

Introduction aux Bases de Données

Sup Galilée-INFO1

Céline Rouveirol

2013-2014

Crédits

Ce cours s'appuie sur le cours de :

- François Goasdoué et Marie-Pierre Dorville, LRI, Université Paris 11
- Laurent Audibert, IUT Villetaneuse, Université Paris 13

Objectifs et plan du cours

- Maîtrise des concepts de base en Bases de Données Relationnelles + introduction programmation SQL
- Plan du cours
 - Chapitre 1 : Introduction
 - Chapitre 2 : Conception de BDR
 - Modèle E/A
 - Chapitre 3 : Théorie des BDR
 - Modèle relationnel
 - Du modèle E/A au modèle relationnel
 - Algèbre relationnelle
 - Chapitre 4 : Mise en oeuvre d'une BDR : SQL
 - Tables et contraintes
 - Requêtes, vues, ...

Introduction aux bases de données

- Une base de données (BD) est un ensemble structuré de données cohérentes
 - ensemble** : collection disponible, accessible, extensible d'informations
 - ajout, suppression, modification
 - structuré** : collection organisée grâce à des structures logiques
 - tables, cubes, arbres, ...
 - données cohérentes** : des contraintes peuvent être posées sur les données.

- Un système de gestion de bases de données (SGBD) est un logiciel qui permet de manipuler des bases de données :
 - construction
 - maintenance
 - interrogation
 - sécurité
- Propriétés fondamentales d'un SGBD
 - Indépendance des données : données physiques / conceptuelles / applications
 - Accès efficace aux données (Langage de Manipulation de Données)
 - Partage des données (Accès concurrents + sécurité des données)
 - Intégrité et sécurité des données
 - Administration centralisée des données
 - et récupération sur «crash»

- Applications de gestion
 - Banque - Finance - Assurance
 - Gestion du personnel / clients / stocks d'une entreprise
 - Système d'information d'une entreprise : ensemble organisé de ressources (données, procédures, matériels, logiciels, ...) permettant d'acquérir, de stocker, de structurer et de communiquer des informations sous forme de textes, images, sons, ou de données codées dans des organisations [Wikipedia].
- Applications transactionnelles
 - Gestion de réservations (transports, hotels, spectacles, ...)
 - Consultations et modifications très fréquentes

Pourquoi ne pas utiliser un simple système de gestion de fichiers ?

- Les données dans les programmes qui les utilisent
 - duplication des données : pb de maintenance, de cohérence des données, accroissement de la masse des données
- Les données dans un fichier
 - performance (temps d'accès aux données)
 - coût d'exploitation des données : multiplications des programmes ad-hoc pour l'accès aux données (requêtes), la gestion de la cohérence, concurrence, confidentialité, etc.

- 3 niveaux de description
 - Niveau interne (gestion des accès) : schéma physique
 - Niveau conceptuel (intégrité-cohérence) : schéma logique, issu d'un processus de modélisation
 - Niveau externe (confidentialité), n schémas externes associés à n types d'utilisateurs
- Architecture client-serveur
 - Un serveur pour le SGBD : gérant les bases de données communes
 - De multiples clients pour accéder aux BDs :
 - gérant les interfaces serveur-utilisateurs/applications
 - permettant la distribution des applications.

- Le modèle hiérarchique (années 60) : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements, organisés dans une structure arborescente de façon à ce que chaque enregistrement n'ait qu'un seul possesseur. Il s'agit du premier modèle de SGBD
- Le modèle réseau (années 70) : lever de nombreuses difficultés du modèle hiérarchique grâce à la possibilité d'établir des liaisons de type n-n, les liens entre objets pouvant exister sans restriction. Pour retrouver une donnée dans une telle modélisation, il faut connaître le chemin d'accès (les liens), ce qui rend les programmes dépendants de la structure de données

- Le modèle Relationnel (Codd, fin des années 60). Les données sont représentées dans des tables, sous forme de n-uplets. Modèle le plus utilisé : celui que nous considérerons dans la suite. A donné lieu au langage SQL, extension de l'algèbre relationnelle, standardisation en 1987. Près de 80% des utilisations en entreprise.
- Le modèle Orienté Objet (géré par des OODBMS). Les données sont des objets, encore en phase de recherche, BD avancées en M1.
- Le modèle Multidimensionnel. Les données sont représentées sous la forme d'un cube. Très utilisé en analyse de données, M2.
- Le modèle Semi-structuré (fichiers XML). Les données sont représentées sous la forme d'arbre, M2.

- (O)RDBMS :
 - Oracle : actuellement en version 10g,
 - Microsoft : Access (inclus dans Office), SQL Server
 - IBM : DB2
 - Les libres : PostgreSQL, MySQL, ...
- OODBMS : O².

- 1960 :

Mémoire	Coûteuse
Applications	Gestion, stock
Données	alphanumérique

- 2000 :

Mémoire	Bon marché
Applications	CAO, Génomique, Multi-média
Données	alphanumérique, image son, vidéo
Besoins	Tera-octets : 10 ¹² octets ¹

Conception de bases de données relationnelles

- 1 La démarche de modélisation conceptuelle
- 2 Le modèle Entité – Association (E/A)
- 3 Du modèle E/A au modèle relationnel

Conception de bases de données relationnelles

Une démarche de conception consiste à construire une représentation de la réalité à l'aide d'un modèle conceptuel de données ou modèle sémantique

- représentation graphique

Le modèle conceptuel sera (ensuite) traduit dans un modèle logique de données

Conception de bases de données relationnelles

Outils de modélisation

modèle conceptuel
« universel » E/A
(entité/association)
modèle sémantique



modèles logiques

- réseau
- hiérarchique
- relationnel
- ...

Conception de bases de données relationnelles

le modèle entité/association (P. CHEN 1976)

Il repose sur 3 concepts fondamentaux

- l'entité : représentation d'un objet de la réalité
- l'association : lien entre entités
- l'attribut : caractéristique propre à une entité ou une association

Entité ou type d'entité (1/2)

- regroupement d'objets qui ont les mêmes caractéristiques ou propriétés

⇒ *Exemple* : les enseignants, les étudiants, les salles, les cours, ... constituent différents types d'entités.

- une entité est identifiée par un nom et décrite par les propriétés des objets qu'elle représente, appelées attributs de l'entité

⇒ *Exemple* : un étudiant est caractérisé par un nom, un prénom, une date de naissance.

Attribut

- prend sa valeur dans un « domaine de valeurs »
- il peut être simple (ex : nom) ou composé (ex : date de naissance composée de jour, mois, année)
- identifiant** : c'est un attribut ou un groupe d'attributs qui identifie de façon unique une occurrence d'entité.

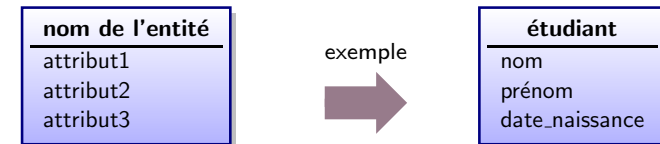
⇒ *Exemple* : aucun des attributs nom, prénom, date de naissance (ou leur regroupement) ne peut être l'identifiant de l'entité étudiant ; il faut donc créer un nouvel attribut numéro qui est unique pour chaque étudiant.

Entité ou type d'entité (2/2)

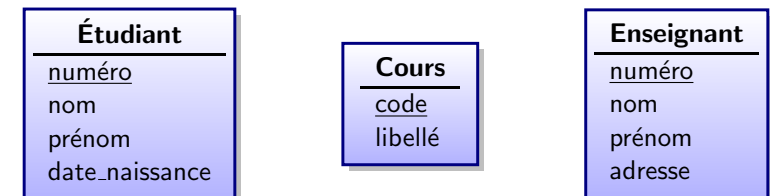
- une occurrence d'entité est un objet particulier

⇒ *Exemple* : « Dupont, Jean, 15/12/1982 » est une occurrence de l'entité étudiant.

- formalisme utilisé pour représenter une entité :



Représentation des entités



Association (1/5)

- ensemble de liens entre deux ou plusieurs entités où chaque entité joue un rôle déterminé

⇒ *Exemple* : l'association inscription représente tous les liens possibles entre les étudiants inscrits aux différents cours.

- une association est identifiée par un nom et des attributs qui lui sont propres

⇒ *Exemple* : l'association inscription est caractérisée par l'année d'inscription.

Association (2/5)

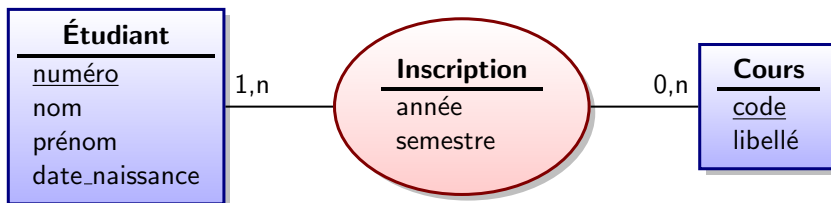
- Cardinalité** : traduit le type de liens entre les occurrences des entités de l'association. Elle s'exprime par un couple de 2 valeurs qui sont les nombres minimum et maximum de fois qu'une occurrence d'entité participe à l'association ; les valeurs possibles sont 0, 1, n

⇒ *Exemple* :

- un étudiant est inscrit au moins à 1 cours et au plus à n cours ⇒ cardinalité 1,n
- dans un cours peuvent être inscrits 0 étudiant au moins et n étudiants au plus ⇒ cardinalité 0,n

Association (3/5)

Représentation d'une association



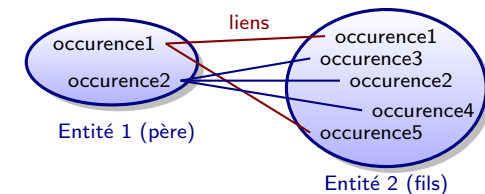
Le « type du lien » (ou de l'association) s'exprime au moyen de la valeur maximale de chaque cardinalité.

Dans l'exemple ci-joint, l'association est de type n-n (ou m-n ou plusieurs à plusieurs)

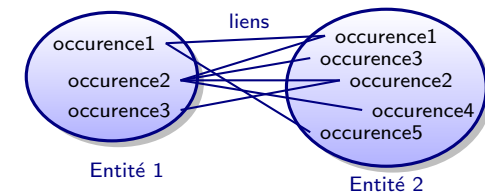
Association (4/5)

Différents types d'associations

Association 1-n



Association n-n

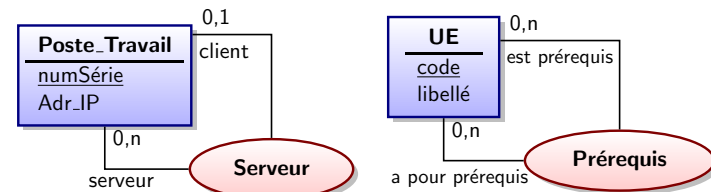


Association 1-1



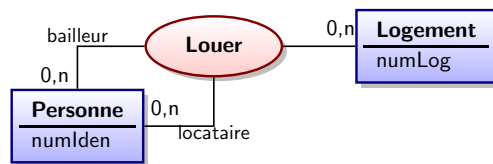
Association (5/5)

- Une **association réflexive** est une association binaire ou naire qui fait intervenir au moins deux fois la même entité.



Association 1-n

Association n-n



Entité faible

- Formalisée comme une entité mais son identification s'effectue relativement à une autre entité via une association.
- L'identifiant de l'entité faible est constituée de celui de l'entité faible et de celui de l'entité forte liée.

⇒ Exemple : On numérote les salles de cours indépendamment des bâtiments... difficile en pratique !

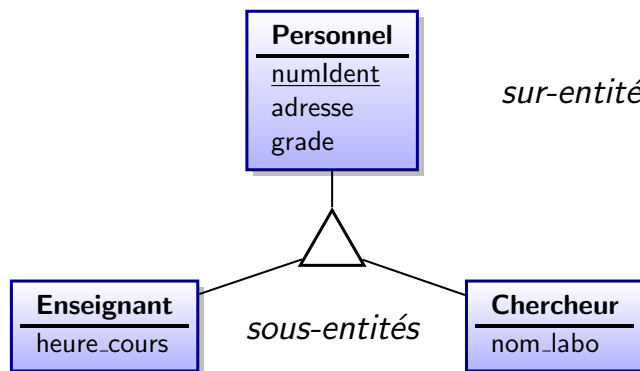


⇒ Exemple : On numérote les salles de cours relativement à un bâtiment.



Héritage

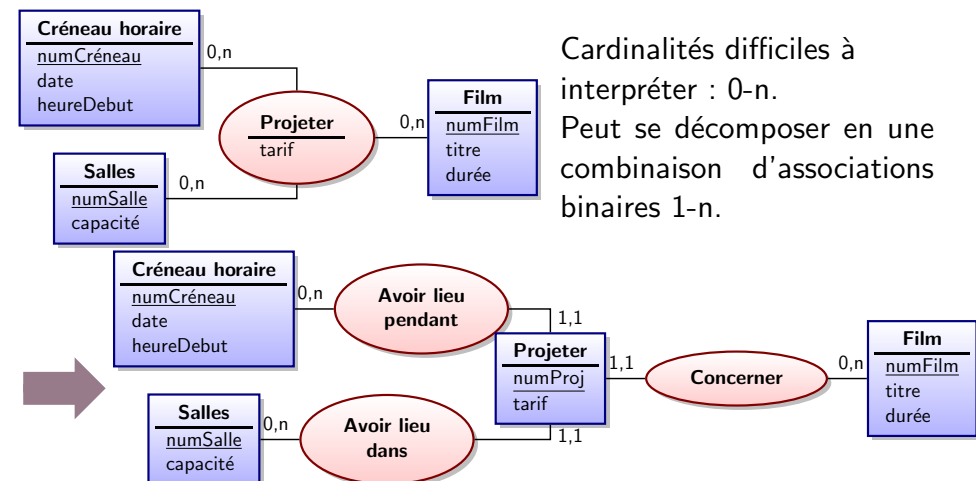
Le concept d'héritage fait état de sous-entités et d'une sur-entité.



Association d'héritage

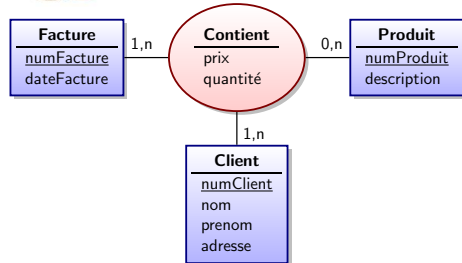
Association n-aire

- Association mettant en jeu plus de 2 entités :



Cardinalités difficiles à interpréter : 0-n.
Peut se décomposer en une combinaison d'associations binaires 1-n.

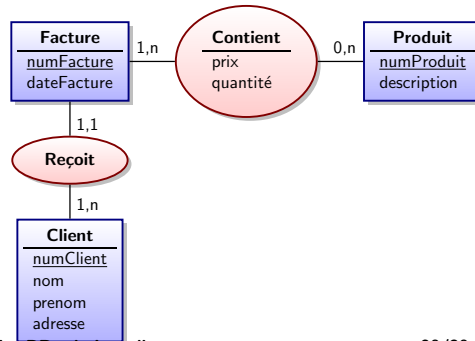
Association n-aire



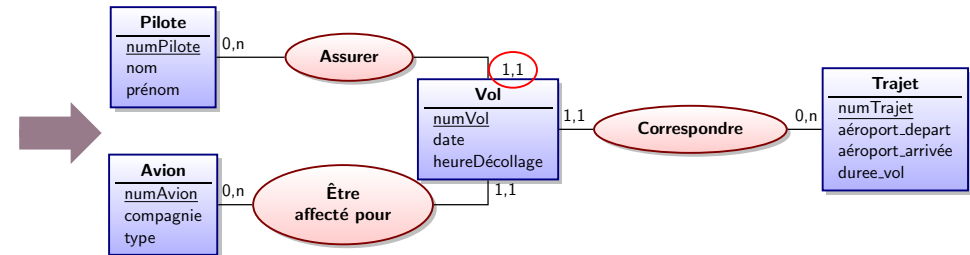
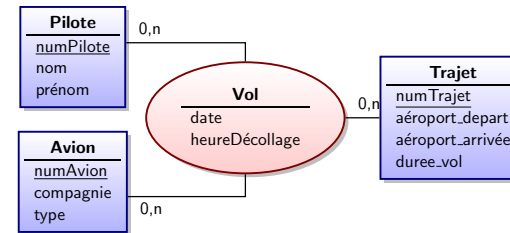
Modélisation correcte

Association n-aire inappropriée : une facture est associée à un seul client.

L'identificateur du client est répété inutilement pour chaque produit d'une même facture.

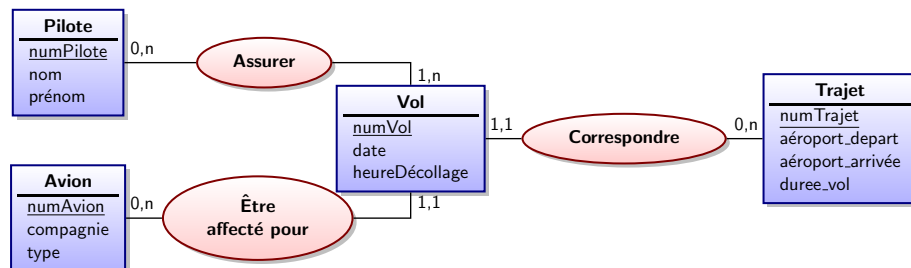


Association n-aire, exemple



Association n-aire, exemple

Modélisation correcte



Exemple

Un aéroport désire gérer les compagnies, leurs avions et les vols affrétés. Une compagnie est caractérisée par un code et un nom. Chaque avion est désigné par une immatriculation, un type, une capacité. Un avion est la propriété d'une compagnie.

Un avion peut être affrété par une compagnie à différentes dates, même plusieurs fois par jour par différentes compagnies. Pour chaque affrètement il faudra stocker le nombre de passagers transportés et le coût du vol pour la compagnie.

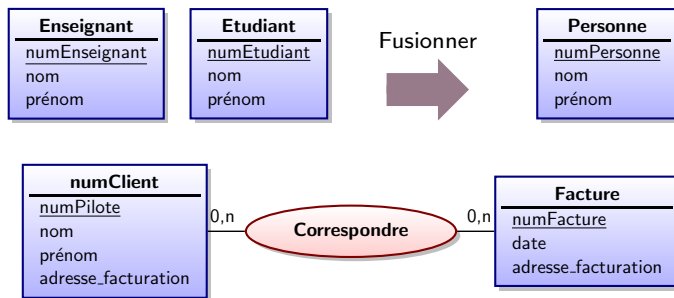
L'aéroport décide maintenant de stocker les caractéristiques de chaque type d'avion : le code de la désignation commerciale, le nombre maximum de passagers et la désignation commerciale (ex : l'A320 peut transporter au maximum 180 passagers et se dénomme « AIRBUS A320 »).

Réaliser un schéma conceptuel modélisant cette réalité.

Quelques principes pour concevoir un « bon » schéma E/R

Un bon schéma E/R limite les redondances et les incohérences : “chaque chose a une place”.

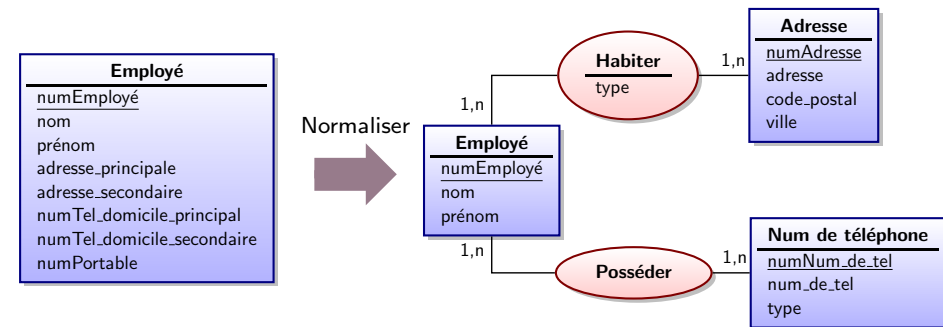
- Le nom d’une entité, d’une association ou d’un attribut doit être unique.



Exemples tirés du cours de L. Audibert

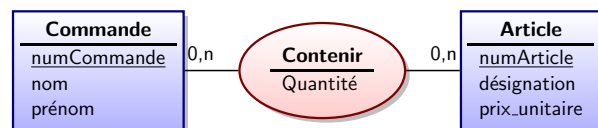
Quelques principes pour concevoir un « bon » schéma E/R

- Il faut remplacer un attribut multiple par une association et une entité supplémentaires.



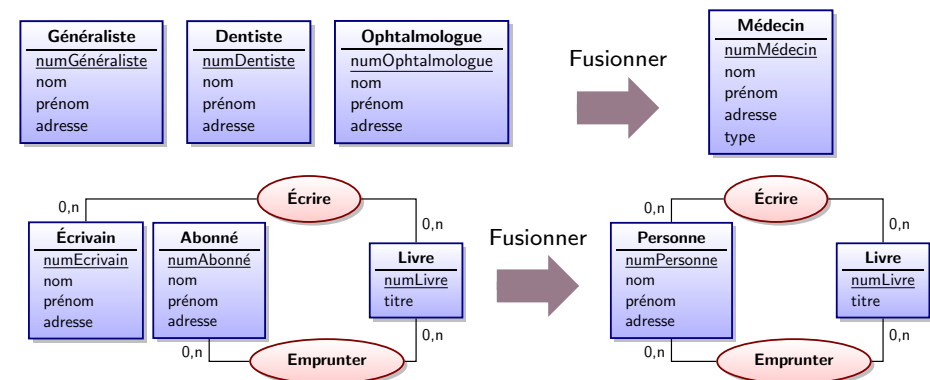
Quelques principes pour concevoir un « bon » schéma E/R

- Il faut supprimer tout attribut dérivé d’autres attributs (de la même entité ou non).



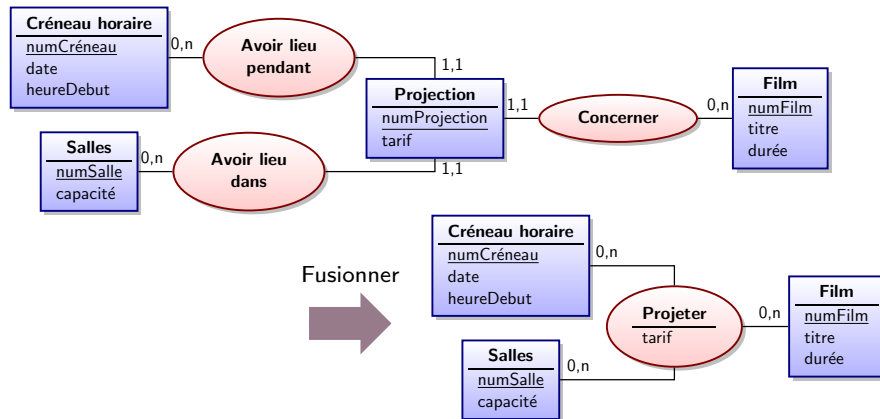
Quelques principes pour concevoir un « bon » schéma E/R

- Il faut tenter de factoriser les types-entités et les types associations quand c’est possible, éventuellement en ajoutant un nouvel attribut.



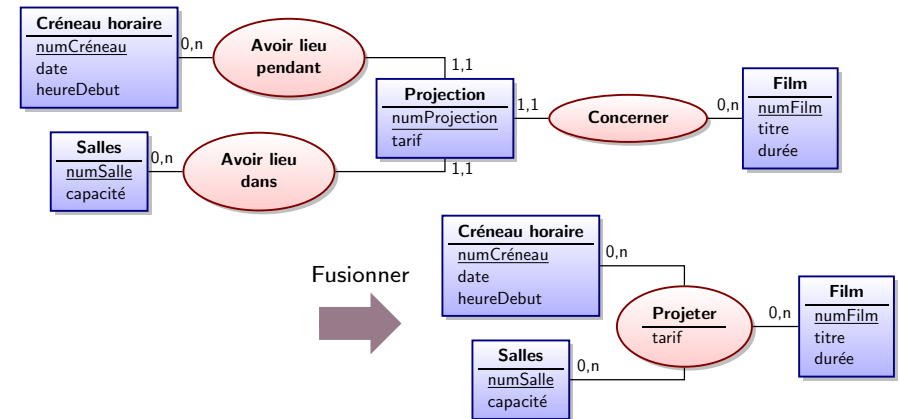
Quelques principes pour concevoir un « bon » schéma E/R

- Tout type-entité qui peut être remplacé par une association doit l'être.



Quelques principes pour concevoir un « bon » schéma E/R

- Supprimer une association dont toutes les cardinalités sont toutes 1.



Quelques principes pour concevoir un « bon » schéma E/R

- Attention aux redondances d'associations.

