD)
 - ensemble structuré de données
 - enregistré sur des mémoires secondaires
 - créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs ou de programmes

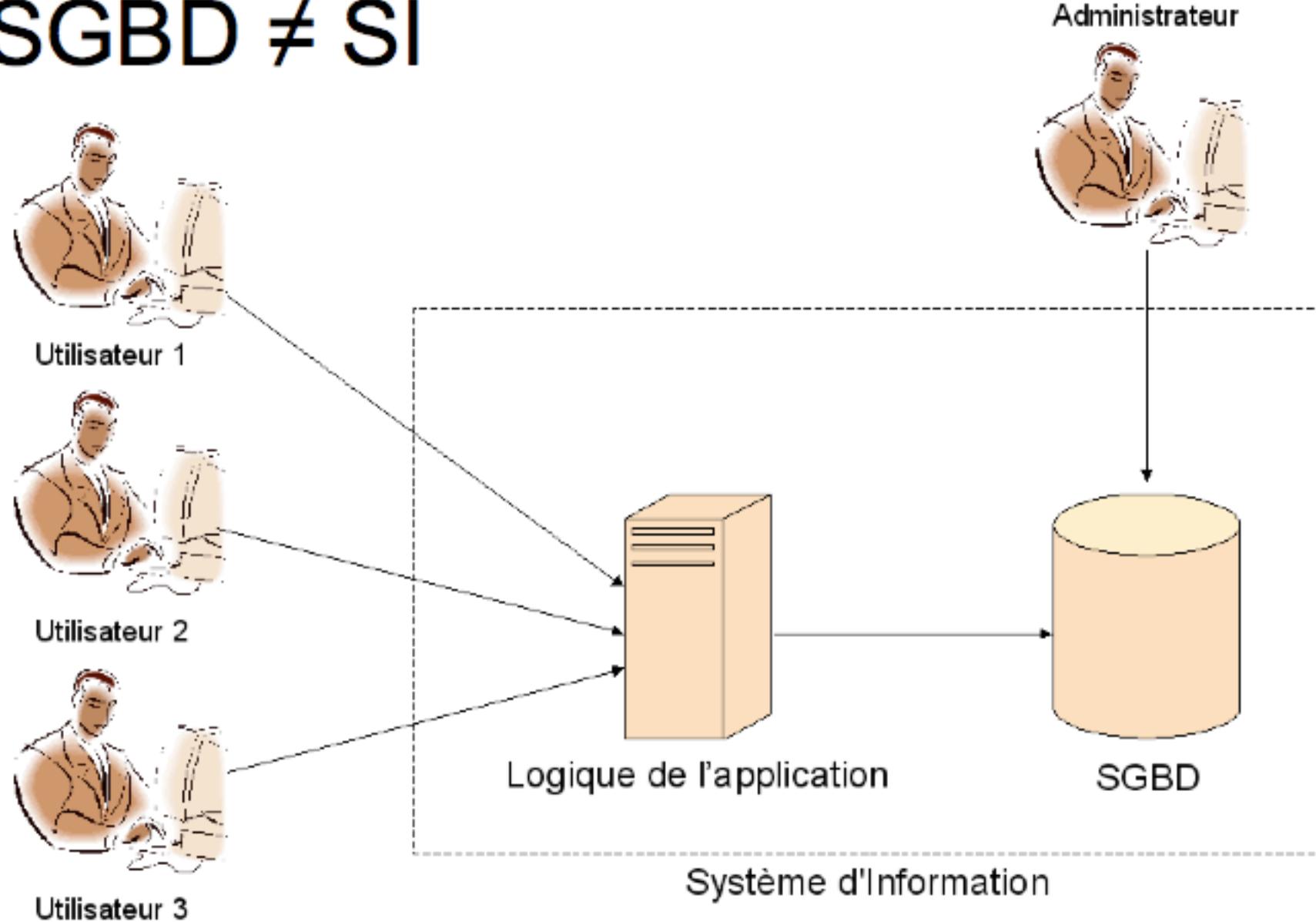
Définition : SGBD

- Un Système de Gestion de Bases de Données (SGBD) est un ensemble de services (applications logicielles) permettant de :
 - Définir la structure d'une BD
 - Rechercher des données dans une BD
 - Insérer, modifier, supprimer des données dans une BD
 - Assurer la cohérence et la confidentialité des données

Définition : SI

- Système d'information (SI) :
 - permet de gérer les informations nécessaires au bon fonctionnement d'une entreprise.
 - Les BD sont au cœur d'un SI.

SGBD ≠ SI



Attention !

- Base de données \neq Fichiers
- Une BD est composée physiquement de fichiers
- Une BD est gérée par un SGBD qui s'appuie sur un système de gestion de fichiers pour gérer les données

SGBD : niveaux d'abstraction

Norme ANSI/SPARC (1975)

- ***Niveau externe*** *Utilisateur quelconque*
 - « vues » des utilisateurs selon leurs droits
- ***Niveau conceptuel*** *Concepteur, programmeur*
 - description/manipulation de la structure de la base
- ***Niveau interne (physique)*** *Administrateur*
 - organisation physique des fichiers, gestion des accès concurrents, sauvegardes, droits ...

Données “structurées”

- Organisées par thèmes et selon certaines règles.

Non organisées

Magalie: Fille.

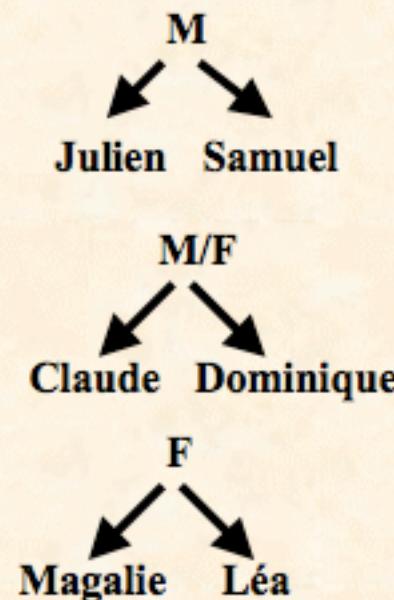
Claude, Dominique: M/F.

Garçons: Julien, Samuel.

F: Léa.

...

Organisées



Prénom	Sexe
Magalie	F
Samuel	M
Léa	F
Claude	M/F
Julien	M
Dominique	M/F
...	...

“Modèle” des données

- Ensemble de concepts pour décrire la *structure* d'une base de données
 - données et types des données
 - liens entre ces données
 - opérateurs, éventuellement
- Sorte de « langage »

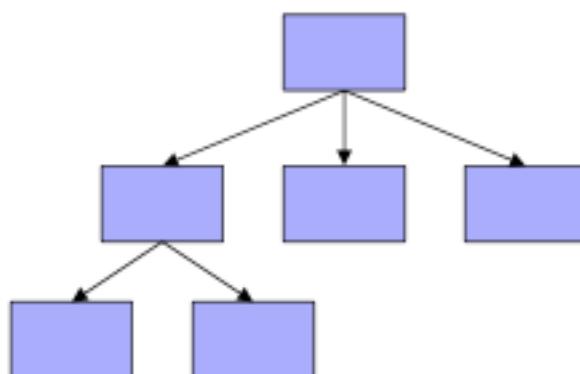
Niveaux du modèle

- ***Modèle conceptuel des données***
 - description graphique des données et des règles exprimant les liens entre les données
- ***Modèle logique des données***
 - description de la structure de la BD
- ***Modèle physique des données***
 - données elle mêmes, contenues dans leur structure physique

Modèle hiérarchique

- 1960-1975 : SGBD hiérarchique ou réseau

- ▶ les données sont classées hiérarchiquement
- ▶ arborescence descendante ou réseau
- ▶ utilise des pointeurs entre les différents enregistrements
- ▶ premier modèle de SGBD



Modèle relationnel

- 1975-... : SGBD relationnel (SGBDR)
 - données enregistrées dans des tableaux 2D
 - ▶ lignes + colonnes
 - fondé sur la théorie des ensembles

Table Personne

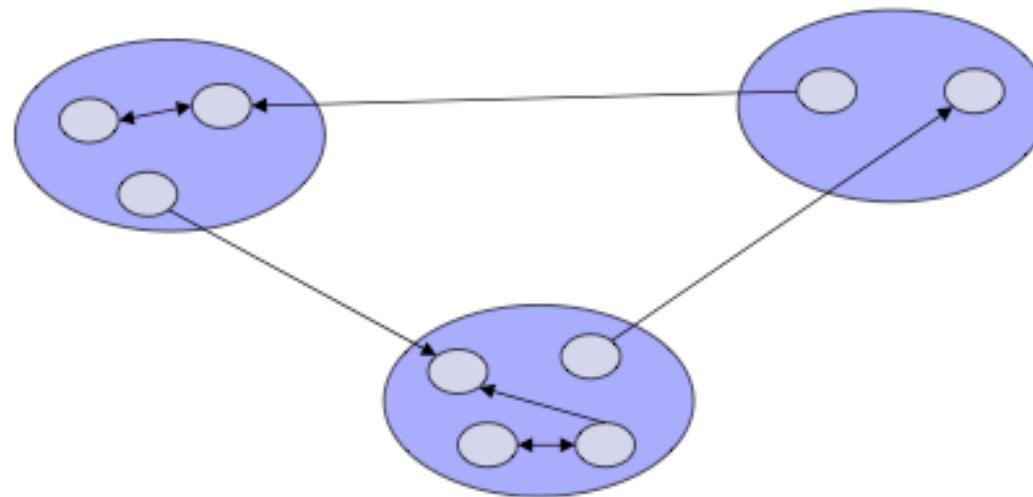
ID_Personne	Nom	Prénom
1	Collet	Jean
2	Rolland	Christine
3	Pierre	Bernard

Table Fonction

ID_Fonction	Libellé
1	Président
2	Ingénieur
3	Ouvrier

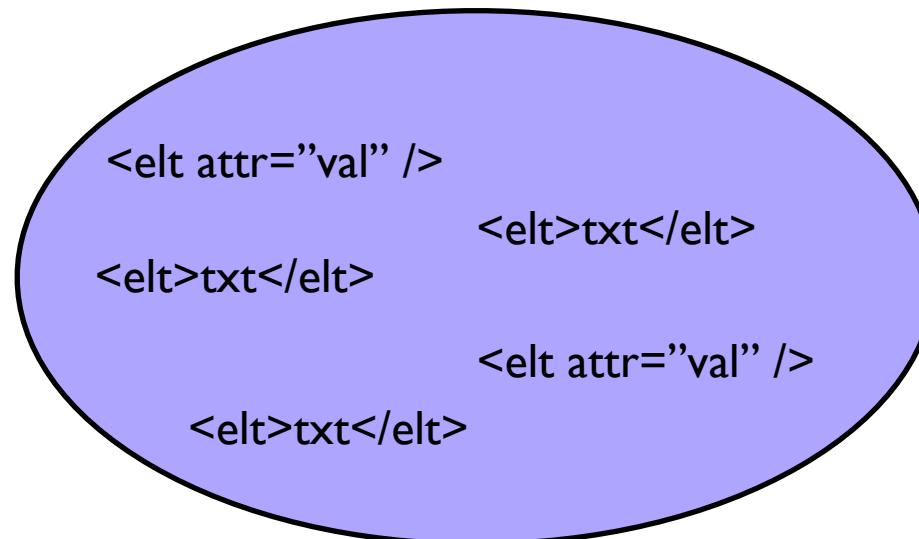
Modèle à objets

- 1990... : SGBD Objet (SGBDO)
 - données stockées sous forme d'objets
= structures appelées classes présentant des données membres.
 - Les champs sont des instances de ces classes



Modèle XML

- 1999... : SGBD XML natif
 - Stockage de documents XML
 - Langages de requêtes adaptés :
 - ▶ XPath, XQuery, SQL/XML,



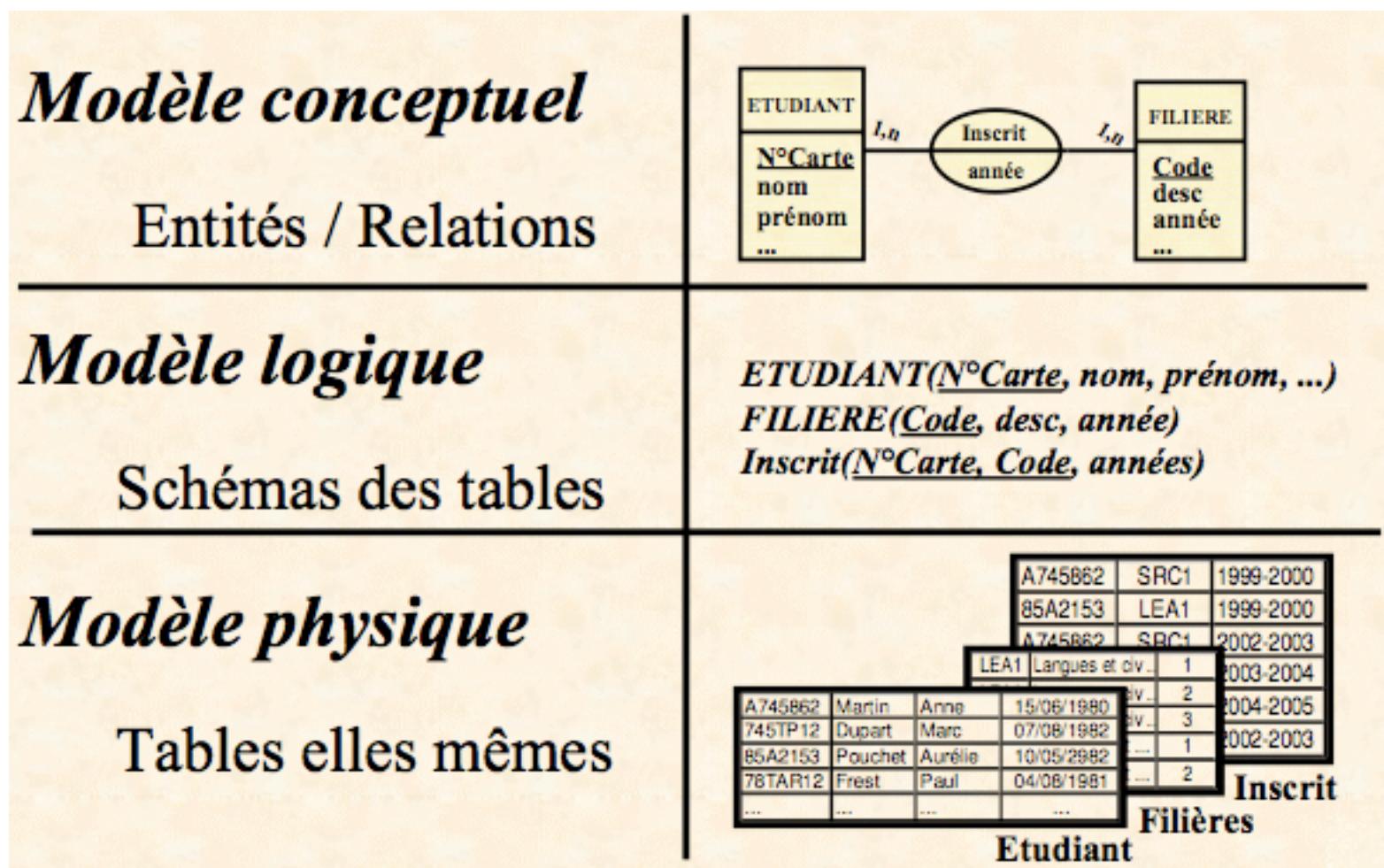
Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Modèle relationnel

- Définition du modèle
- Données et domaines
- Tables et enregistrements
- Produit cartésien de domaines
- Schémas de table
- Identifiants / clés primaires

Modèle relationnel des données



Définir un modèle relationnel

- Répertorier les données intéressantes
- Déterminer les règles exprimant les liens entre les données
- Regrouper les données par « thème »
- En déduire les schémas des tables de la base de données en *respectant au mieux les règles.*

Possibilité de s'appuyer sur un modèle graphique

Organisation des données

- Les **données** sont organisées sous forme de tableaux de valeurs :
 - les **tables**
- Chaque **table** représente une **relation** au sens mathématique du terme dans la théorie des ensembles.

Attributs de relations

- Un **attribut** (champ) est un nom donné à une colonne d'une **relation** (table)
- Il prend ses valeurs dans un **domaine**

Enregistrements {

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant

attribut 1 attribut 2

Domaines de valeurs

- Ensembles finis ou infinis de valeurs.
 - Représentés par une liste d'éléments ou bien une condition nécessaire et suffisante d'appartenance :
 - **ENTIER, RÉEL, CHAÎNE DE CARACTÈRES**
 - le domaine des doigts de la main :
 - ▶ {pouce, index, majeur, annulaire, auriculaire}

Tables et enregistrements

- Une table est un ensemble d'enregistrements

Etudiant

T	N°Carte	Nom	Prénom	Date-naiss
A	78TAR12	Frest	Paul	04/08/1981
B	745TP12	Dupart	Marc	07/08/1982
C	85A2153	Pouchet	Aurélie	10/05/2982
D	78TAR12	Frest	Paul	04/08/1981
E

Nom de la table

Attributs de la table

Enregistrement

Valeur d'attribut

- Un enregistrement est un ensemble de valeurs pour les attributs de la table



Pas deux enregistrements identiques !

Exemple de BD

Table Personne

ID Personne	Prénom	Nom	Fonction
1	Oualid	Khayati	1
2	Christine	Rolland	3
3	Christophe	Courtin	2

Table Planning

Salle	Date	Horaire	Intervenant	Formation	Cours
153	30/11/2005	8:30-10	1	1	3
69	15/1/2006	10:15-11:45	1	2	1

Table Fonction

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant

Table Cours

ID Cours	Libellé
1	Base de données
2	Algorithmique
3	Systèmes d'exploitation

Table Formation

ID Formation	Libellé
1	SRC1
2	Licence Pro
3	SRC2

Produit cartésien

- Le produit cartésien d'un ensemble de domaines D_i , noté $D_1 \times D_2 \times D_3 \times \dots \times D_n$ est l'ensemble des enregistrements (appelés aussi tuples) $\langle V_1, V_2, \dots, V_n \rangle$ tels que V_i appartient à D_i
- Jour = $\{j \in \mathbb{N} / 1 \leq j \leq 31\}$
- Mois = $\{m \in \mathbb{N} / 1 \leq m \leq 12\}$
- Année = $\{a \in \mathbb{N} / 1900 \leq a \leq 3000\}$
- Date = Jour \times Mois \times Année

Exemple de produit cartésien

Table Fonction

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant

Table Cours

ID Cours	Libellé
1	Base de données
2	Algorithmique
3	Systèmes d'exploitation

Table Fonction x Table Cours

		ID Cours	Libellé
1	Enseignant	1	Base de données
1	Enseignant	2	Algorithmique
1	Enseignant	3	Systèmes d'exploitation
2	Chef département	1	Base de données
2	Chef département	2	Algorithmique
2	Chef département	3	Systèmes d'exploitation
3	Étudiant	1	Base de données
3	Étudiant	2	Algorithmique
3	Étudiant	3	Systèmes d'exploitation

Quelles données ?

- *Définir ces données et leur type (format)*
 - en fonction des informations concernant le domaine à modéliser.
- *Distinguer données élémentaires et données calculées*
 - Les données calculées peuvent être déduites des données élémentaires.
 - Les données calculées apparaissent rarement dans le modèle final.

Types de données

Entier, réel, caractère, date, heure, ...

Donnée	Type	Taille	Elémentaire/calculée?
N°Carte	Caractère	8	E
Nom	Caractère	20	E
Prénom	Caractère	20	E
Date-naiss	Date	8	E
Age	Entier	3	C
Nb-redoublts	Entier	2	C
...	

Eviter les données calculées dans le modèle !

Schéma de la table

C'est aussi un/des attribut(s)

Nom de la table (Identifiant, [Liste d'attributs])

Exemple: *Etudiant(N°Carte, Nom, Prénom, Date-naiss)*

Table physique
correspondant à
ce schéma

T	<i>N°Carte</i>	<i>Nom</i>	<i>Prénom</i>	<i>Date-naiss</i>
A	78TAR12	Frest	Paul	04/08/1981
B	745TP12	Dupart	Marc	07/08/1982
L	85A2153	Pouchet	Aurélie	10/05/2982
E	78TAR12	Frest	Paul	04/08/1981

Qu'est ce qu'un identifiant et à quoi ça sert?

Identifiant / Clé primaire

- Dans une table, il ne *doit pas y avoir* deux enregistrements identiques.
- L'identifiant/Clé primaire garantit cette *unicité*
- La clé primaire traduit également une partie des *règles* exprimant les liens entre les données

Identifiant / Clé primaire

- **Définition formelle:** *un identifiant est un sous-ensemble d'attributs qui permet de caractériser tout enregistrement d'une table de façon unique.*



En vue d'optimisation de place, ce sous ensemble doit être ‘ minimal ’.

Importance de l'identifiant

Contraintes/Règles

« un étudiant ne peut pas redoubler, mais peut être inscrit dans deux filières différentes la même année»



Etudiant	Etudes	Année
A745862	SRC1	1999-2000
85A2153	LEA1	1999-2000
A745862	SRC1	2002-2003
A745862	SRC2	2003-2004
A745862	SRC2	2004-2005
78TAR12	LEA1	2002-2003
...

Inscrit(Etudiant,Etudes,Année)

Importance de l'identifiant

Contraintes/Règles

« un étudiant peut redoubler, mais ne peut pas être inscrit dans deux filières différentes la même année»



Etudiant	Etudes	Année
A745862	LEA1	2002-2003
85A2153	LEA1	1999-2000
A745862	SRC1	2002-2003
A745862	SRC2	2003-2004
A745862	SRC2	2004-2005
78TAR12	LEA1	2002-2003
...

Inscrit(Etudiant, Année, Etudes)

Importance de l'identifiant

Contraintes/Règles

« un étudiant peut redoubler, être inscrit dans plusieurs filières différentes la même année »

« un étudiant ne peut pas être inscrit plusieurs fois dans la même filière la même année



Etudiant	Etudes	Année
A745862	SRC1	2002-2003
85A2153	LEA1	1999-2000
A745862	SRC1	2002-2003
A745862	SRC2	2003-2004
A745862	SRC2	2004-2005
78TAR12	LEA1	2002-2003
...

Inscrit(Etudiant, Année, Etudes)

Importance de l'identifiant

Contraintes/Règles

«On ne gère pas l'historique des inscriptions.
« Un étudiant ne peut pas être inscrit dans plusieurs filières différentes»



N°Carte	Nom	Prénom	Date-naiss	Etudes
78TAR12	Frest	Paul	04/08/1981	SRC1
745TP12	Dupart	Marc	07/08/1982	LEA1
85A2153	Pouchet	Aurélie	10/05/2982	SRC1
A745862	Martin	Anne	15/06/1980	SRC2



Etudiant	Etudes
78TAR12	SRC1
745TP12	LEA1
85A2153	SRC1
A745962	SRC2

Etudiant(N°Carte,Nom,Prénom,Date-naiss, Etudes)

Clé étrangère

Définition :

On appelle clé étrangère un attribut d'une table qui fait référence à un attribut clé primaire d'une autre table.



N°Carte	Nom	Prénom	Date-naiss	Etudes
78TAR12	Frest	Paul	04/08/1981	SRC1
745TP12	Dupart	Marc	07/08/1982	LEA1
85A2153	Pouchet	Aurélie	10/05/2982	SRC1
A745862	Martin	Anne	15/06/1980	SRC2



Code	Description	Année
LEA1	Langues et civ ..	1
LEA2	Langues et civ ..	2
LEA3	Langues et civ ..	3
SRC1	Systèmes et ...	1
SRC2	Systèmes et ...	2
...

Etudiant(N°Carte,Nom,Prénom,Date-naiss, #Etudes)

Filière(Code,Description, Année)

Résumé

- **Table** : ensemble d'enregistrements.
- **Enregistrement** : ensemble de valeurs pour chaque attribut de la table.
- **Schéma de table**: description d'une table :
Nom(identifiant, liste d'attributs)
- **Identifiant** : (sous)-ensemble *minimal* d'attributs caractérisant chaque enregistrement de façon *unique*.

Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Rappel : modèle physique

Table Personne

ID Personne	Prénom	Nom	Fonction
1	Oualid	Khayati	1
2	Christine	Rolland	3
3	Christophe	Courtin	2

Table Fonction

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant

Table Planning

Salle	Date	Horaire	Intervenant	Formation	Cours
153	30/11/2005	8:30-10	1	1	3
69	15/1/2006	10:15-11:45	1	2	1

Table Cours

ID Cours	Libellé
1	Base de données
2	Algorithmique
3	Systèmes d'exploitation

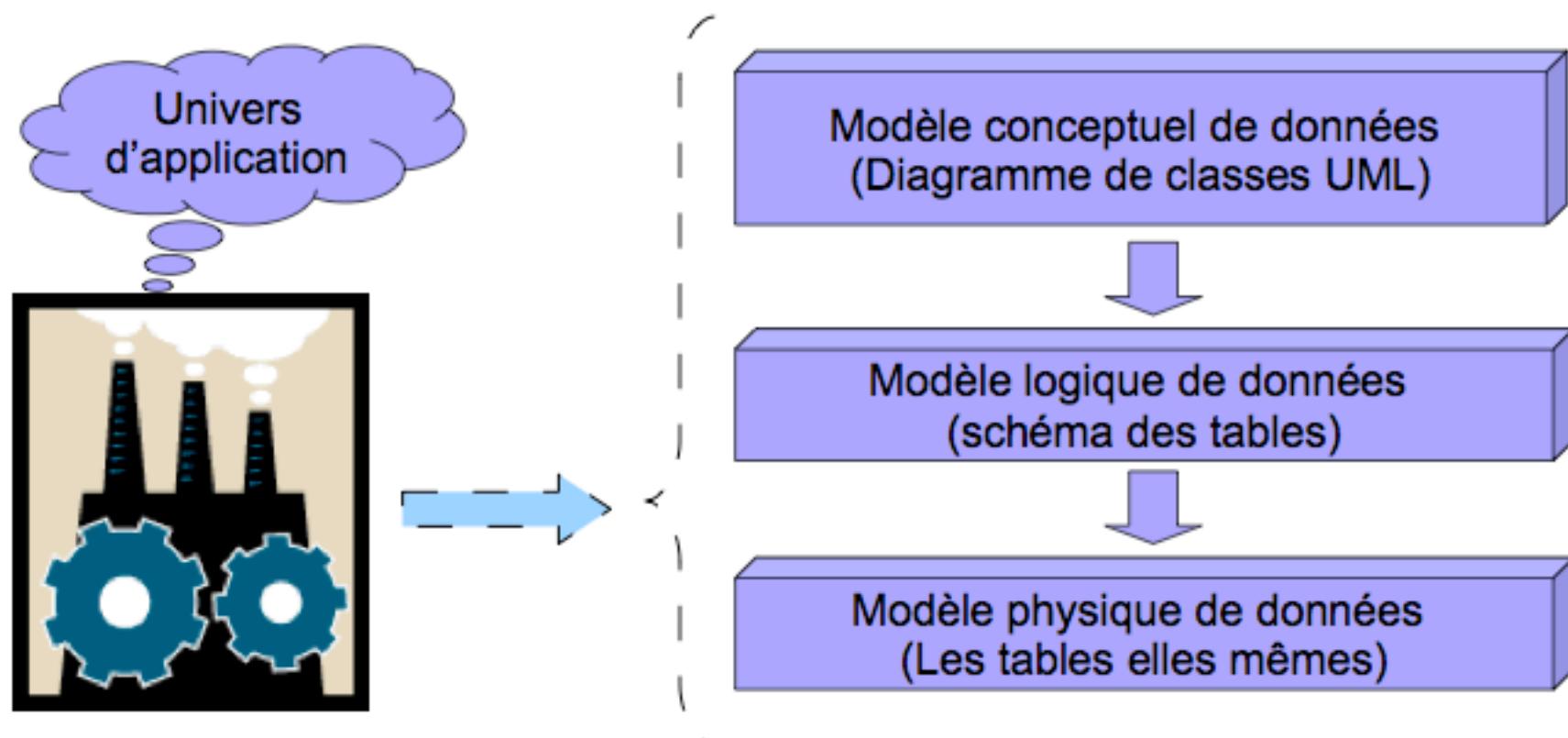
Table Formation

ID Formation	Libellé
1	SRC1
2	Licence Pro
3	SRC2

Rappel : modèle logique

- FILM(TITRE,PAYS,ANNEE,REALISATEUR,DUREE)
- DISTRIBUTION(TITRE,ACTEUR)
- CINEMA(NOMCINE, VILLE, TELEPHONE)
- CINESALLE(NOMCINE,NUMSALLE,NBPLACES,PMR)
- PROGRAMME(NOMCINE,NUMSALLE,SEMAINE,TITRE,VERSION)
- BOXOFFICE(NOMCINE,SEMAINE,TITRE,NBENTREE)

Démarche de développement de SI



Modèle conceptuel de données

- Représentation graphique de la structure de la base de données
 - L'utilisation d'un modèle graphique facilite
 - la conception
 - la compréhension de la structure de la base de données
- => Modèle entité-association

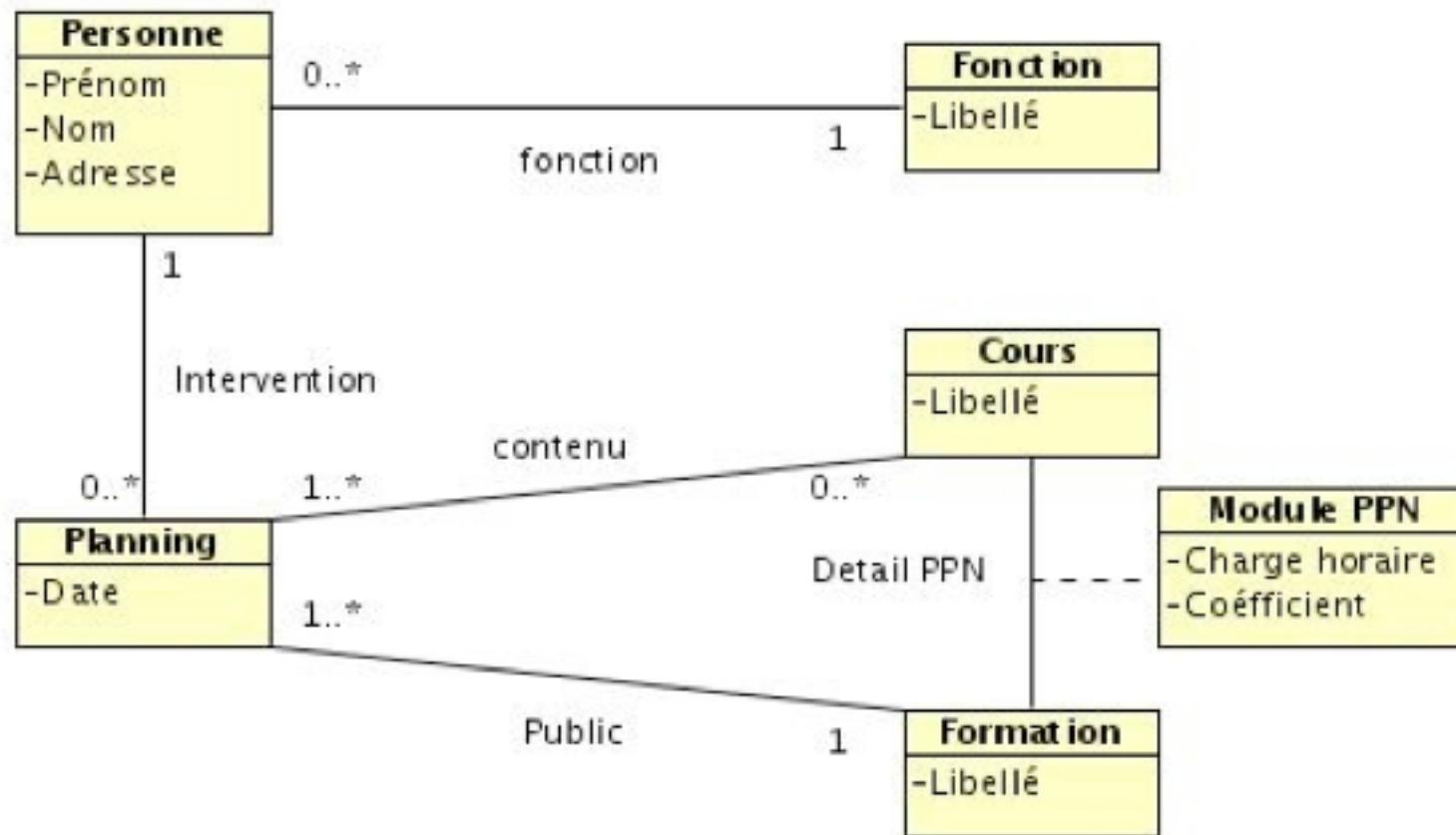
Modèle entité-association

- Rappel : un modèle est un ensemble de concepts permettant de définir la structure d'une base de données
- Les concepts du modèle entité-relation :
 - entités
 - associations
 - cardinalités

Objectifs

- Comprendre et valider un modèle entité-association
- Représenter graphiquement les ensembles de données et leurs liens sémantiques
 - sans se préoccuper du modèle logique (hiérarchique, relationnel, objet)
 - et encore moins du modèle physique (SGBD choisi)

Modèle entité-association

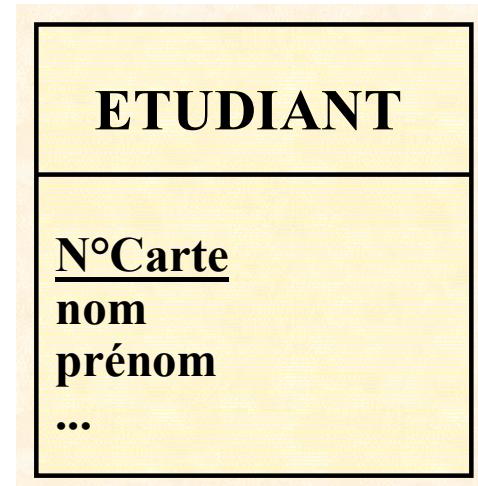


Entité

- Définition

- ensemble d'attributs caractérisés par un nom et décrivant une entité du monde réel
- attributs soulignés = clé primaire

- Représentation graphique :
à l'aide d'un rectangle



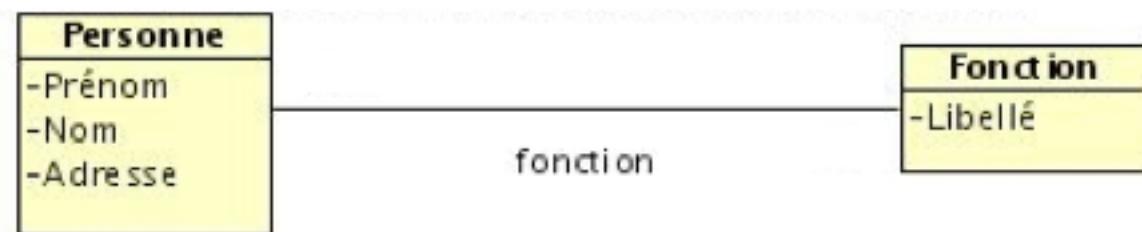
Association

- Définition

- lien entre des entités, généralement caractérisés par un nom.
- Peut posséder des **attributs**

- Représentation graphique

- Se dessine à l'aide d'un trait relié aux entités concernées



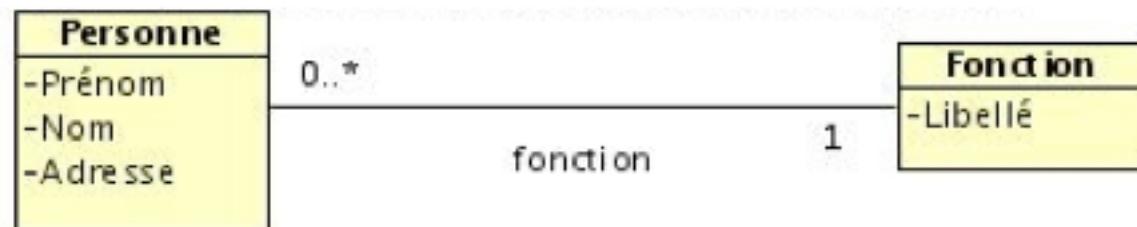
Cardinalités

- Définition

- valeurs définissant le type de lien exprimé par une relation

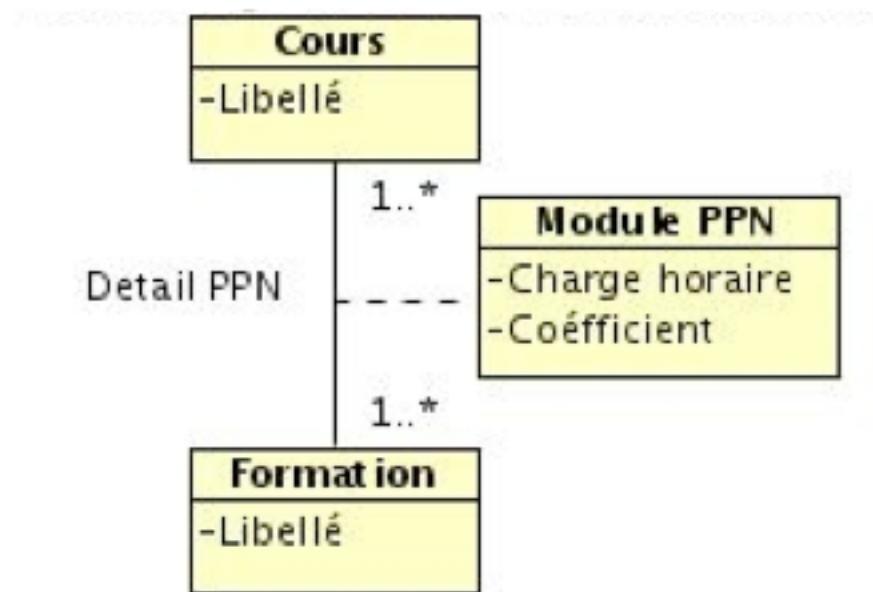
- Représentation graphique

- Couple de valeurs (1/0,1) ou (0/1,n)



Type d'association

- Selon le nombre d'entités qu'elle relie:
 - réflexive
 - binaire
 - ternaire
 - ...
 - n-aire



Du modèle conceptuel au modèle logique

- Tout deux définissent la structure de la base de données
 - ils doivent donc être cohérents entre eux
- Le modèle logique peut se déduire du modèle conceptuel
- Cette déduction suit des règles précises !

Modèle conceptuel -> modèle logique I

I. Créer une relation par entité

- attributs = attributs de l'entité
- choisir une clé primaire

Personne
-Prénom
-Nom
-Adresse



Personne (Prénom, Nom, Adresse)

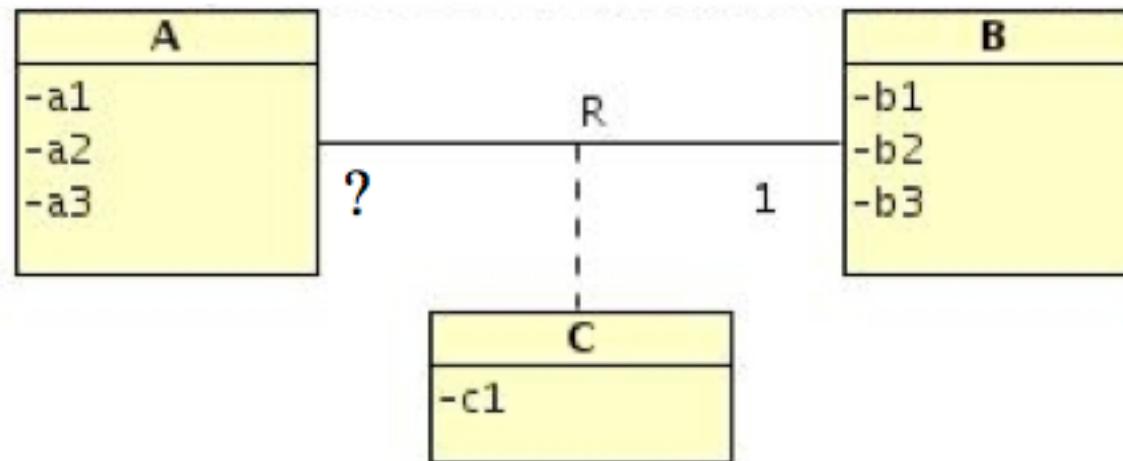
ou

Personne (ID_Personne, Prénom, Nom, Adresse)

Modèle conceptuel -> modèle logique 2

2. Se déterminer en fonction des types d'association (binaire ou +) et des cardinalités (1 ou autre)
 - ▶ association binaire et cardinalité de 1 : faire référence à la clé primaire
 - ▶ Autres associations et/ou autres cardinalités : créer une nouvelle relation avec les clés primaires des entités existantes

-> Modèle logique cardinalité de 1

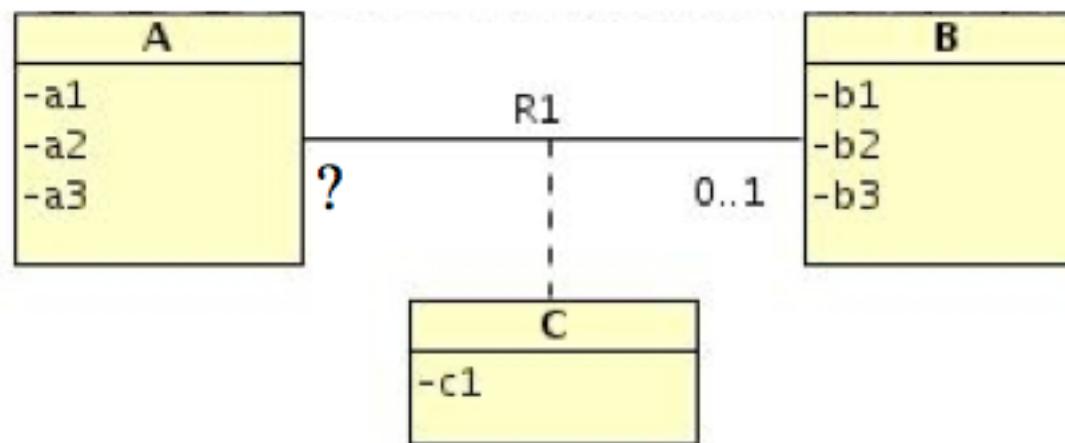


A(a1, a2, a3)
B(b1, b2, b3)



A(a1, a2, a3, #b1, c1)
B(b1, b2, b3)

-> Modèle logique cardinalité de 0..1

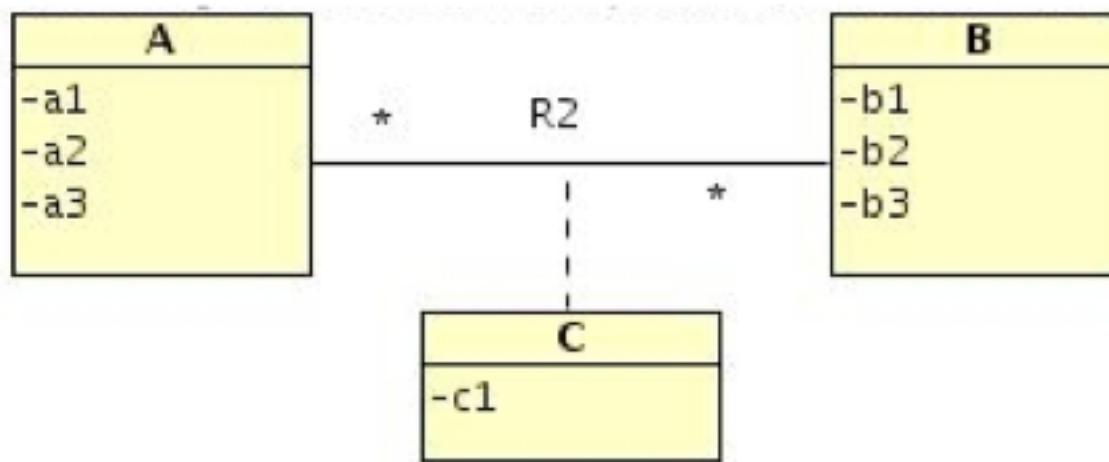


A1(a1, a2, a3)
B1(b1, b2, b3)



R1(#a1, #b1, c1)

-> Modèle logique cardinalité *

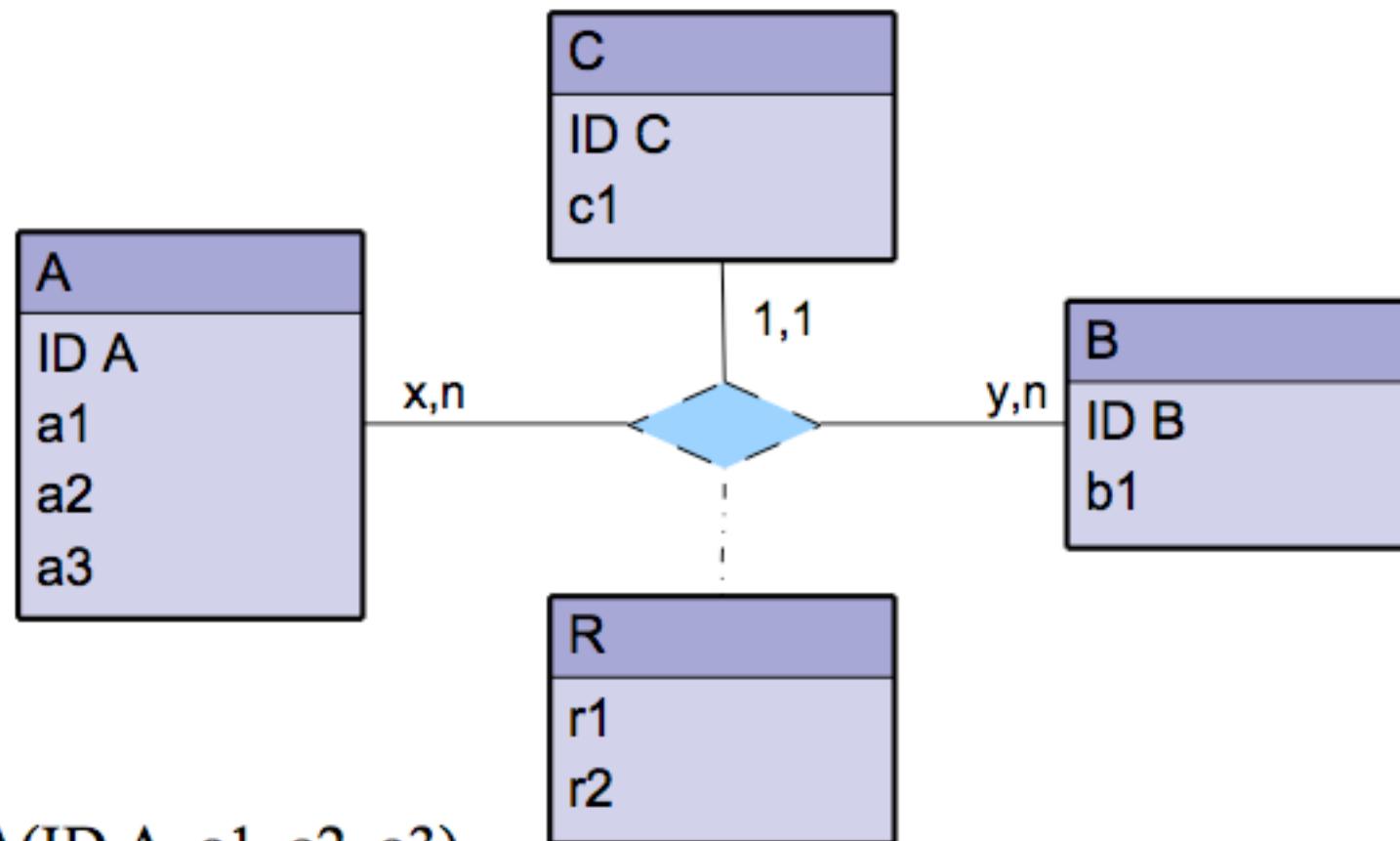


A(a1, a2, a3)
B(b1, b2, b3)



R2(#a1, #b1, c1)

-> Modèle logique relation ternaire



A(ID A, a1, a2, a3)
B(ID B, b1)
C(ID C, c1)

= = => R(#ID A, #ID B, #ID C, r1,r2)

Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Objectifs

- Support mathématique cohérent
 - sur lequel repose le modèle relationnel
- Décrire les opérations qu'il est possible d'appliquer sur les relations
 - Approche plus opérationnelle que mathématique
- Notations non standardisées
 - courantes mais pas forcément universelles

Opérations sur relations

- Les opérations unaires (Sélection, Projection): elles consistent à éliminer des lignes ou/et des colonnes de la table
- Les opérations ensemblistes (Union, Produit cartésien, Différence, etc.): elles s'applique sur deux ou plusieurs tables

Sélection unaire

- La sélection génère une relation regroupant exclusivement toutes les occurrences de la relation R qui satisfont le critère de sélection
- Syntaxe : $R = \text{SELECTION} (R_1, \text{Critère})$

Critère de sélection défini avec les opérateurs $<$ $<=$

$>$ $=>$ $=$ $!=$ \wedge \vee et $()$

Fonction

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant



R

ID Fonction	Libellé
1	Enseignant

$R = \text{SEL} (\text{Fonction}, \text{ID Fonction} = 1)$

Projection unaire

- La projection génère une relation regroupant exclusivement toutes les occurrences de la relation R réduites aux attributs de la liste d'attributs de projection donnés.
- Syntaxe : $R = \text{PROJECTION} (R_1, \text{Liste d'attributs})$

Personne				R
ID Personne	Prénom	Nom	Fonction	
1	Oualid	Khayati	1	
2	Christine	Rolland	3	
3	Christophe	Courtin	2	

$R = \text{PROJ} (\text{Personne}, \text{ID Personne}, \text{Prénom})$

Composition d'opérations

- Suite d'opérations sur les relations
 - permettant d'obtenir une réponse précise

Exemple: Noms et prénom des Etudiants nés avant 1982?

Relation de départ =
Etudiant

$R_1 = Sel (Etudiant, Date-naiss < '01/01/1982')$

$R_2 = Proj (R1, nom, prénom)$

Etudiant

N°Carte	Nom	Prénom	Date-naiss
A745862	Martin	Anne	15/06/1980
745TP12	Dupart	Marc	07/08/1982
85A2153	Pouchet	Aurélie	10/05/2982
78TAR12	Frest	Paul	04/08/1981

R_1

N°Carte	Nom	Prénom	Date-naiss
A745862	Martin	Anne	15/06/1980
78TAR12	Frest	Paul	04/08/1981

R_2

Nom	Prénom
Martin	Anne
Frest	Paul

Produit cartésien

- Syntaxe : $R = R1 \times R2$

Table Fonction

ID Fonction	Libellé
1	Enseignant
2	Chef département
3	Étudiant

Table Cours

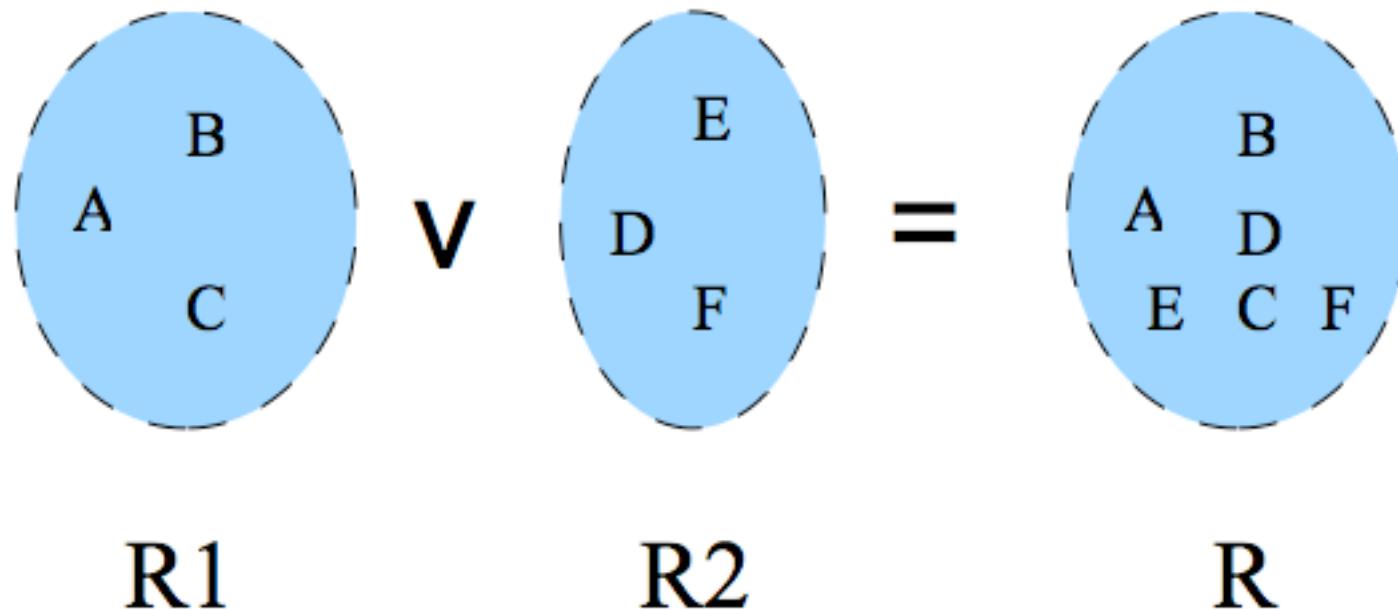
ID Cours	Libellé
1	Base de données
2	Algorithmique
3	Systèmes d'exploitation

$R = \text{Fonction} \times \text{Cours}$

ID Fonc	Libellé Fonc	ID Cours	Libellé
1	Enseignant	1	Base de données
1	Enseignant	2	Algorithmique
1	Enseignant	3	Systèmes d'exploitation
2	Chef département	1	Base de données
2	Chef département	2	Algorithmique
2	Chef département	3	Systèmes d'exploitation
3	Étudiant	1	Base de données
3	Étudiant	2	Algorithmique
3	Étudiant	3	Systèmes d'exploitation

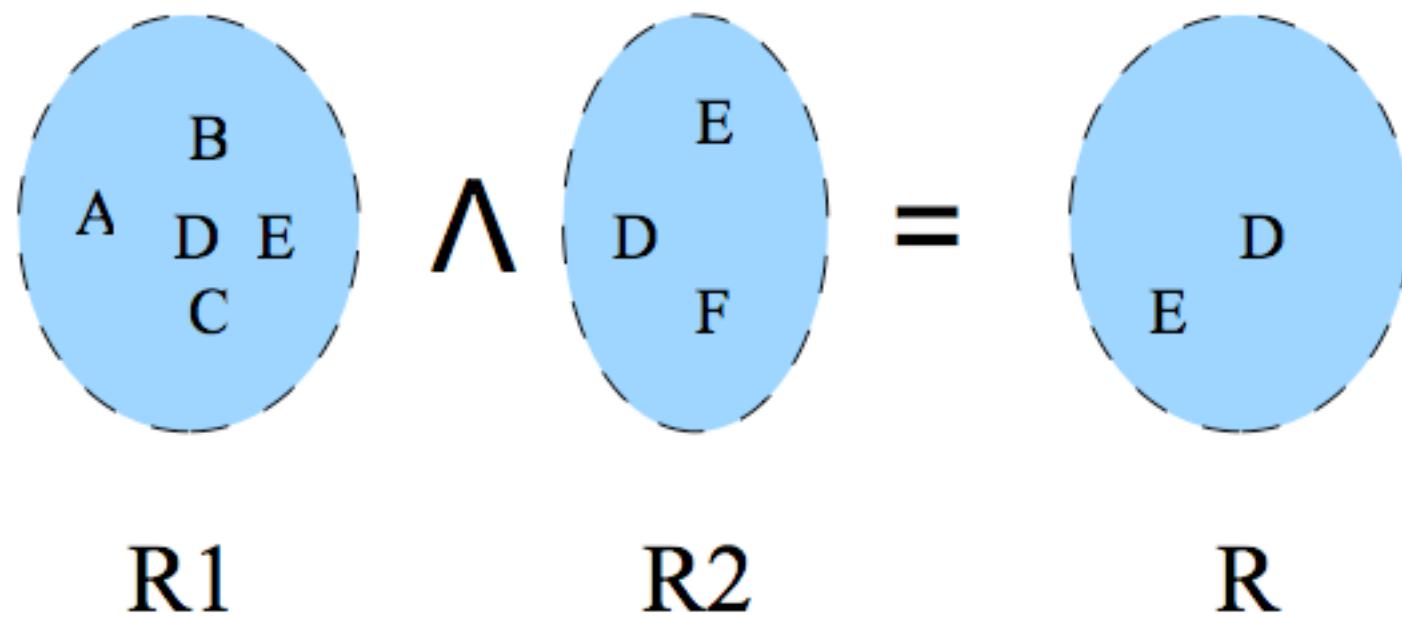
Union

- Syntaxe 1 : $R = R_1 \vee R_2$
- Syntaxe 2 : $R = \text{UNION} (R_1, R_2)$



Intersection

- Syntaxe 1 : $R = R1 \wedge R2$
- Syntaxe 2 : $R = \text{INTERSECTION} (R1, R2)$



Jointure

- Syntaxe : $R = \text{JOINTURE}(R1, R2, Critère)$

La relation R contient toutes les lignes de $R1$ concaténées avec les lignes de $R2$ pour lesquelles le critère de jointure est vérifié.

- Équivalence :

$$R3 = R1 \times R2$$

$$R = \text{SELECTION}(R3, Critère)$$

Exemple de jointure

Exemple: Noms, prénoms et études des étudiants?

R = Join(Etudiant, Inscriptions, *N°Carte* = *N°Etu*)

N°Carte	Nom	Prénom	Date-naiss
A745862	Martin	Anne	15/06/1980
745TP12	Dupart	Marc	07/08/1982
85A2153	Pouchet	Aurélie	10/05/2982
78TAR12	Frest	Paul	04/08/1981

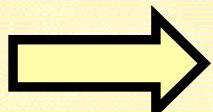
Etudiant

N°Etu	Etudes	Année
A745862	SRC1	1999-2000
85A2153	LEA1	1999-2000
A745862	SRC1	2002-2003
A745862	SRC2	2003-2004
A745862	SRC2	2004-2005
78TAR12	LEA1	2002-2003

Inscriptions

R

N°Carte	Nom	Prénom	Date-naiss	Etudes	Année
A745862	Martin	Anne	15/06/1980	SRC1	1999-2000
A745862	Martin	Anne	15/06/1980	SRC1	2002-2003
A745862	Martin	Anne	15/06/1980	SRC2	2003-2004
A745862	Martin	Anne	15/06/1980	SRC2	2004-2005
85A2153	Pouchet	Aurélie	10/05/2982	LEA1	1999-2000
78TAR12	Frest	Paul	04/08/1981	LEA1	2002-2003



Regroupement

- Syntaxe :

$R = \text{Regroupement}(R1, \text{Att}, \text{Fonction(expression)})$

Cette opération regroupe les lignes de $R1$ possédant la même valeur pour l'attribut Att , et calcule le résultat de la fonction pour chaque groupe de lignes obtenu.

- Fonctions : COMPTE, SOMME, MOYENNE, MIN, MAX

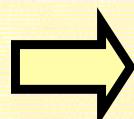
Regroupement : compte

Exemple: nombre d'étudiants inscrits dans chaque filières?

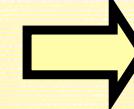
**R = Regroupement (Inscriptions, Etudes,
COMPTE(Etudes))**

Inscriptions

N°Etu	Etudes
A745862	SRC1
85A2153	LPATC
B785862	SRC1
A745862	SRC2
A745862	LPATC
78TAR12	SRC1



Etudes	N°Etu
SRC1	A745862
SRC1	B785862
SRC1	78TAR12
SRC2	A745862
LPATC	A745862
LPATC	85A2153



R

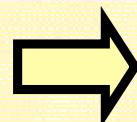
Etudes	Compte (Etudes)
SRC1	3
SRC2	1
LPATC	2

Regroupement : somme

Exemple: montant total des cotisations perçues par filières?

**R = Regroupement (Inscriptions, Etudes,
SOMME(cotis))**

Etudes	N°Etu	Cotis
SRC1	A745862	150
SRC1	B785862	150
SRC1	78TAR12	85
SRC2	A745862	160
LPATC	A745862	100
LPATC	85A2153	100



R

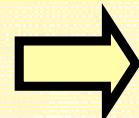
Etudes	somme (cotis)
SRC1	385
SRC2	160
LPATC	200

Regroupement : min

Exemple: montant minimal de cotisation perçue par filières?

**R = Regroupement (Inscriptions, Etudes,
MIN(cotis))**

Etudes	N°Etu	Cotis
SRC1	A745862	150
SRC1	B785862	150
SRC1	78TAR12	85
SRC2	A745862	160
LPATC	A745862	100
LPATC	85A2153	100



R

Etudes	Min (cotis)
SRC1	85
SRC2	160
LPATC	100

Division

- Syntaxe :

$R = \text{DIVISION}(R1, R2)$

La division génère une relation regroupant exclusivement toutes les parties de lignes de la relation R1 qui sont associées à toutes les lignes de la relation R2.

Division : exemple

Exemple: étudiants inscrits dans toutes les filières?

R = Inscriptions / Filières

Inscriptions

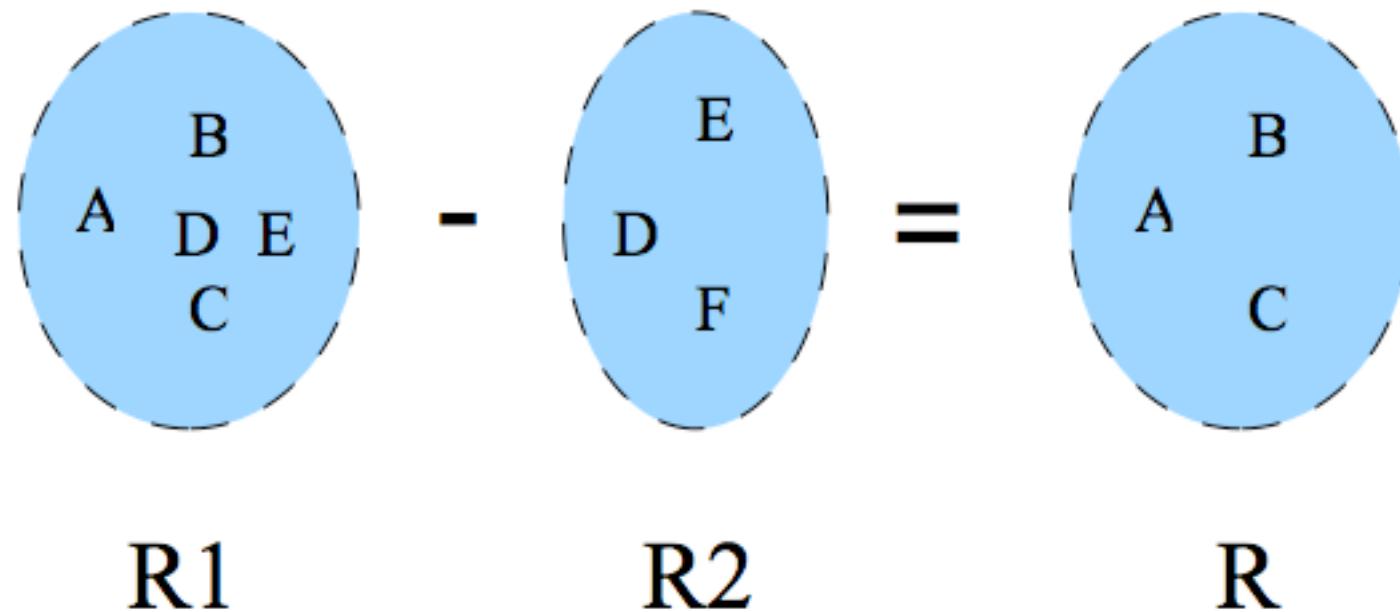
Filières	Code
SRC1	A745862
SRC2	85A2153
LPATC	A745862
	A745862
	A745862
	78TAR12

N°Etu	Etudes
A745862	SRC1
85A2153	LPATC
A745862	SRC1
A745862	SRC2
A745862	LPATC
78TAR12	SRC1



Différence

- Syntaxe 1 : $R = R_1 - R_2$
- Syntaxe 2 : $R = \text{DIFFERENCE}(R_1, R_2)$



Composition d'opérations

Exemple: noms et prénoms des étudiants
inscrits en src1 en 1999-2000?

R1 = Sel (Inscriptions, Année = ‘ 1999-200 ’)

R2 = Sel (R1, filière = ‘ src1 ’)

R3 = Join (R2, Etudiants, N°Carte = N°Etu)

Résultat = Proj (R3, Nom, Prénom)

ou



R1 = Inscriptions X Etudiants

R2 = Sel (R1, (N°Carte = N°Etu \wedge filière = ‘ src1 ’ \wedge Année = ‘ 1999-200 ’))

Résultat = Proj (R2, Nom, Prénom)

Exemple de la bibliothèque

ABONNE (N°Carte, nom, prénom)

EMPRUNTS(Code, numéro, date, N°Carte)

LIVRE (Code, titre)

EXEMPLAIRE(Code, num, date-achat, état, prix d'achat)

Exemplaire

<i>Code</i>	<i>num</i>	<i>année achat</i>	<i>état</i>	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

Abonné

<i>N°Carte</i>	<i>nom</i>	<i>prenom</i>
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei

Livre

<i>Code</i>	<i>titre</i>
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

Emprunts

<i>Code</i>	<i>num</i>	<i>date</i>	<i>N°Carte</i>
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12

Liste de requêtes

Exemplaire

Code	num	année achat	état	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

Abonné

N°Carte	nom	prenom
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei

Livre

Code	titre
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

Emprunts

Code	num	date	N°Carte
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12



1. Liste des abonnés? (nom, prénom)
2. Code des livres en TB état?
3. Titres des livres en TB état?
4. Noms des personnes ayant loué « L'alchimiste »?
5. N°Carte des abonnés ayant loué tous les livres?
6. Nombre d'exemplaires par livre?
7. Prix moyen d'achat de chaque livre?
(Code, titre, prix moyen)

I. Liste des abonnés

Abonné

N°Carte	nom	prenom
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei

R₁

nom	prenom
Duran	Eric
Riu	Léa
Chong	Wei

R₁ = Proj(Abonné, nom, prénom)

Exemplaire

Code	num	année achat	état	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

Abonné

N°Carte	nom	prenom
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei

Livre

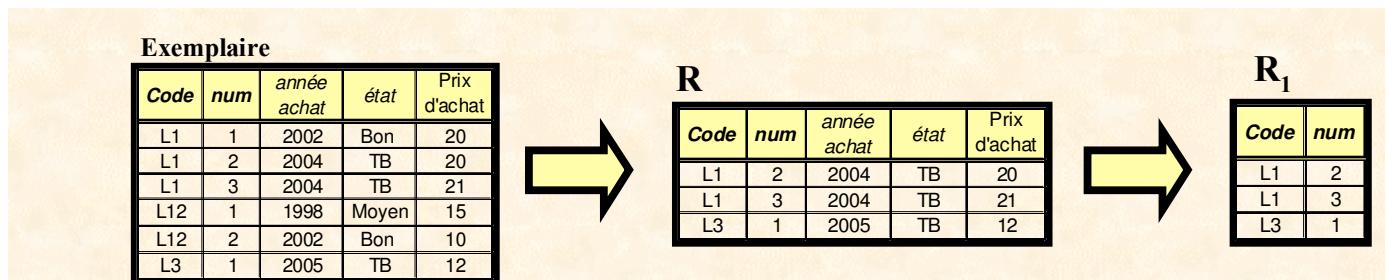
Code	titre
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

Emprunts

Code	num	date	N°Carte
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12

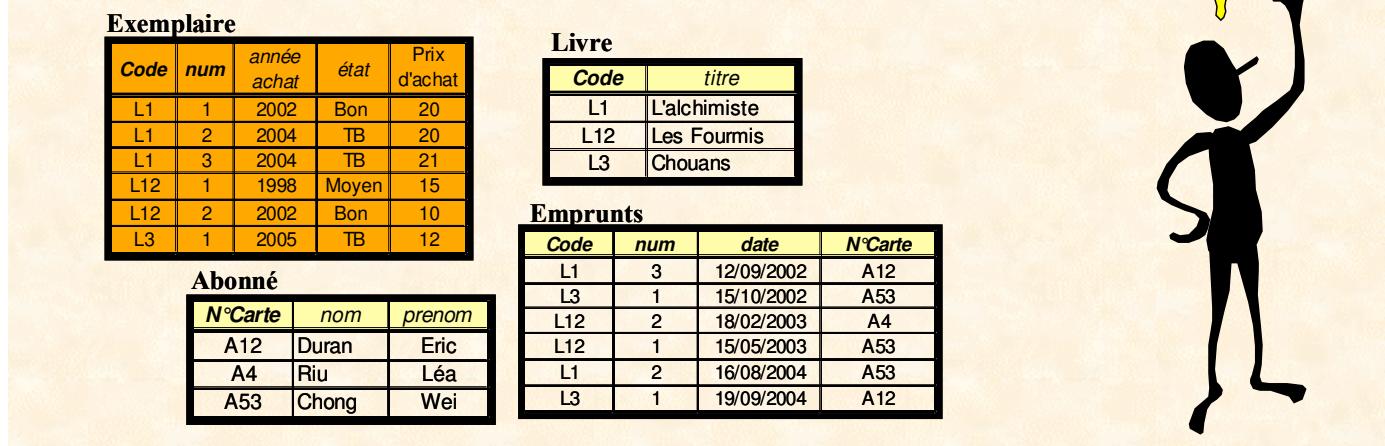


2. Livres en TB état

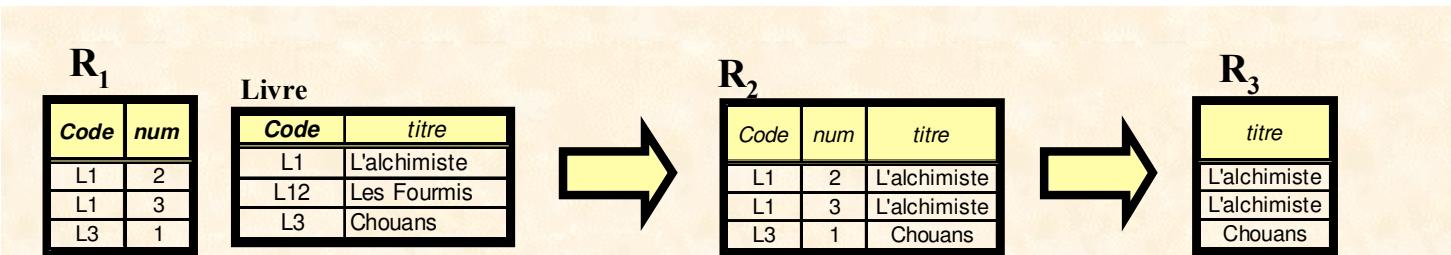


R = Sel (Exemplaire, état = 'TB')

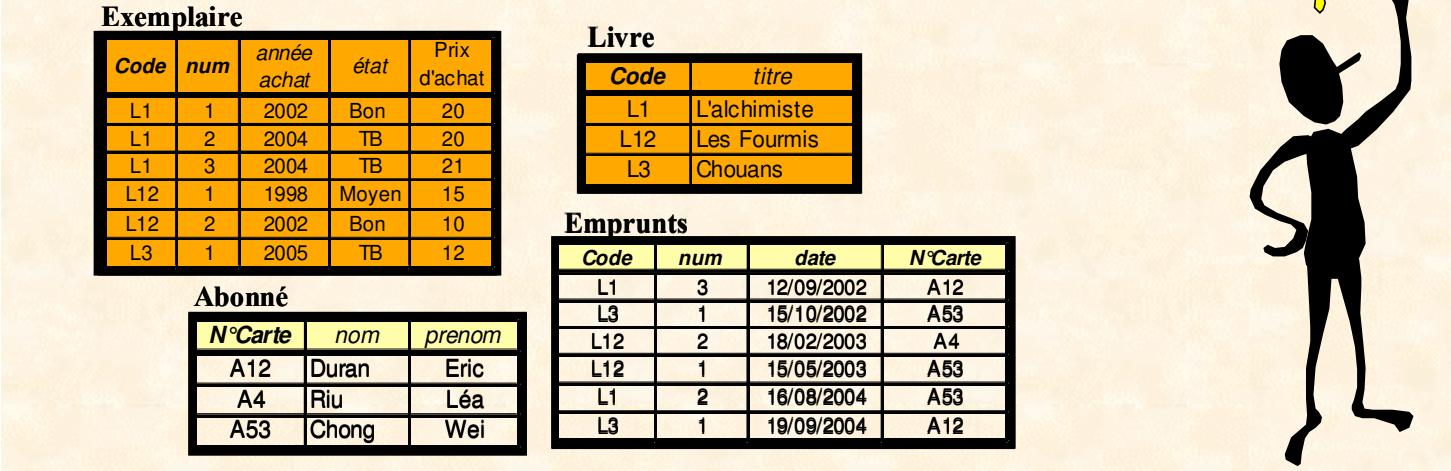
R₁ = Proj (R, Codes)



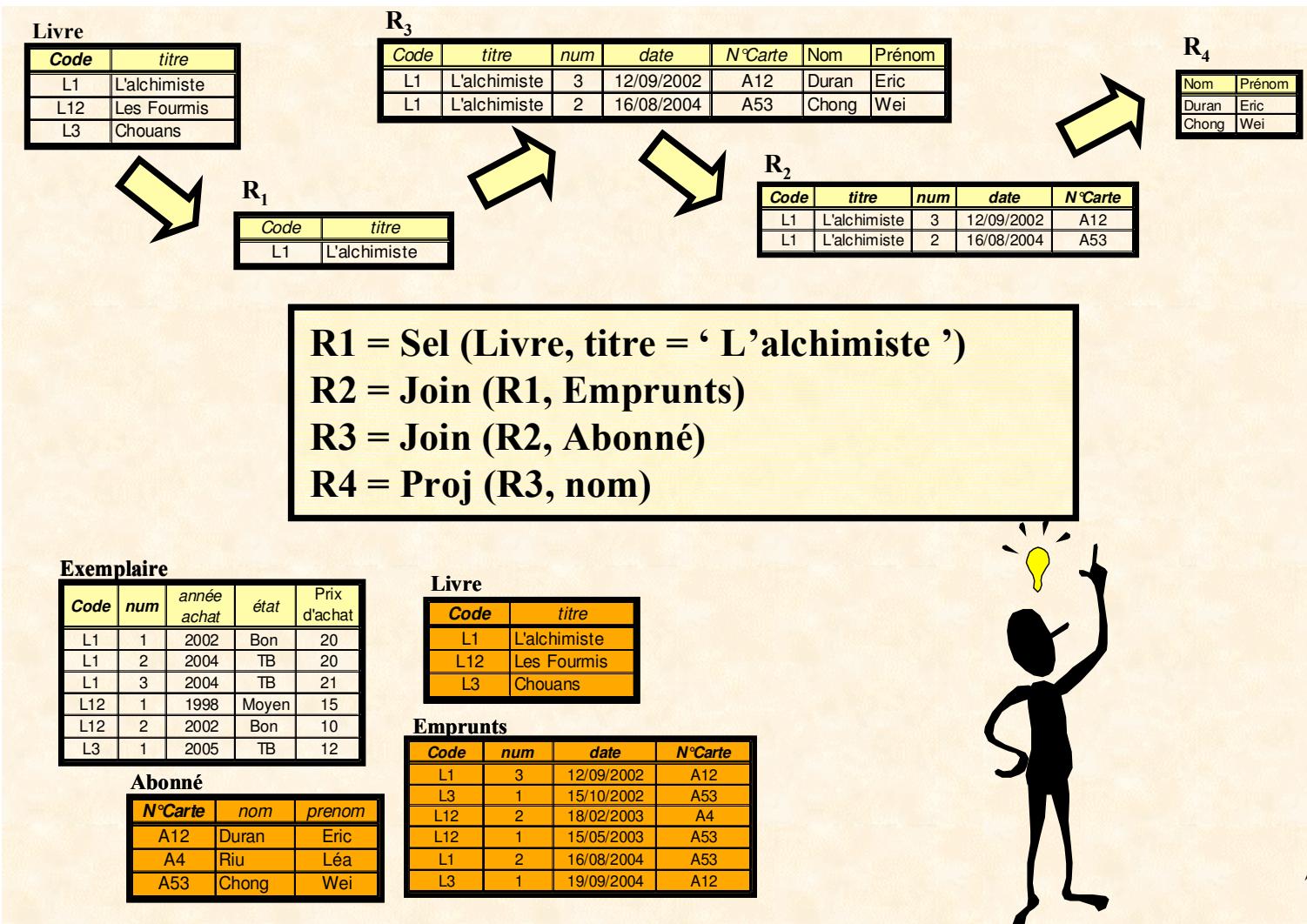
3. Titre des livres en TB état



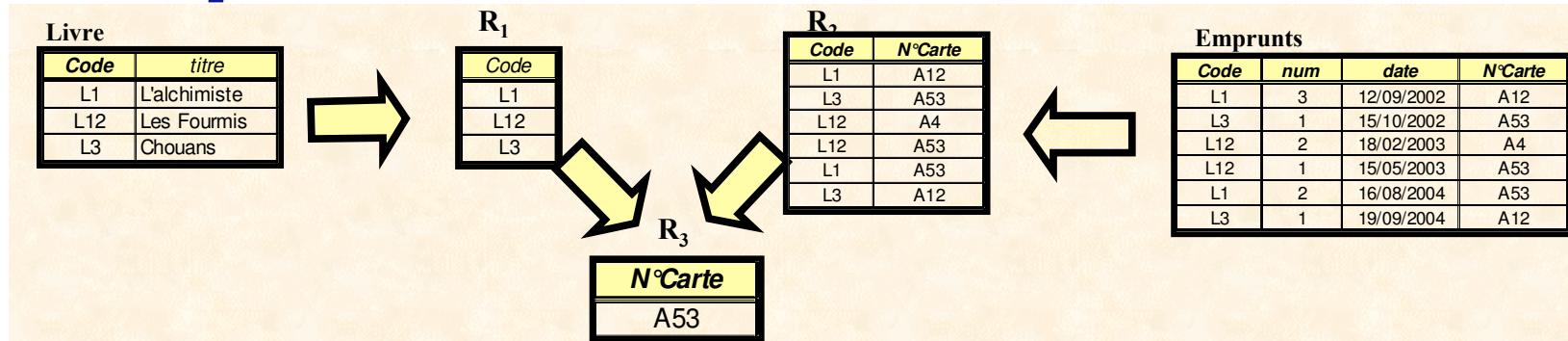
R₂ = Join(R₁, Livre)
R₃ = Proj (R₂, Titre)



4. Nom des personnes ayant emprunté l'alchimiste



5. N° carte des abonnés ayant emprunté tous les livres



R₁ = Proj(Livre, Code)
R₂ = Proj(Emprunts, Code, N°Carte)
R₃ = R₁/R₂

The tables shown are:

- Exemplaire** table:

Code	num	année achat	état	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

- Livre** table:

Code	titre
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

- Abonné** table:

N°Carte	nom	prenom
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei

- Emprunts** table:

Code	num	date	N°Carte
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12



6. Nombre d'exemplaires par livre

Exemplaire				
<i>Code</i>	<i>num</i>	<i>année achat</i>	<i>état</i>	<i>Prix d'achat</i>
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

→

R_1	
<i>Code</i>	<i>compte (code)</i>
L1	3
L12	2
L3	1

R1 = Regroupement(Exemplaire, Code, compte(Code))

Exemplaire				
<i>Code</i>	<i>num</i>	<i>année achat</i>	<i>état</i>	<i>Prix d'achat</i>
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

Livre	
<i>Code</i>	<i>titre</i>
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

Emprunts			
<i>Code</i>	<i>num</i>	<i>date</i>	<i>N°Carte</i>
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12

Abonné		
<i>N°Carte</i>	<i>nom</i>	<i>prenom</i>
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei



7. Prix d'achat moyen de chaque livre

Exemplaire

Code	num	année achat	état	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

R₁

Code	compte (code)
L1	20,3
L12	12,5
L3	12

R₂

Code	titre	moyenne (prix)
L1	L'alchimiste	20,3
L12	Les Fourmis	12,5
L3	Chouans	12

R₁ = Regroupement(Exemplaire, Code, moyenne(prix d'achat))

R₂ = Join(R₁, Livre)

Exemplaire

Code	num	année achat	état	Prix d'achat
L1	1	2002	Bon	20
L1	2	2004	TB	20
L1	3	2004	TB	21
L12	1	1998	Moyen	15
L12	2	2002	Bon	10
L3	1	2005	TB	12

Livre

Code	titre
L1	L'alchimiste
L12	Les Fourmis
L3	Chouans

Emprunts

Code	num	date	N°Carte
L1	3	12/09/2002	A12
L3	1	15/10/2002	A53
L12	2	18/02/2003	A4
L12	1	15/05/2003	A53
L1	2	16/08/2004	A53
L3	1	19/09/2004	A12

Abonné

N°Carte	nom	prenom
A12	Duran	Eric
A4	Riu	Léa
A53	Chong	Wei



Plan

- I Introduction
- II Modèle relationnel
- III Modèle conceptuel
- IV Algèbre relationnelle
- V Langage SQL

Qu'appelle-t-on SQL ?

- Langage d'accès normalisé aux BD
- Supporté par la +part des SGBD
- Plusieurs normes (SQL2)
- S'appuie sur le schéma conceptuel
- Moins puissant qu'un langage de programmation (pas de conditions, boucles, ...)

SQL : Structured Query Language

- Langage structuré de requêtes
 - Langage de définition de données (LDD)
 - ▶ Créer, modifier, et supprimer des tables
 - Langage de manipulation de données (LMD)
 - ▶ Insérer, modifier, supprimer des enregistrements
 - ▶ Extraire des données (requêtes)
 - Langage de contrôle de données (LCD)
 - ▶ Créer, modifier, supprimer des droits d'accès

LDD : création de relation

- Pour créer une relation, connaître (au minimum):
 - Son nom,
 - Ses attributs et leur domaine
 - Sa clé primaire
- CREATE TABLE Etudiant (N°Carte Char(7) NOTNULL, Nom Char(20), Prenom Char(20) PRIMARY KEY (N°Carte))
- *CREATE TABLE <nom_de_relation> (<nom_d'attribut> <type_de_données> [NOT NULL] [, ...] [, PRIMARY KEY (<nom_d'attribut> [, ...])])*

LLD : suppression de relation

- Pour supprimer une relation, connaître :
 - Son nom
- **DROP TABLE Etudiant**
- ***DROP TABLE <nom_de_relation>***

LDD : modification de relation

- Pour modifier une relation, connaître :
 - Son nom
 - Les attributs à ajouter ou à supprimer.
- ALTER TABLE Etudiant ADD Adresse Char(100)
- *ALTER TABLE <nom_de_relation> ADD|DROP (<nom_d'attribut> <type_de_données>)*

LMD : insertion d'enregistrement

- Pour insérer un enregistrement, il faut connaître :
 - Toutes les valeurs d'attributs de l'enregistrement
 - ou certaines valeurs seulement
- `INSERT INTO Etudiant VALUES ('A745865', 'Martin', 'Anne', ' 20/06/1980 ')`
- `INSERT INTO Etudiant (N°Carte, Nom) VALUES('A745865', 'Martin')`
- *`INSERT INTO <nom_de_relation> [<nom_d'attribut>, ...] [VALUES <valeur> [...]]`*

LMD : suppression d'enregistrement

- Pour insérer un enregistrement, il faut connaître :
 - Toutes les valeurs d'attributs de l'enregistrement
 - ou certaines valeurs seulement
- `INSERT INTO Etudiant VALUES ('A745865', 'Martin', 'Anne', ' 20/06/1980 ')`
- `INSERT INTO Etudiant (N°Carte, Nom) VALUES('A745865', 'Martin')`
- *`INSERT INTO <nom_de_relation> [<nom_d'attribut> [, ...]] [VALUES <valeur> [...]]`*

LMD : modification d'enregistrement

- Pour un modifier des enregistrements, connaître :
 - Le(s) enregistrement (s) à modifier
 - Le critère permettant de le(s) identifier
 - Les attributs à modifier et leur valeur
- UPDATE Etudiant SET (date-naiss =02/08/1982)
WHERE (Nom = ' Bon ' AND Prenom = ' Julie ')
- *UPDATE <nom_de_relation> SET
[(<nom_d'attribut>=<valeur> [, ...]) WHERE Critère de
Sélection*

LMD : extraction de données

```
SELECT [liste d'attributs]
FROM [liste de tables]
WHERE [critère de sélection]
GROUP BY [attribut]
[HAVING condition sur les groupes]
ORDER BY [liste d'attributs]
ASC | DESC
```

LMD : extraction de données

```
SELECT [ * | DISTINCT] att1 [, att2, att3, ...]
FROM Table1 [, Table2, Table3, ...]
[WHERE conditions de sélection et/ou de jointure]
[GROUP BY att1 [, att2, ...] [HAVING conditions de sélection]]
[ORDER BY att1 [ASC | DESC] [, att2 [ASC | DESC], ...] ;
```