

Bases de Données

Amélie Gheerbrant



Université Paris Diderot

UFR Informatique

Laboratoire d'Informatique Algorithmique : Fondements et Applications

amelie@liafa.univ-paris-diderot.fr

29 septembre 2014

Rappel : la base de données "Air France"

PILOTE

PLNUM	PLNOM	PLPRENOM	VILLE	SALAIRE
1	MIRANDA	SERGE	PARIS	21000
2	LETHANH	NAHN	TOULOUSE	21000
3	TALADOIRE	GILLES	NICE	18000
4	BONFILS	ELIANE	PARIS	17000
5	LAKHAL	LOTHI	TOULOUSE	19000
6	BONFILS	GERARD	PARIS	18000
7	MARCENAC	PIERRE	NICE	17000
8	LAHIRE	PHILIPPE	LYON	15000
9	CICCHETTI	ROSINE	NICE	18000
10	CAVARERO	ANNIE	PARIS	20000

AVION

AVNUM	AVNOM	CAPACITE	LOCALISATION
1	A300	300	NICE
2	A310	300	NICE
3	B707	250	PARIS
4	A300	280	LYON
5	CONCORDE	160	NICE
6	B747	460	PARIS
7	B707	250	PARIS
8	A310	300	TOULOUSE
9	MERCURE	180	LYON
10	CONCORDE	160	PARIS

VOL

VOLNUM	PLNUM	AVNUM	VILLEDEP	VILLEARR	HEUREDEP	HEUREARR
100	1	1	NICE	TOULOUSE	11:00:00	12:30:00
101	1	8	PARIS	TOULOUSE	17:00:00	18:30:00
102	2	1	TOULOUSE	LYON	14:00:00	16:00:00
103	5	3	TOULOUSE	LYON	18:00:00	20:00:00
104	9	1	PARIS	NICE	06:45:00	08:15:00
105	10	2	LYON	NICE	11:00:00	12:00:00
106	1	4	PARIS	LYON	08:00:00	09:00:00
107	8	4	NICE	PARIS	07:15:00	08:45:00
108	1	8	NANTES	LYON	09:00:00	15:30:00
109	8	2	NICE	PARIS	12:15:00	13:45:00
110	9	2	PARIS	LYON	15:00:00	16:00:00
111	1	2	LYON	NANTES	16:30:00	20:00:00
112	4	5	NICE	LENS	11:00:00	14:00:00
113	3	5	LENS	PARIS	15:00:00	16:00:00
114	8	9	PARIS	TOULOUSE	17:00:00	18:00:00
115	7	5	PARIS	TOULOUSE	18:00:00	19:00:00

Rappel : base de données

- ▶ Une base de données se compose de plusieurs relations.
- ▶ L'information qui concerne une application est divisée en parties (tables), chaque relation stockant une partie de l'information
 - ▶ pilote : stocke l'information sur les pilotes
 - ▶ avion : stocke l'information sur les avions
 - ▶ vol : stocke l'information sur les vols (dont le pilote et l'avion du vol)
- ▶ Stocker toute l'information dans une seule relation comme
airfrance(plenum, plnom, plprenom, ville, salaire, avnum, avnom, capacité, localisation, volnum, villedep, villearr, heuredep, heurearr)

est possible mais pas souhaitable :

entraîne répétition de l'information et valeurs de données nulles

Rappel : la table "Avion"

<u>AVNUM</u>	AVNOM	CAPACITE	LOCALISATION
1	A300	300	NICE
2	A310	300	NICE
3	B707	250	PARIS
4	A300	280	LYON
5	CONCORDE	160	NICE
6	B747	460	PARIS
7	B707	250	PARIS
8	A310	300	TOULOUSE
9	MERCURE	180	LYON
10	CONCORDE	60	PARIS

un **tuplet** de la relation AVION
 =
 une **ligne** de la table AVION

Une **donnée numérique** de la relation AVION
 =
 une **valeur de type « entier »** de la table AVION

un **attribut** de la relation AVION
 =
 une **colonne** de la table AVION

L'**extension** de la relation AVION
 =
 La **table** AVION

Rappel : Le modèle relationnel

- ▶ une base de données se compose de **tables** (**relations**)
- ▶ les colonnes de chaque table sont nommées par des **attributs**
- ▶ chaque attribut est associé à un **domaine** (ensemble de valeurs admissibles)
- ▶ les données dans chaque table sont constituées par l'ensemble des rangées (**tuples**) fournissant des valeurs pour les attributs
- ▶ pas d'ordre sur les tuples (relations = ensembles non ordonnés)
- ▶ (en général) ordre sur les valeurs des attributs dans un tuple

Clefs primaires

- ▶ La **clé primaire** d'une relation R est l'attribut ou l'ensemble d'attributs (avec le moins d'attributs possible) qui identifie de manière unique chaque tuple de la relation.
- ▶ Exemple :
PLNUM est la clé primaire de PILOTE car (on suppose que) chaque pilote possède un numéro unique.
- ▶ La clé primaire est **soulignée**.
- ▶ Il n'y a qu'**une seule** clé primaire par relation.
- ▶ La valeur des attributs clefs primaires ne peut **jamais** être **nulle** dans aucun tuple de la relation.

Clefs étrangères

On a :

$VOL.AVNUM \subseteq AVION.AVNUM$
car toute valeur de
VOL.AVNUM est dans
AVION.AVNUM.

VOL

VOLNUM	PLNUM	AVNUM	VILLEDEP	VILLEARR	HEUREDEP	HEUREARR
100	1	1	NICE	TOULOUSE	11:00:00	12:30:00
101	1	8	PARIS	TOULOUSE	17:00:00	18:30:00
102	2	1	TOULOUSE	LYON	14:00:00	16:00:00
103	5	3	TOULOUSE	LYON	18:00:00	20:00:00
104	9	1	PARIS	NICE	06:45:00	08:15:00
105	10	2	LYON	NICE	11:00:00	12:00:00
106	1	4	PARIS	LYON	08:00:00	09:00:00
107	8	4	NICE	PARIS	07:15:00	08:45:00
108	1	8	NANTES	LYON	09:00:00	15:30:00
109	8	2	NICE	PARIS	12:15:00	13:45:00
110	9	2	PARIS	LYON	15:00:00	16:00:00
111	1	2	LYON	NANTES	16:30:00	20:00:00
112	4	5	NICE	LENS	11:00:00	14:00:00
113	3	5	LENS	PARIS	15:00:00	16:00:00
114	8	9	PARIS	TOULOUSE	17:00:00	18:00:00
115	7	5	PARIS	TOULOUSE	18:00:00	19:00:00

AVION

AVNUM	AVNOM	CAPACITE	LOCALISATION
1	A300	300	NICE
2	A310	300	NICE
3	B707	250	PARIS
4	A300	280	LYON
5	CONCORDE	160	NICE
6	B747	460	PARIS
7	B707	250	PARIS
8	A310	300	TOULOUSE
9	MERCURE	180	LYON
10	CONCORDE	160	PARIS

VOL.AVNUM une **clef étrangère** de la table VOL, car :

- ▶ toutes les valeurs de VOL.AVNUM sont également des valeurs de AVION.AVNUM
- ▶ AVION.AVNUM est la clef primaire de la table AVION

Rappel

La **conception** de bases de données relationnelles

1. Point de départ : description **informelle** d'une application potentiellement très compliquée
2. Abstraction et optimisation du cahier des charges (**modélisation**) sous une forme **graphique** et **compacte**
3. Création d'entités comprises par le système (extraction des **relations** de la base de données)
4. Optimisation des relations (**normalisation**)
5. Manipulation de la base avec SQL.

Introduction

- ▶ Réfléchir avant de créer une base de données :
 - ▶ Faire un diagramme du schéma de la base
 - ▶ Transformer le diagramme en un schéma relationnel
- ▶ Intérêt :
 - ▶ Représentation **graphique** et intuitive :
 - ▶ Comprendre facilement
 - ▶ Visualiser rapidement les erreurs
 - ▶ Effectuer des corrections rapides
 - ▶ Permet de discuter avec des non-informaticiens
 - ▶ Constitue le début de la documentation de la BD

Introduction

- ▶ Plusieurs formalismes de diagrammes existent :
 - ▶ les schémas entités-associations E-A
 - ▶ les diagrammes de classes UML
 - ▶ des méthodes (Merise, ...)

tous basés sur les diagrammes E-A
(E-R en anglais : Entity-Relationship)

- ▶ Démarche :
 1. Faire le diagramme EA
MCD : Modèle Conceptuel de Données
 2. En déduire le schéma relationnel
MLD : Modèle Logique de Données
 3. Construire la BD sur machine grâce à un SGBD
 4. Manipuler la BD grâce à SQL

Les concepts

- ▶ **Entité** : un objet qui existe dans le monde réel et possède une identité

Exemples :

- ▶ le pilote Annie Cavarero, matricule 10, basé à Paris, dont le salaire est 20000

- ▶ **Association** : une « relation » entre deux ou plusieurs entités
- Exemples :

- ▶ le pilote 10 pilote le vol 105
- ▶ le vol 105 est assuré par l'avion 2

- ▶ **Attribut** : propriété d'une **entité** ou d'une **association** qui prend ses valeurs dans un domaine (string, [1..10], etc.)

Exemples :

- ▶ le N° du pilote Cavarero est 10
- ▶ la ville de Cavarero est Paris

Classes d'entités et d'associations

- ▶ Une **classe d'entités** est un ensemble d'entités similaires, ayant les mêmes **attributs**.
- ▶ Une **classe d'associations** est un ensemble d'associations entres les entités d'une ou de plusieurs classes.



Par abus de langage, on utilise souvent entité (association) à la place de classe d'entité (d'association).

Les entités

- ▶ Définition : Une **classe d'entités** est un ensemble d'éléments appartenant à une même classe et partageant des propriétés communes.
- ▶ Exemple : la classe d'**entités** Aéroport

Aéroport
<u>Nom</u>
Localisation

- ▶ Aéroport = **nom** de la classe d'entités
 - ▶ Nom, Localisation = **attributs** de Aéroport
 - ▶ Nom = **identificateur** de Aéroport
 - ▶ l'aéroport d'Orly appartient à la class d'entité Aéroport
- ▶ Chaque classe d'entités est traduite par une table pour obtenir le schéma relationnel : Aeroport(Nom, Localisation)

Les attributs et les clefs

- ▶ Attribut = sorte de "sous entité" décrivant plus finement l'entité
- ▶ Chaque entité a un **identifiant** :
 - ▶ un (ou plusieurs) attribut(s) qui permettent d'**identifier de manière unique toutes ses instances**
 - ▶ appelé **clef primaire** (CP)
 - ▶ **souligné** dans l'entité
 - ▶ Exemple : le nom d'un aéroport suffit à l'identifier

Aéroport
<u>Nom</u>
Localisation

- ▶ L'identifiant d'une entité est traduit comme la CP de la table correspondante : Aeroport(Nom, Localisation)

Clefs primaires : exemples

Appartement	Personne
<u>N°Bâtiment</u>	<u>N°SécuritéSoc</u>
<u>N°Appartement</u>	Nom
Taille	Prénom
Etage	DateNaissance
	LieuNaissance

- Pour pouvoir identifier de manière unique chaque appartement, il faut à la fois les n° du bâtiment et de l'appartement.
 - ⇒ Schéma relationnel : Appartement(N°Bâtiment, N°Appartement, Taille, Etage)
- Parfois on ajoute un attribut fictif (souvent un numéro) uniquement pour servir de clé à l'entité considérée. Ici : le numéro de sécurité sociale.
 - ⇒ Schéma relationnel : Personne(N°SécuritéSoc, Nom, Prénom, DateNaissance, LieuNaissance))

Identificateurs

► Identificateur d'entité

- un ou plusieurs attributs permettant d'identifier une entité dans une classe d'entités
- Exemple : PLNUM pour PILOTE

Attention : l'identificateur doit être **minimal**
(on ne doit pas pouvoir lui enlever un attribut sans qu'il cesse d'être un identificateur)

► Identificateur d'association

- un identificateur composé de tous les identificateurs d'entités reliées par l'association
- exemple : PLNUM,VOLNUM pour Effectue

Les concepts

- ▶ **Entité** : un objet qui existe dans le monde réel et possède une identité

Exemples :

- ▶ le pilote Annie Cavarero, matricule 10, basé à Paris, dont le salaire est 20000

- ▶ **Association** : une « relation » entre deux ou plusieurs entités

Exemples :

- ▶ le pilote 10 pilote le vol 105
- ▶ le vol 105 est assuré par l'avion 2

- ▶ **Attribut** : propriété d'une **entité** ou d'une **association** qui prend ses valeurs dans un domaine (string, [1..10], etc.)

Exemples :

- ▶ le N° du pilote Cavarero est 10
- ▶ la ville de Cavarero est Paris

Classes d'entités et d'associations

- ▶ Une **classe d'entités** est un ensemble d'entités similaires, ayant les mêmes **attributs**.
- ▶ Une **classe d'associations** est un ensemble d'associations entres les entités d'une ou de plusieurs classes.



Par abus de langage, on utilise souvent entité (association) à la place de classe d'entité (d'association).

"Crowfoot" : cardinalités d'une classe d'associations

Un intervalle $[min,max]$ indique pour une classe d'entités C et une classe d'associations A , le nombre d'associations de type A qu'une entité de C peut avoir avec d'autres entités.

○ se lit "zéro"

|

se lit "un"

➤

se lit "n" (i.e., un nombre arbitrairement grand)

valeurs possibles

D'où:

⊞

se lit "de zéro à 1"

⊞

se lit "1 et 1 seul"

⊞➤

se lit "de zéro à n"

⊞➤

se lit "de 1 à n"

intervalles possibles

Exemple

Schéma E-A :

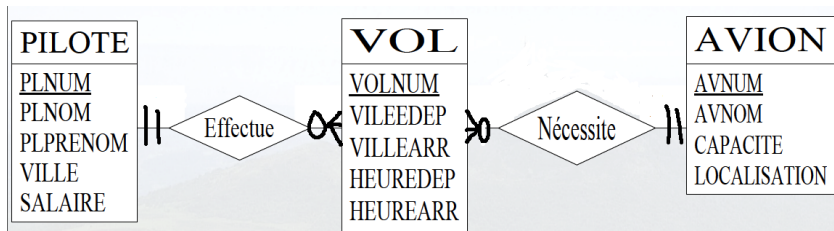
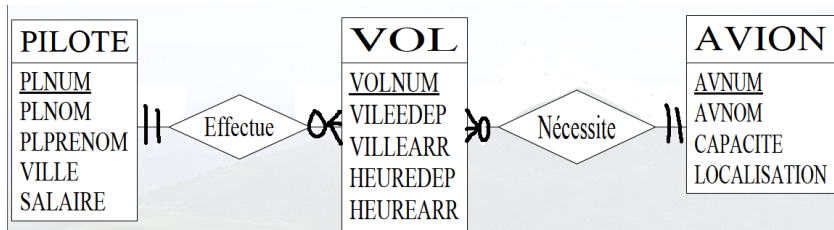


Schéma relationnel :

- ▶ PILOTE (PLNUM, PLNOM, PLPRENOM, VILLE, SALAIRE)
- ▶ AVION (AVNUM, AVNOM, CAPACITE, LOCALISATION)
- ▶ VOL (VOLNUM, PLNUM, AVNUM, VILLEDEP, VILLEARR, HEUREDEP, HEUREARR)

Exemple (plus d'exemples au tableau)

Schéma E-A :

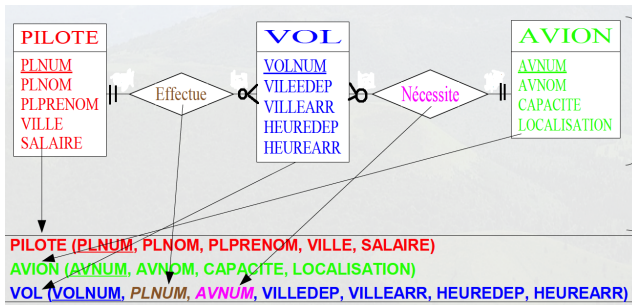


Lecture des **cardinalités** du schéma E-A (notation crowfoot) :

- ▶ À un pilote correspond 0 ou plusieurs vols.
- ▶ À un vol correspond 1 et un 1 seul pilote.
- ▶ À un vol correspond 1 et 1 seul avion.
- ▶ À un avion correspond 0 ou plusieurs vols.

Schémas E-A et relationnels

- ▶ Principe intuitif du passage EA à relationnel :
 - ▶ les entités deviennent des tables
 - et en fonction des cardinalités :
 - ▶ certaines associations sont représentées par des entités
 - ▶ certaines associations sont représentées comme clés étrangères (en italique ci-dessous) de tables dans d'autres tables



Passage aux tables

- ▶ **Entité** : chaque classe d'entités devient une table
- ▶ **Relation n-n** (cardinalité max = n des 2 côtés de la relation) : la relation devient une table dont la clef primaire est la combinaison des clefs des entités participantes
- ▶ **Relation 1-n** (cardinalité max = 1 d'un côté, n de l'autre) : la clef primaire de la relation du côté 1 est reportée comme clef étrangère de la relation du côté n
- ▶ **Relation 1-1** (cardinalité max = 1 des deux côtés) : la clef primaire de l'une des relations est reportée comme clef étrangère de l'autre (de 1-1 vers 0-1 si les cardinalités sont différentes, on choisit un côté sinon)
- ▶ **Règle générale** : l'identificateur de l'entité (ou de la relation) devient la clef primaire de la table associée

Les entités fortes / faibles

- ▶ Une **entité forte** n'a pas besoin d'une autre entité pour exister.

Exemples : PILOTE, AVION, VOL

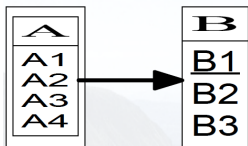
- ▶ Une **entité faible** :

- ▶ a besoin d'une autre entité pour exister,
- ▶ a une CP composée avec celle de l'autre entité.

Exemple : ROMAN et ESSAI par rapport à LIVRE (avec CP ISBN)

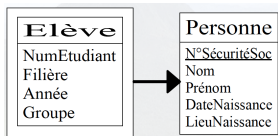
- ▶ Une exemple d'entité faible : l'**héritage**
 - ▶ La CP de l'entité faible est la même que celle de l'autre entité.
 - ▶ L'ensemble des valeurs de la CP de l'entité faible est un sous ensemble de l'ensemble des valeurs de la CP de l'autre entité.
 - ▶ se dessine avec un double rectangle.

Entités faibles : héritage



- ▶ Notation : avec une flèche de l'entité faible à l'autre entité.
- ▶ Traduction en relationnel :
 - ▶ B(B1, B2, B3)
 - ▶ A(B1, A1, A2, A3, A4)
 - ▶ où B1 est clé étrangère de A
- ▶ La CP de l'entité faible est celle de l'autre entité.
- ▶ Si A est l'entité faible et B l'autre entité, alors on dit que :
 - ▶ il existe une association d'héritage entre A et B
 - ▶ toute instance de A est "est une instance de" B
 - ▶ tout objet A "est un" B

Héritage : exemple



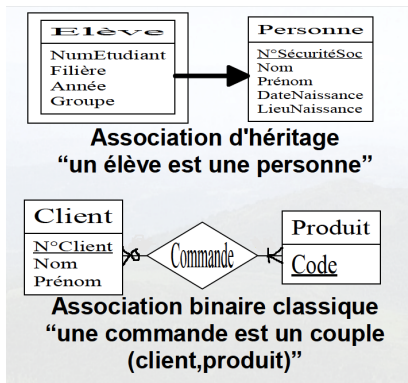
Elève = entité faible

(ensemble des N° de s.s. des élèves contenu dans celui des personnes)

- ▶ **Attributs de Elève** : **N°SécuritéSoc (CP)**, NumEtudiant, Filière, Année, Groupe
- ▶ **Attributs de Personne** : N°SécuritéSoc (CP), Nom, Prénom, DateNaissance, LieuNaissance
- ▶ **Schéma relationnel** :
 - ▶ PERSONNE(N°SécuritéSoc, Nom, Prénom, DateNaissance, LieuNaissance)
 - ▶ ELEVE(N°SécuritéSoc, NumEtudiant, Filière, Année, Groupe)
 - ▶ et **N°SécuritéSoc** est **clé étrangère** de ELEVE

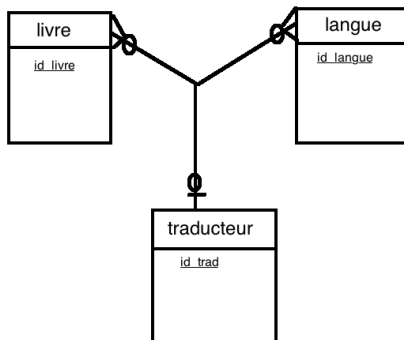
Types d'associations

- ▶ Association **binaire** classique
- ▶ Association d'**héritage**



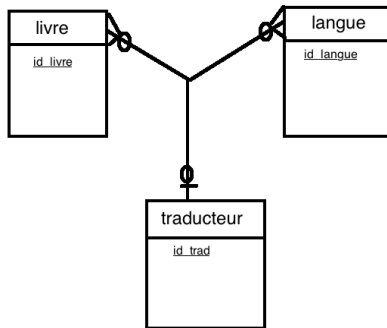
Il y a aussi des associations **réflexives** et des associations **ternaires**.

Associations n-aires



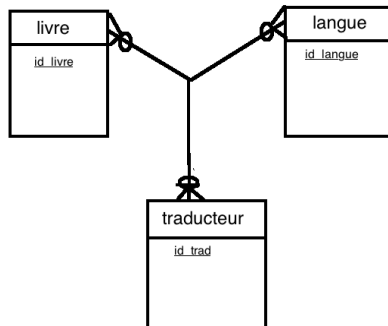
- ▶ À 1 livre et 1 langue correspond 0 ou 1 traducteur.
- ▶ À 1 langue et 1 traducteur correspond 0 ou plusieurs livres.
- ▶ À 1 livre et 1 traducteur correspond 0 ou plusieurs langues.

Associations n-aires : passage aux tables



- ▶ Chaque entité devient une table : `Livre(id_livre)`, `Langue(id_langue)`, `Traducteur(id_trad)`.
- ▶ La relation devient une table : `Traduction(id_livre, id_langue, id_trad)` ⇒ **minimalité de la clef!**

Associations n-aires : passage aux tables

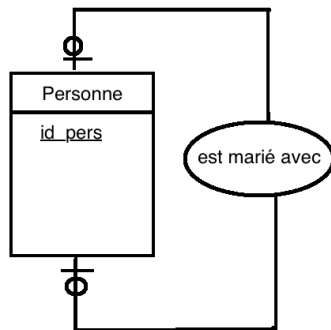


Un livre peut être traduit par plusieurs traducteurs dans une même langue : Traduction(id_livre, id_langue, id_trad)

⇒ **minimalité de la clef !**

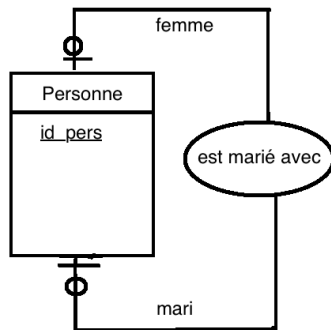
Associations réflexives

Relation 1-1 réciproque

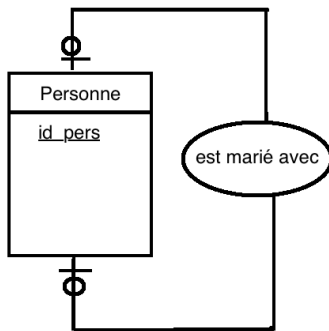


Associations réflexives

Relation 1-1 réciproque avec annotation des rôles



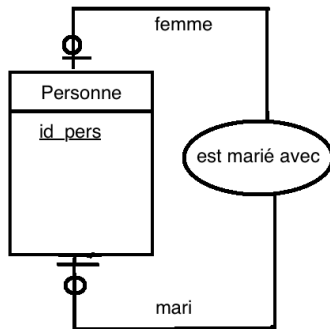
Associations réflexives : passage aux tables



Renommage des attributs :

- ▶ conjoint_1 \subseteq id_personne et conjoint_2 \subseteq id_personne
- ▶ Personne(id_personne), Mariage(conjoint_1, conjoint_2)

Associations réflexives : passage aux tables

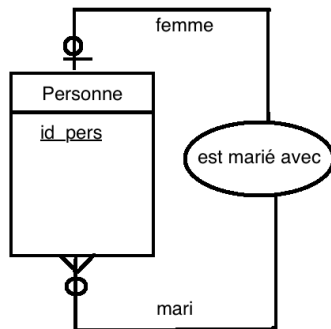


Renommage des attributs :

- ▶ femme \subseteq id_personne et mari \subseteq id_personne
- ▶ Personne(id_personne), Mariage(femme,mari)
[ou Mariage(femme,mari)]

Associations réflexives : passage aux tables

Polyandrie



Renommage des attributs :

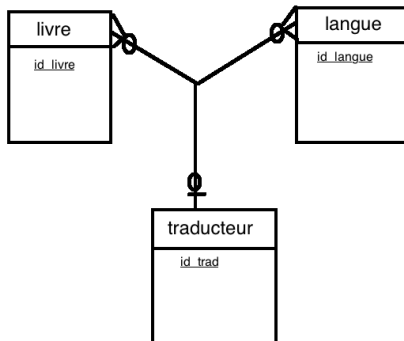
- ▶ $femme \subseteq id_personne$ et $mari \subseteq id_personne$
- ▶ $Personne(\underline{id_personne})$, $Mariage(femme, \underline{mari})$

Construire un schéma E-A : la démarche

1. Déterminer la **liste des entités**.
2. Pour chaque entité :
 - 2.1 établir la liste de ses **attributs**,
 - 2.2 parmi ceux-ci, déterminer un **identifiant**,
3. Déterminer les **relations** entre les entités
4. Pour chaque relation :
 - 4.1 dresser la liste des **attributs** propres à la relation
 - 4.2 vérifier la dimension (binaire, ternaire ?)
 - 4.3 définir les **cardinalités**
5. Vérifier le schéma obtenu, notamment :
 - 5.1 supprimer les **transitivités**
 - 5.2 s'assurer que le schéma répond au **cahier des charges**

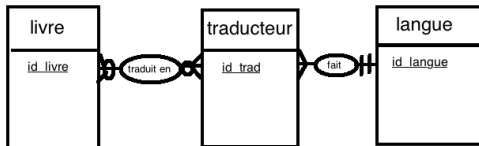
Les erreurs à ne pas commettre

Surestimer la dimension d'une relation



Les erreurs à ne pas commettre

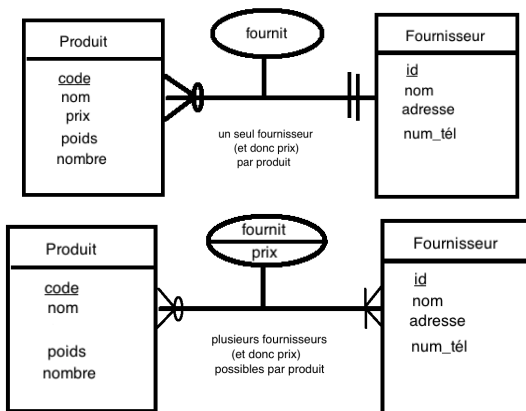
Surestimer la dimension d'une relation



Une relation binaire suffit si on suppose qu'un traducteur ne peut traduire qu'en une seule langue.

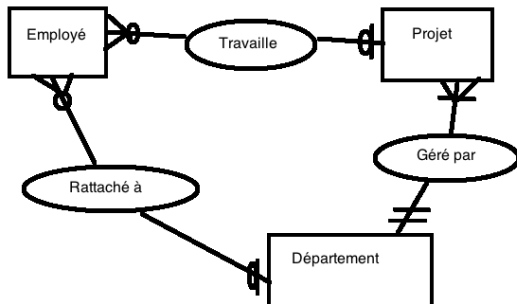
Les erreurs à ne pas commettre

Attribuer à une relation les attributs des entités participantes, et vice versa



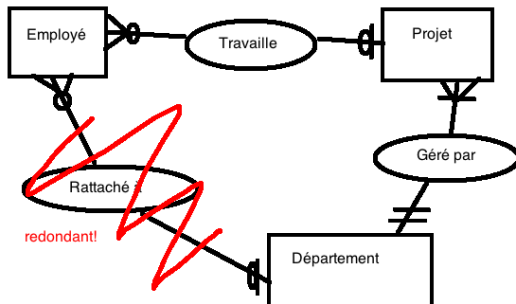
Les erreurs à ne pas commettre

Exprimer des relations redondantes, i.e., déductibles par transitivité



Les erreurs à ne pas commettre

Exprimer des relations redondantes, i.e., déductibles par transitivité



Les erreurs à ne pas commettre

Se tromper de niveau de discours

Les entités doivent représenter des ensembles d'objets ou de concepts pertinents.

- ⇒ Gestion d'un ensemble de magasins : magasin = entité
- ⇒ Gestion d'un seul magasin : magasin \neq entité

Introduire des attributs calculables

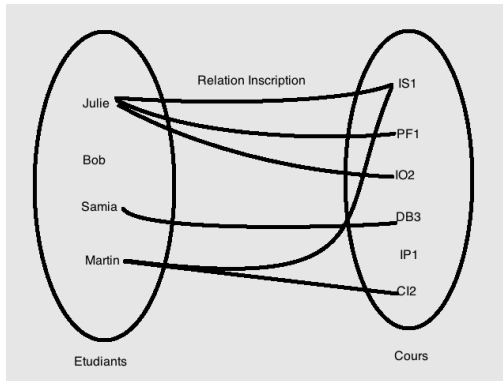
Ne pas inclure d'attributs dont la valeur est calculable à partir d'autres attributs ou en comptant les occurrences d'une entité.

- ⇒ Montant total d'une commande
- ⇒ Âge si date de naissance
- ⇒ Nombre d'étudiants inscrits

Entité ou attribut ?

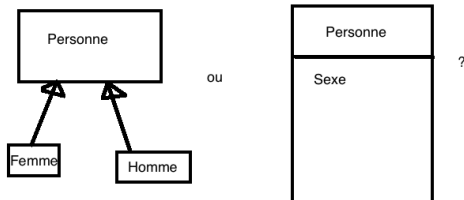
- ▶ **Attribut** : concept décrit par une seule valeur + attribuable à un autre concept (entité ou relation) du schéma
 - ▶ ex : Pays comme attribut d'Adresse
- ▶ **Entité** : concept décrit par plusieurs attributs
 - ▶ ex : Pays comme classe d'entités possédant un code, un nombre d'habitants, un PIB, etc.

Entité ou relation ?



Inscription comme relation avec attribut année : un étudiant ne peut prendre 2 ans de suite le même cours, car chaque couple (étudiant,cours) ne peut figurer qu'une seule fois dans la relation (relation = sous ensemble du produit cartésien des entités).

Héritage ou attribut ?



- ▶ 1ère solution s'il y a des propriétés spécifiques au sexe (e.g. situation militaire),
- ▶ 2ème solution sinon.