



BASES DE DONNÉES-2

TI513 / TI613

L3 / L3 New

Jean-Paul DAMESTOY
Karim LAHLOU
Amar RAMADANE CHERIF
Safia OUERDANE

1

Déroulement

- Cours/TP : 22h45 (6 séances 3h30 + 1 séance 1h45)
-
- Evaluation :
 - Note de DE : 60 %
 - Notes QCM + Projet : 40 %

**Objectifs : Etre capable de bien concevoir,
implémenter, manipuler et administrer une base de
données**

2

2

Plan

- **Partie I : Conception de bases de données**
 - Modèle conceptuel de données (MCD)
 - Modèle logique de données relationnel (MLD)
 - Traduction MCD vers MLD et reverse-engineering
- **Partie II : Dépendances fonctionnelles et normalisation de bases de données**
 - Dépendances fonctionnelles
 - Formes normales et normalisation de bases de données
- **Partie III : Concepts avancés en bases de données**
 - Vues et droits d'accès
 - Transactions, résistance aux pannes et concurrence d'accès
 - Procédures stockées et déclencheurs
- **Partie IV : Interopérabilité des bases de données et applications**
 - Bases de données et Java
 - API JDBC (Java DataBase Connectivity)

3

3

Références

❑ Quelques livres utiles:

- **Database Systems: The Complete Book** : Hector Garcia-Molina
Jeffrey D. Ullman Jennifer Widom, Pearson Prentice Hall
(<https://people.inf.elte.hu/miiqaai/elektroModulatorDva.pdf>)
- **Bases de données** : Georges Gardarin, Edition Eyrolles
- **PostgreSQL par la pratique**. John C. Worsley, Joshua D. Drake
(O'Reilly – 2002 ISBN : 2-84177-211-X)
- Lien du site de Jeff ULLMAN : <http://infolab.stanford.edu/~ullman/>

4

4

INTRODUCTION

5

5

INTRODUCTION

Une **base de données** (correcte) est un ensemble
structuré,
cohérent (non contradictoire)
et **pertinent** (représentatif du problème traité et sans redondances)
d'informations fortement liées.

Il en découle la nécessité d'appliquer une **méthode de conception**
stricte,
rendant possible la conception globale d'une base de données
(c-à-d éviter d'avoir un ensemble de points de vue partiels),
permettant des maintenances aisées,
impliquant une documentation complète et des notions de modularité.

Il est impératif que les données soient séparées des traitements.

6

6

INTRODUCTION

Un **Système de Gestion de Bases de Données** (SGBD) est un logiciel qui permet d'interagir avec une base de données (définition des données, consultation, mises à jour) avec les objectifs suivants :

- > **Relation entre les données** : assurer la corrélation des données;
- > **Cohérence** : contrôle d'intégrité de la base et gestion des autorisations de modifications;
- > **Souplesse (confort), économie de place mémoire et rapidité d'accès** : langage de haut niveau SQL (Structured Query Language) qui permet des requêtes en un langage presque naturel du genre "Je veux la liste de tous les étudiants aux yeux bleus qui portent des lunettes";
- > **Sécurité** : sauvegardes périodiques;
- > **Partage des données** : gérer l'accès à des données communes pour des applications différentes;
- > **Indépendance des données** : idéalement (ce n'est pas tout à fait le cas), l'utilisateur n'aurait pas à savoir comment elles sont implantées physiquement.

7

7

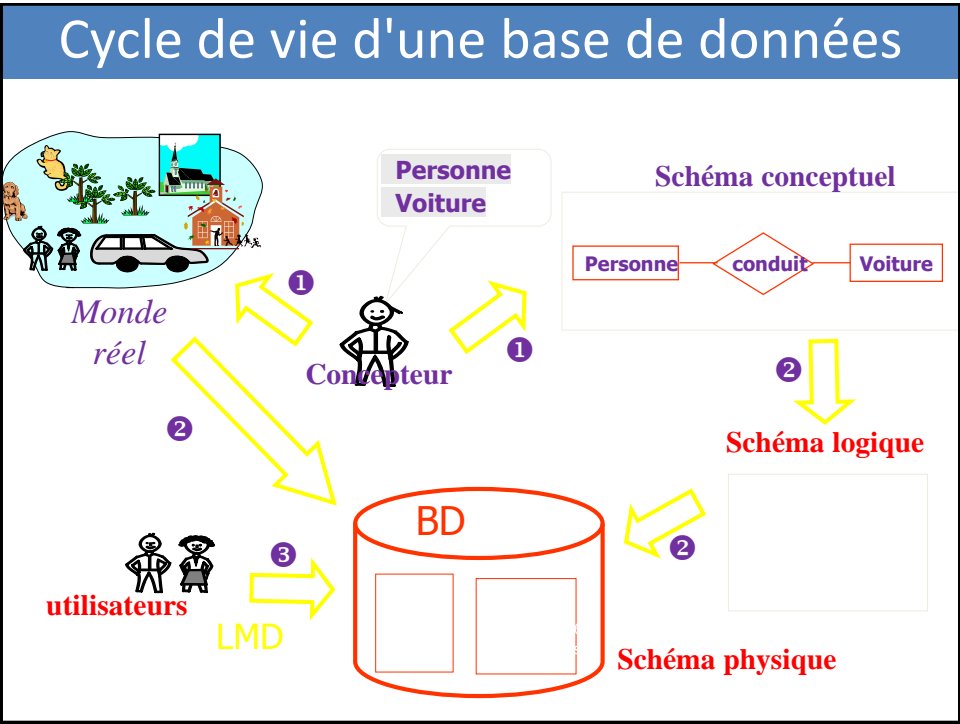
INTRODUCTION

Un **SGBD** est constitué de trois niveaux :

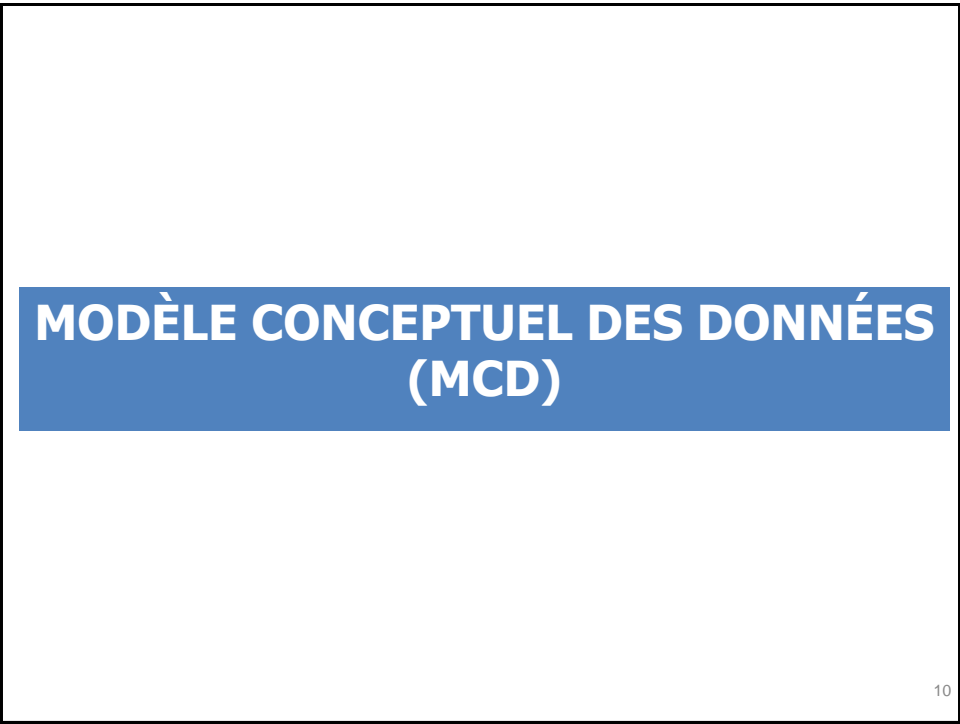
- > **Niveau physique** : implantation des données en machine;
Ce niveau étant le problème des informaticiens purs et durs.
- > **Niveau conceptuel** : représentation logique du système étudié;
C'est le cœur de nos préoccupations aujourd'hui
- > **Niveau externe** : ce que chaque utilisateur a le droit de faire et de savoir.
Ces considérations étant annexes et « faciles », elles ne seront pas abordées ici.

8

8



9



10

MCD : Introduction

Connu sous les noms :

Modèle ou schéma **Entités/Associations (E/A)**

ou

Modèle **Conceptuel de Données (MCD, méthode MERISE)**

Il existe aussi d'autres normes qui sont utilisées, comme le diagramme de classe UML.

En entreprise : on s'adapte à sa tradition.

11

11

MCD : Introduction

-> Cette méthode s'applique quelque soit le type de SGBD.

-> Elle est non seulement vitale pour les gros problèmes, mais aussi très utile pour cerner les petits problèmes que l'on croit avoir compris (programmation).

-> Elle est totalement indépendante du langage de programmation et de son implantation physique.

=> IL S'AGIT D'UNE **ETAPE CONCEPTUELLE** DANS LAQUELLE ON RECENSE LES **INTERACTIONS LOGIQUES ENTRE LES DONNÉES.**

12

12

MCD : Recensement des informations

Il s'agit d'élaborer un **dictionnaire des informations** à prendre en compte.

C'est une démarche sans méthode stricte ni modèle réellement définis :

=> on récolte de l'information et on classe.

Deux moyens :

-> **Discuter avec les responsables et les intervenants** du domaine à considérer

=> savoir faire parler les gens ET LES ECOUTER

(Technique de conduite d'entretiens)

-> **Lire des documents** : savoir lire entre les lignes;

Astuce : A chaque affirmation, se poser deux questions :

-> et alors ?

-> et sinon ?

Remarque : A chaque étape significative, il faut rédiger un document récapitulatif et, bien que cela ne soit pas toujours facile, essayer de le faire signer par la personne qui a fourni l'information ("Machin à Truc pour accord ...") et signer soi-même.

13

13

MCD : Recensement des informations

On obtient alors une **liste d'informations** qu'il va falloir épurer :

-> On précisera les **polysémies** ("masse" signifie "poids" ou "gros marteau" ?);

-> On éliminera les **synonymies**;

-> On rencontrera des **informations élaborées** : il faudra alors énoncer la règle d'élaboration (pour être sûr qu'il s'agit effectivement d'informations élaborables);

on conservera, ou non, ces informations selon l'urgence et la fréquence de leur emploi (on va peut-être conserver les informations nécessitant 3 heures de traitement !).

-> On gardera les **informations élémentaires** (non élaborées) dans le modèle E/A.

=> On attaque donc le modèle E/A avec une liste significative et épurée d'informations à prendre en compte.

14

14

MCD : Définitions

Entité :

quelque chose de **générique** (pouvant prendre un ensemble de valeurs) dotée d'une existence propre. Elle peut être de deux types :

concrète : un livre, une personne, une pièce mécanique,...

abstraite : généralement l'explication d'un code.

Remarque : un livre est une entité générique dans le sens où il peut s'agir de

"Les Misérables" de Victor Hugo, de "Germinal" de Zola,...

15

15

MCD : Définitions

Association :

représentation d'une association entre entités. On peut avoir à faire à :

-> 1 seule entité : un étudiant est en binôme avec un étudiant;

-> 2 entités : un étudiant a emprunté un livre;

-> 3 entités : un étudiant a emprunté un livre dans une certaine bibliothèque (si on tient compte de plusieurs bibliothèques).

Au-delà de 3 entités, il faut essayer de casser l'association en plusieurs petites :

"Un professeur fait cours dans telle matière dans telle salle pour telle promotion »

devient

"cette matière est enseignée dans telle salle pour telle promotion »

et

"un professeur fait cours dans telle salle"

à condition qu'ayant la salle et la promotion on ait la matière.

Le nombre d'entités est la *dimension de l'association*.

16

16

MCD : Définitions

Attribut :

donnée élémentaire relative à une entité ou une association, tel le nom de l'étudiant par exemple.

Instance (ou occurrence) d'une entité:

combinaison unique de valeurs prises par les attributs d'une entité. Deux instances d'étudiants : Jacques Durand et Georges Martin. Autrement dit, une entité est une machine à cake, une instance est un cake. Encore dit d'une autre manière :

une entité est un formulaire vide, une instance un formulaire rempli.

Identifiant (ou clé primaire) :

un ou plusieurs attributs (souvent artificiels comme un numéro) tels qu'il existe une bijection entre l'occurrence de l'entité et l'identifiant.

Je m'explique : la valeur de l'identifiant ne peut se retrouver qu'une seule fois parmi les occurrences de l'entité.

Occurrence d'une association :

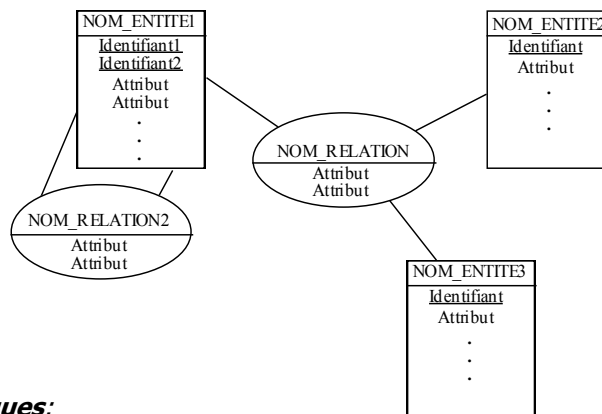
combinaison unique des occurrences intervenant dans l'association et de ses attributs propres. Ainsi nous avons trois occurrences :

- Martin a emprunté "Les Misérables" le 15-07-89;
- Durand a emprunté « Le seigneur des anneaux" le 25-12-93;
- Martin a emprunté "Les Misérables" le 07-08-94;

17

17

MCD : Symbolisme



Remarques:

- l'entité NOM_ENTITE1 a un identifiant composé de la réunion de deux attributs.
- NOM_RELATION2 implique deux fois une seule entité.
- choisir des noms (d'entités, d'associations, d'attributs) significatifs, c-à-d éviter les termes trop généraux tels "est un" ou "a".

18

18

MCD : règles

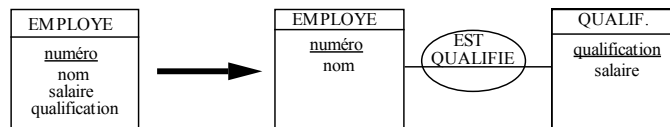
a) L'identifiant doit réellement être un identifiant; chaque entité doit avoir un identifiant.

La règle sur laquelle tout repose :

b) Un attribut est un scalaire, i.e. ce n'est pas une liste, ce n'est pas un sous objet.

chaque attribut doit une valeur pour toute occurrence, i.e. pas de case vide

c) Les dépendances transitives entre attributs sont interdites :



à condition que la qualification implique un salaire.

19

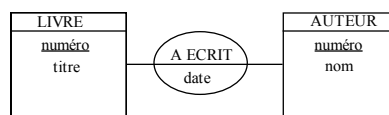
19

MCD : règles

d) Une association doit toujours mettre en relation le même nombre d'entités (il est interdit d'avoir une relation entre A, B et C qui, de temps en temps, ne relierait que A et B)

e) Si les attributs d'une association peuvent être mis dans une entité supplémentaire en relation avec l'association, le faire.

f) Si un attribut de l'association n'a de sens que pour un sous-ensemble des entités concerné, alors cet attribut fait partie du sous-ensemble concerné et non de l'association complète.



"date" ne doit pas être dans l'association : la date ne qualifie que le livre (un livre n'a été écrit qu'une fois). Par contre :



à un sens puisqu'un livre peut être édité plusieurs fois par le même éditeur à des dates différentes

20

20

MCD : Cardinalités

Elles indiquent pour **chaque couple entité-association** le **nombre minimum et maximum d'occurrences de l'association** pouvant exister à un instant quelconque pour une occurrence de l'entité.

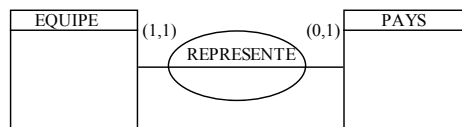
Elle se note **(mini, maxi)** sur chaque "branche" de l'association.

Mini : au moins

Maxi : au plus

Elles peuvent être du type (0,1), (0,n) (1,1) ou (1,n) mais jamais du type (n,n).

Exemple :



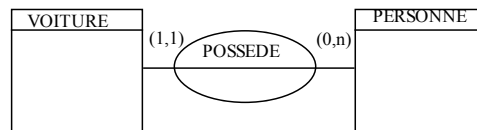
Une équipe représente **un et un seul** pays; un pays est représenté par **aucune ou une** équipe.

Remarquer l'emploi du passif pour fixer les cardinalités

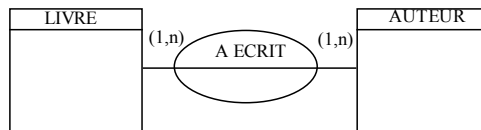
21

21

MCD : Cardinalités



Une voiture appartient à **une seule** personne, une personne peut avoir **aucune ou plusieurs** voitures.



Un livre a été écrit par **un ou plusieurs** auteurs, un auteur peut avoir écrit **un ou plusieurs** livres.

22

22

MCD : recommandations

- 1) Répertorier l'ensemble des informations.
- 2) Epurer cet ensemble.
- 3) Trouver une entité concrète.
- 4) Y attacher tous les attributs.
- 5) Déterminer l'identifiant.
- 6) Si on trouve une information qui se rattache à cette entité, mais qui n'est pas un attribut alors on a une relation et éventuellement une nouvelle entité, on repart alors à l'étape 4 avec cette nouvelle relation et/ou entité.
- 7) Essayer de retourner en 3 avec les informations restantes.
- 8) Attacher au modèle toutes les informations restantes devant y être incluses (entités abstraites: explication de codage).
- 9) Toute information élaborée est-elle alors facilement élaborable ?
- 10) Fixer les cardinalités.
- 11) Les traitements prévus vont-ils pouvoir s'effectuer ?

Remarque très importante : Vérifier et revérifier le modèle E/A car une fois celui-ci réalisé, le reste du travail se fait quasi automatiquement.

23

23

MODÈLE LOGIQUE DE DONNÉES (MLD)

24

24

MLD ou le modèle relationnel

Le **modèle relationnel (ou MLD en Merise)** et les systèmes qui les utilisent s'imposent aujourd'hui sur le marché des bases de données : Oracle, SQLServer, Postgresql, MySQL, Access, ...

C'est un modèle qui privilégie les RELATIONS.

La transformation modèle E/A au modèle relationnel s'appuie sur

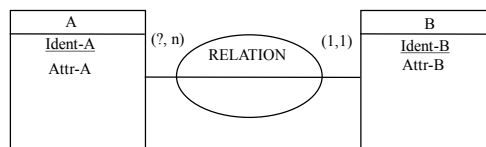
- > les associations,
- > leurs cardinalités,
- > sur le nombre d'entités concernées.

25

25

Une ou deux entités

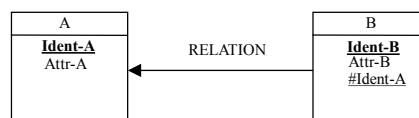
a) cardinalités $(?,n)$ - $(1,1)$



Connaissant B, je peux connaître directement le A qui est en relation avec lui puisqu'il n'y en a qu'un;
par contre, connaissant A, pour connaître tous les B qui lui sont associés il faut que je parcoure toutes les occurrences de B.

Modélisation : A va donner une copie de son identifiant à B qui en fera un de ses attributs. On dit que A est le propriétaire de la relation et B le membre.

La relation est alors symbolisée par une flèche.



26

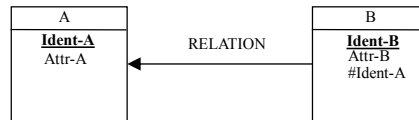
26

Une ou deux entités

On peut en déduire le schéma relationnel avec une **représentation algébrique**, de la forme :

Nom_de_l'entité(identifiant, attribut1, attribut2, #clé_étrangère)

Par exemple :



devient

A(Ident-A, Attr-A)

B(Ident-B, Attr-B, #Ident-A)

27

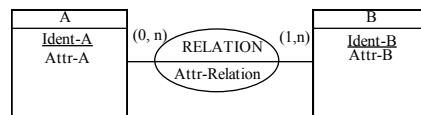
27

Une ou deux entités

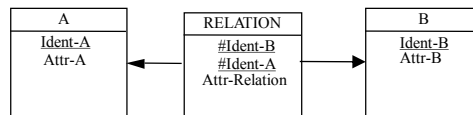
b) cardinalités (?n)-(0,1)

Idem.

c) cardinalités (?n)-(?,n)



On ne peut plus passer directement de B vers A. La relation devient une *entité-lien*. Elle conserve ses attributs et reçoit la copie des identifiants des entités concernées. L'identifiant de la relation devient alors l'ensemble de ses attributs.



A(Ident-A, Attr-A)

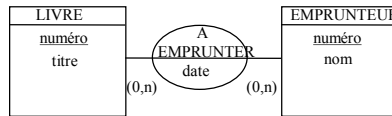
RELATION(#Ident-B, #Ident-A, Attr-Relation)

B(Ident-B, Attr-B)

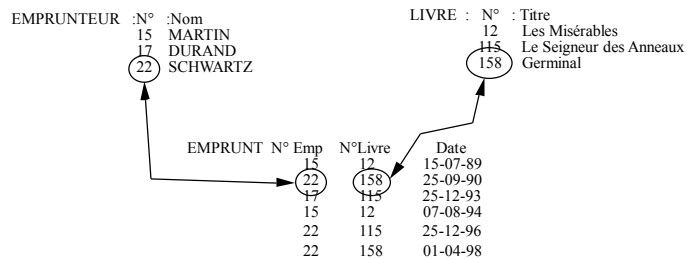
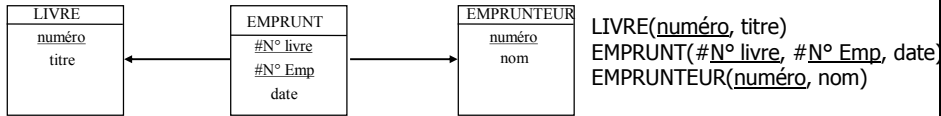
28

28

Une ou deux entités



Généralement, on essaye de mettre le substantif du verbe de la relation

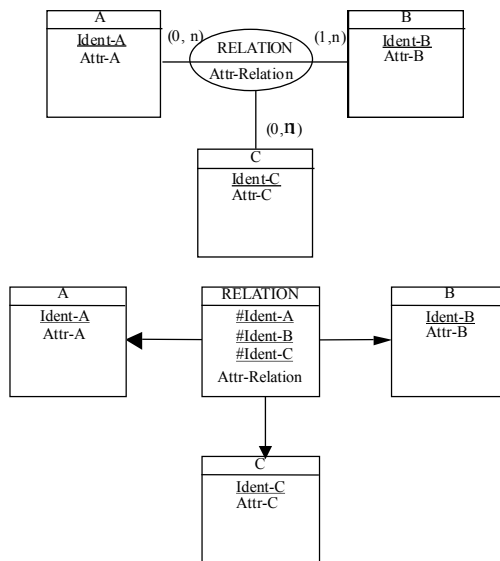


29

29

Relation à plus de deux entités

On procède comme pour le cas à cardinalité (? , n) - (? , n); c-à-d la relation devient une entité-lien et récupère les identifiants des entités concernées.



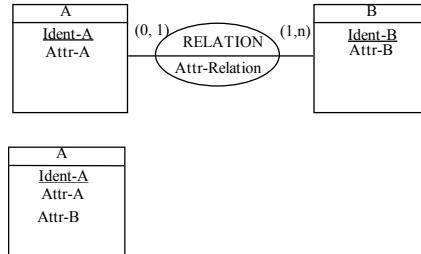
30

30

Cas de simplification

Dans le cas suivant où une entité est toute petite, on peut s'autoriser à inclure les attributs de cette entité dans une autre.

Ex : B contient le nom marital, bien que tous les A n'en n'aient pas, on va préférer laisser à vide souvent cet attribut plutôt que d'avoir à gérer deux entités.



Implantation :

Le modèle relationnel établi, chaque entité et entité-lien devient une table dont la structure de ses éléments est constituée de l'identifiant et des attributs de l'entité.

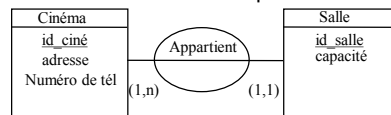
31

31

31

Entité Faible

Il existe des cas où une entité ne peut exister qu'en étroite association avec une autre, et est identifiée relativement à cette autre entité. On parle alors **d'entité faible**.



Il est difficilement imaginable de représenter une salle sans qu'elle soit rattachée à son cinéma. C'est en effet au niveau du cinéma que l'on va trouver quelques informations générales comme l'adresse ou le numéro de téléphone.

On peut considérer alors qu'il est beaucoup plus naturel **de numéroté les salles par un numéro interne à chaque cinéma**.

La clé d'identification d'une salle :

- la **clé de Cinéma**, qui indique dans quel cinéma se trouve la salle ;
- le **numéro de la salle** au sein du cinéma.

-> l'entité salle ne dispose pas d'une *identification absolue*, mais d'une *identification relative* à une autre entité. Bien entendu cela force la salle à toujours être associée à **un et un seul** cinéma.

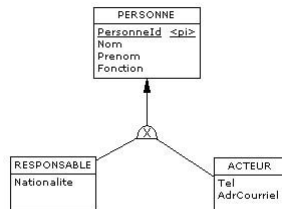
32

32

Héritage

Les **sous-classes** RESPONSABLE et ACTEUR feront l'objet elles aussi d'entités-types, associées à l'entité-type PERSONNE par une **relation d'héritage**. Les entités-types RESPONSABLE et ACTEUR possèdent les mêmes attributs que les **sous-classes**.

Exemple d'héritage avec un MCD



Il y a plusieurs possibilités pour implémenter un héritage → au programme de Advanced DataBases en M1

33

33

A VOUS DE JOUER !
TD1

34

34