

DI

DEVELOPPEUR INFORMATIQUE

Modèle Entité-Association

Dossier d'informations

<u>Module</u>	1
<u>Séquence</u>	4
<u>Capacité</u>	2

1. Introduction : étapes de modélisation

Une des tâches essentielles, et certainement la plus difficile, des développeurs de BD est la conception du schéma de la BD. L'objectif est de structurer le domaine d'application visée de sorte à la représenter sous forme de types et de tables (relationnel). La démarche de conception s'effectue traditionnellement par abstraction successives : des problèmes et besoins de l'utilisateur vers l'implémentation physique et l'optimisation sur un SGBD. On distingue 5 grandes étapes :

- ✓ La perception du domaine d'application (du monde réel) et des besoins des utilisateurs: enquêtes, entretiens, analyses, études de cas. Les besoins sont représentées par bout dans des schémas conceptuels appelés parfois schémas externes.
- ✓ Elaboration du schéma **conceptuel** global : les différents résultats de l'étape précédente sont synthétisés dans un unique schéma qui regroupe l'expression de tous les besoins. On parle alors d'intégration des différents schémas conceptuels.
- ✓ Conception du schéma **logique** (ou relationnel): le schéma de la BD est généré dans le modèle du SGBD en partant du schéma conceptuel. Il s'agit par exemple de créer des tables dans un SGBD relationnel. Des règles permettent de passer d'un schéma conceptuel à un schéma relationnel.
- ✓ Affinement du schéma logique : le schéma logique obtenu à partir d'un schéma conceptuel doit devenir un "bon" schéma. Il doit éviter les redondances et optimiser la représentation sous forme de tables tout en assurant la complétude et la correction des résultats. Une théorie dite de normalisation permet d'obtenir un bon schéma.
- ✓ Elaboration du schéma **physique** : les bons choix d'implémentation sur le SGBD doivent être faits pour améliorer les performances du système en fonction des besoins précis du domaine d'application (des utilisateurs). Il faut prendre en compte les transactions, choisir les bonnes structures physiques comme le clustering, le partitionnement, l'indexation, etc.

Ce dossier traite de la manière:

- . de représenter un schéma conceptuel dans le modèle Entité/Association général
- . de représenter un schéma conceptuel par le formalisme Merise

Le modèle Entité/Association (E/A) permet de spécifier la structure des informations qui vont être contenues dans la base et d'offrir une représentation abstraite et indépendante du modèle logique qui sera choisi (relationnel dans notre cas).

2. Concepts de base et diagramme EA

L'idée fondamentale de ce modèle est de proposer un nombre réduit de concepts pour représenter tout domaine d'application et, surtout, d'offrir une représentation graphique du schéma conceptuel permettant une compréhension rapide des données dans la base et de leur organisation. La correspondance entre la terminologie du modèle EA et les trois concepts génériques du monde réel (objet, lien, propriété) est la suivante :

entité ou **objet**

association ou **lien**

attribut ou **propriété**

2.1. Entité

Objet du monde réel (concret ou abstrait) à propos duquel on veut enregistrer des informations et qui a une existence propre. Une entité existe indépendamment du fait qu'elle est liée à d'autres objets de la base de données.

Exemples : Mme Dupont, Mr. Durand, la cafetière X32, l'atelier de fabrication A22, le service comptabilité, ...

Type d'entité (TE) = représentation d'une classe d'entités perçues comme similaire et ayant les mêmes caractéristiques.

Exemples : employé, produit, atelier de fabrication, service, ...

On appelle **occurrence** d'un TE la représentation dans la base de données d'une entité appartenant à la classe décrite par le TE.

<Mme Dupont, Mr. Durand> → employé

<la cafetière X32> → produit

<l'atelier de fabrication A22> → atelier de fabrication

<le service comptabilité> → service

Simplification de la terminologie:

- On appellera *entité* un type d'entité
- On appellera *occurrence d'une entité* un individu particulier faisant partie de l'entité

Représentation en intention et en extension

- En intention: description d'une entité par ses attributs:
AUTEUR(code, nom, prénom, ville, date de naissance)

- En extension: ensemble des occurrences de la base de données à un instant donné:

{	A1, Dever, Régis, Lyon, 12 nov 1965 A2, Martin, Chloé, Saint-Etienne, 5 fev 1974 A3, Méline, Guillaume, Bourg, 16 jan 1977	}
---	---	---

2.2. Association

Lien entre plusieurs entités où chacune d'elles joue un certain rôle.

Exemples : l'atelier de fabrication A22 **fabrique** la cafetière X32, l'employé Durand **travaille** dans le service comptabilité, Mr. Durand **est marié avec** Mme Dupont, ...

Type d'association (TA) = représentation d'un ensemble d'associations similaires ayant les mêmes caractéristiques (liant des entités de même type avec les mêmes rôles, et possédant les mêmes propriétés).

Exemples : **fabrique** lie un atelier de fabrication à un produit, **travaille** lie un employé à un service, **est marié avec** lie une personne à une autre personne, ...

On appelle **occurrence** d'un TA la représentation dans la base de données d'une association appartenant à la classe décrite par le TA.

<atelier de fabrication A22, fabrique, cafetière X32> → <atelier de fabrication, fabrique, produit>
 <employé Durand, travaille, service comptabilité> → <employé, travaille, service>
 <Mr. Durand, est marié avec, Mme Dupont> → <employé, est marié avec, employé>

Degré d'une association

Une association peut être **binaire** lorsqu'elle met deux entités en relation.

Exemples: Les trois exemples ci-dessus

Une association est de type **n-aire** lorsqu'elle met plus de deux entités en relation.

Exemple d'une association de degré 3 (ternaire): **Est responsable** lie un développeur, un projet et un client

Si l'association lie deux entités du même type, elle est dite "**cyclique**" ou "**réflexive**" et la spécification du rôle de chaque entité est indispensable

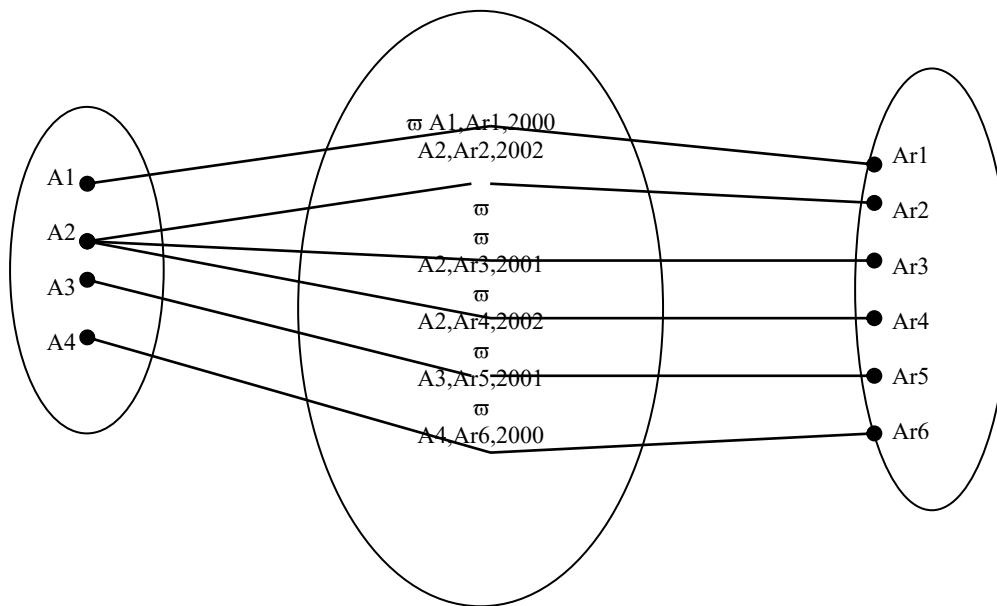
Simplification de la terminologie:

- On appellera *association* un type d'association

- On appellera *occurrence d'une association* toute correspondance entre deux ou plusieurs occurrences d'entités.

Représentation en intention et en extension

- En intention: description d'une association par les entités qu'elle relie par ses éventuels attributs:
ECRITURE(AUTEUR,ARTICLE,année)
- En extension: ensemble des occurrences de la base de données à un instant donné:



2.3. Attribut

Caractéristique, propriété associée à un TE, ou à un TA, ou participant à la composition d'un autre attribut.

Exemples : nom, prénom, salaire sont des attributs du TE employé; quantité-en-fabrication est un attribut du TA fabrique; date-de-mariage est un attribut du TA est marié avec; jour, mois, année sont des attributs composant un attribut date; ...

Domaine d'un attribut

Le domaine d'un attribut est la spécification des valeurs possibles que peut prendre l'attribut

La spécification d'un domaine est donné soit

- Par la liste des valeurs
- Par la propriété que doivent vérifier les valeurs

Exemples:

Couleur = {rouge, bleu, jaune, vert}

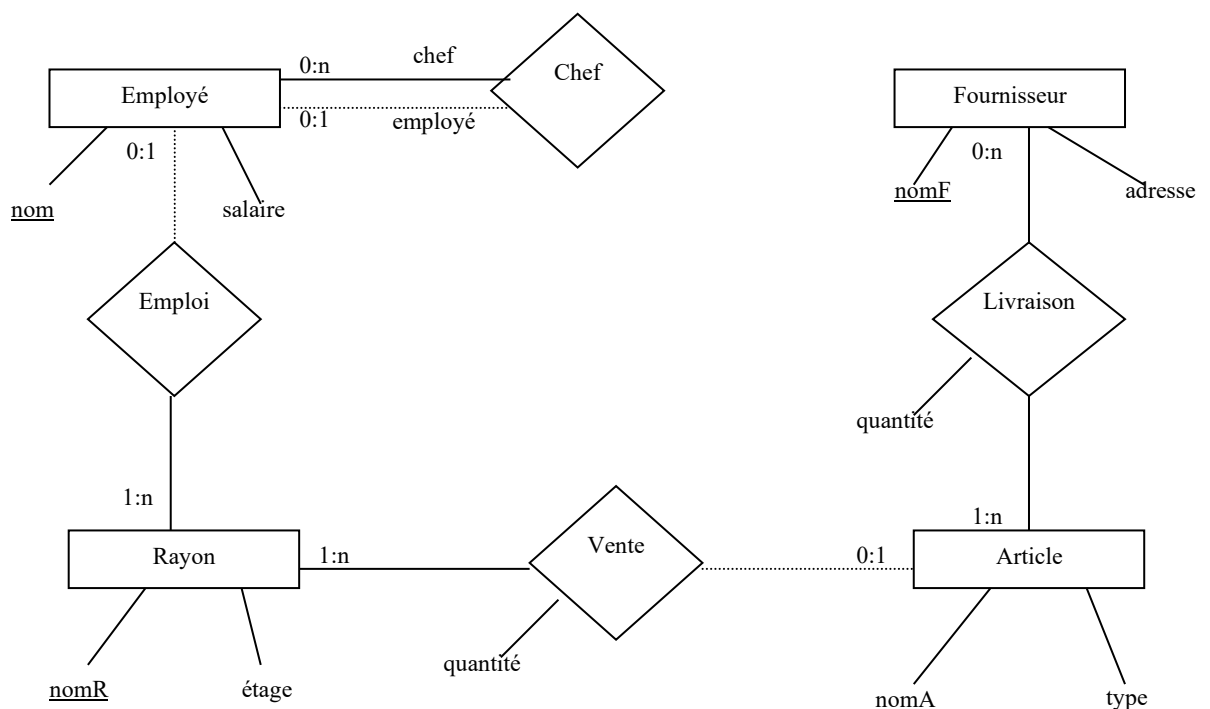
numéro de journal = entier compris entre 1 et 366

nom = chaîne de caractères alphabétique

2.4. Schéma ou diagramme Entité-Association (E/A)

Le modèle EA permet une représentation graphique d'un système d'informations. Dans cette représentation, appelée diagramme EA, les entités sont représentées par des rectangles et les associations par des losanges. Les attributs sont rattachés à l'entité ou à l'association par des traits au dessous de l'entité ou de l'association.

Voici le diagramme EA qui décrit un hypermarché :



Dans ce diagramme, sont représentés quatre entités :

- Employé, d'attributs nom et salaire
- rayon, d'attributs nomR et étage
- Article, d'attributs nomA et type
- Fournisseur, d'attributs nomF et adresse

Ces entités sont reliées par les associations suivantes :

- Livraison, d'attribut quantité, liant Fournisseur et Article
- Vente, d'attribut quantité, liant Rayon et Article
- Emploi, liant Employé et Rayon
- Chef, cyclique, liant Employé (avec le rôle chef) et Employé (avec le rôle employé)

3. Description d'un schéma

3.1. Une entité est décrite par les spécifications suivantes :

- le nom du type d'entité;
- une définition libre précisant la population (l'ensemble des occurrences) exacte de l'entité (pas sur le diagramme);
- la description des attributs de l'entité.

Exemples : description de l'entité "Employé" de la base de données hypermarché,

- nom : Employé
- définition : "toute personne salariée par l'entreprise en ce moment."
- attributs : nom, salaire (avec leur description)

Remarques :

- deux entités différentes ne peuvent pas avoir le même nom.
- la définition libre est une partie très importante de la description de l'entité. C'est elle qui permet de définir exactement, de façon non ambiguë, la population de l'entité. Elle inclut notamment la spécification temporelle qui permet de savoir si le contexte modélisé couvre uniquement la situation actuelle ou aussi l'historique et/ou l'avenir.

3.2. Une association est décrite par les spécifications suivantes :

- le nom de l'association;
- une définition libre précisant la population exacte de l'association;
- les noms des entités participant à l'association, avec leur rôle;
- pour chaque rôle, ses cardinalités : c'est une information supplémentaire exprimant la règle de participation des entités dans les associations. Les cardinalités consistent en deux nombres, min et max, spécifiant le nombre minimal et le nombre maximal d'occurrences de l'association qui peuvent, à un instant donné, lier par ce rôle une occurrence déterminée de entité en question. min et max sont deux entiers tels que $\text{max} \geq \text{min}$, $\text{min} \geq 0$, $\text{max} \geq 1$.
- la description des attributs de l'association s'il en existe.

Exemple : description de l'association "Emploi" de la base de données hypermarché,

- nom : Emploi
- définition : "lie un employé au rayon dans lequel cet employé travaille aujourd'hui."
- Entités participantes avec leur rôle: <Employé, "Est employé">, <Rayon, "Emploie">
- cardinalités : Est employé (min=0,max=1), Emploie (min=1,max=n)
- attributs : /

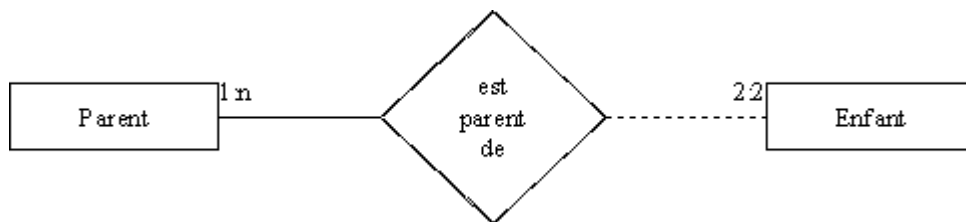
Les cardinalités possibles pour un rôle (ici E de Emploi) sont les suivantes :

- min = 0 : un employé peut ne travailler dans aucun rayon
- min = 1 : un employé doit travailler dans au moins un rayon
- max = 1 : un employé ne peut travailler dans plus d'un rayon
- max = n : un employé peut travailler dans plusieurs rayons

avec les conventions graphiques suivantes :

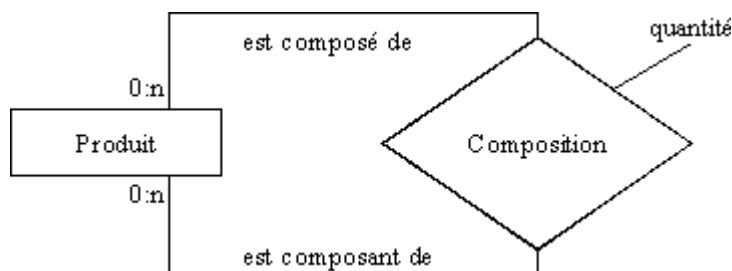
- min = 0, max = 1 représentées par un trait unique discontinu
- min = 1, max = 1 représentées par un trait unique continu
- min = 0, max = n sont représentées par deux traits discontinus
- min = 1, max = n sont représentées par deux traits, l'un continu, l'autre discontinu

Les cardinalités peuvent être indiquées par deux chiffres (min, max ou min:max) au voisinage du trait représentant le lien correspondant :



Signification des cardinalités dans ce diagramme : un parent peut avoir de 1 à n enfants, mais un enfant a toujours deux parents.

Cas particulier des associations réflexives (ou cycliques) : une association est dite réflexive (ou cyclique) si elle lie plusieurs fois la même entité (avec des rôles différents). Dans ce cas, le nom des rôles est essentiel et il doit figurer sur les pattes de l'association.



Exemples : les produits peuvent être composés d'autres produits dans une certaine quantité (et réciproquement).

L'association hiérarchique ou contrainte d'intégrité fonctionnelle (C.I.F)

Elle met en jeu DEUX ENTITÉS MAXIMUM entre lesquelles existe une dépendance fonctionnelle, c'est à dire qu'à UNE valeur de l'identifiant de l'ENTITÉ SOURCE, ne correspond qu'UNE SEULE valeur de l'ENTITÉ BUT.

La CARDINALITÉ côté source de l'association sera 1,1 ou 0,1.

IMPORTANT : une association hiérarchique ne peut pas être porteuse de propriétés !

Les associations EMPLOI et VENTE de l'exemple de l'hypermarché sont des associations hiérarchiques.

L'association non hiérarchique

Elle met en jeu DEUX ENTITÉS OU PLUS.

Une telle association peut être porteuse de données, données qui dépendent de l'ensemble des identifiants des entités auxquelles est reliée l'association.

L'association LIVRAISON de l'exemple de l'hypermarché est une association non hiérarchique porteuse de données

3.3. Un attribut est décrit par les spécifications suivantes :

- nom de l'attribut;
- définition libre (libellé le plus clair possible);
- cardinalité : min et max, spécifiant combien de valeurs de cet attribut sont autorisées pour une occurrence;
- si l'attribut n'est pas composé d'autres attributs : domaine de valeur associé, spécifiant quel est l'ensemble des valeurs autorisées pour l'attribut
- si l'attribut est composé d'autres attributs : description des attributs composants

Exemple : description de l'attribut "nom" de l'entité Employé de la base de données hypermarché,

- nom : nom
- définition : "nom de l'employé, nom de jeune fille pour une femme."
- cardinalités : min = 1, max = 1 (tout employé a un nom et un seul)
- domaine de valeurs : l'ensemble des chaînes de caractères de longueur inférieur à 15

Exemple : description d'un attribut "date de naissance" d'une entité Personne,

- nom : date de naissance
- définition : "date de naissance de la personne."
- cardinalités : min = 1, max = 1
- composition :
 - nom : jour
 - définition : "jour de naissance de la personne"
 - cardinalité : 1,1
 - domaine : les entiers dans l'intervalle [1,31]
- nom : mois
- définition : "mois de naissance de la personne"
- cardinalité : 1,1
- domaine : les entiers dans [1,12]
- nom : année
- définition : "année de naissance de la personne"
- cardinalités : 1,1

- domaine : les entiers dans [1870, 1995]

attribut simple = un attribut qui n'est pas décomposé d'autres attributs, ses valeurs sont atomiques. Un domaine lui est associé.

Exemple : salaire, téléphone

attribut complexe = un attribut qui est décomposé d'autres attributs, ses valeurs sont des valeurs composées. L'attribut d'un attribut peut lui même être complexe.

Exemple : adresse composé des attributs rue, ville, code postal, qui eux, sont simples.

attribut monovalué = un attribut qui ne peut prendre qu'une seule valeur par occurrence (max = 1).

Exemple : nom, date de naissance

attribut multivalué = un attribut qui peut prendre plusieurs valeurs par occurrence (max > 1).

Exemple : prénoms, téléphones

attribut obligatoire = un attribut qui doit prendre une valeur au moins par occurrence (min = 1).

Exemple : nom, prénom

attribut facultatif = un attribut qui peut ne pas prendre de valeur dans une occurrence (min = 0).

Exemple : salaire, téléphone

4. Identifiant des entités et des associations

Un identifiant d'une entité ou d'une association est un ensemble minimum d'attributs tel qu'il n'existe pas deux occurrences de l'entité ou de l'association qui ont la même valeur pour ces attributs.

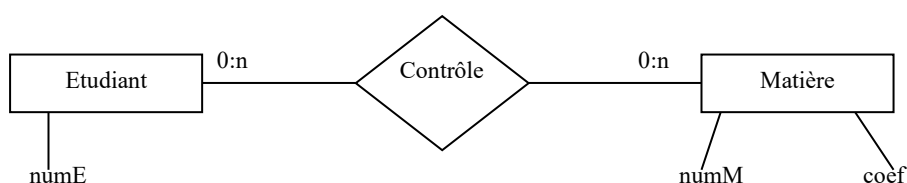
Exemple : n°employé est l'identifiant du TE Employé

Un identifiant est désigné sur un diagramme en le soulignant.

4.1. Identifiant d'une association

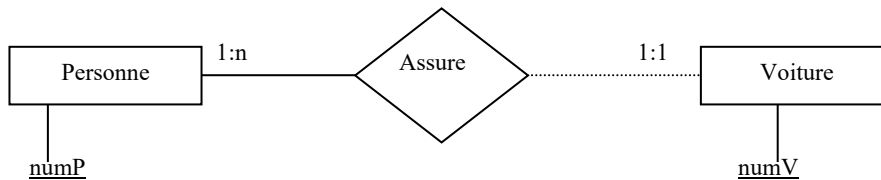
Une association dont tous les rôles ont une cardinalité maximum supérieure à 1 a, en général, un identifiant qui est constitué de l'ensemble des identifiants des entités liées.

Exemple : soit une association Contrôle, avec une moyenne pour un étudiant et une matière suivie,



L'identifiant du TA Contrôle est : Etudiant.NumE + Matière.NumM

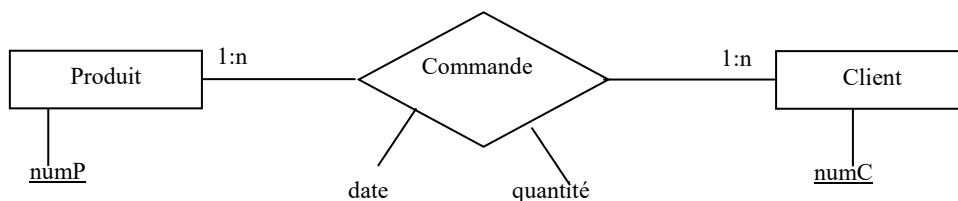
Néanmoins, il n'est pas toujours vrai que l'identifiant d'une association est constitué de l'ensemble des identifiants des entités liées. Si l'un des rôles a une cardinalité maximum égale à 1, l'identifiant de l'entité associé à ce rôle est un identifiant de l'association. Par exemple, soit l'association suivante :



L'identifiant de l'association **Assure** n'est pas **Personne.numP + Voiture.NumV** : il ne constitue pas un ensemble minimal. En effet, l'attribut **numV** de **Voiture** suffit, à lui seul, pour identifier une occurrence d'**Assure**. Ceci est dû à la cardinalité 1:1 du rôle de **Voiture** dans l'association, qui garantit que pour une valeur de **numV** il n'y aura jamais qu'une seule occurrence d'**Assure** avec cette valeur de **numV**.

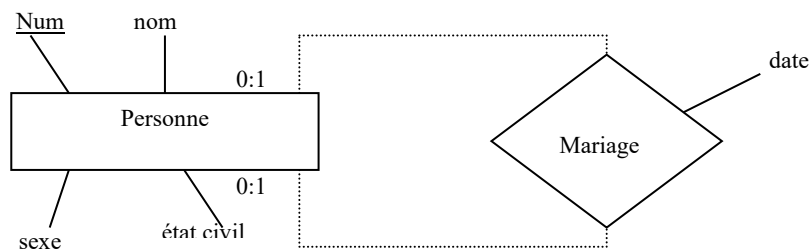
Règle : si l'association a une cardinalité maximum égale à 1 pour une des entités liées, alors tout identifiant de cette entité est identifiant de l'association.

Une autre règle peut être établie pour les cas où plusieurs occurrences de l'association lient les mêmes occurrences des entités. Dans l'exemple ci-dessous, plusieurs commandes peuvent être passées par un même client pour un même produit à des dates différentes. Dans ce cas l'identifiant de l'association contient au moins un attribut de l'association.



L'association **Commande** a pour identifiant : **Produit.NumP + Client.NumC + Commande.date**.

Soit l'association Mariage suivant,

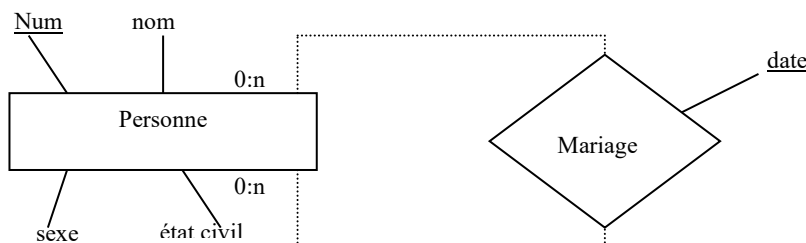


Une occurrence de l'association Mariage est un triplet : <<un époux>, <une épouse>, <date>

Si on admet que l'attribut Num est l'identifiant de Personne, Mariage possède deux identifiants : Personne.Num (rôle époux) et Personne.Num (rôle épouse)

On considère qu'une personne identifie un mariage (une personne n'est pas mariée simultanément à plusieurs personnes).

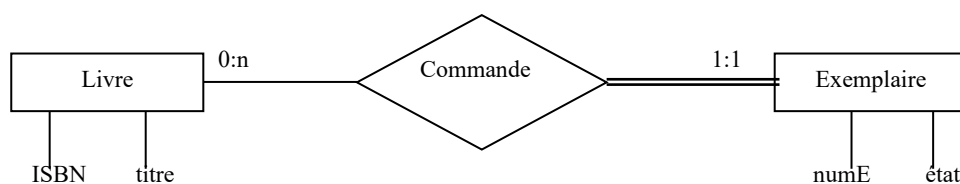
Cette définition n'est valable que si la population de l'association Mariage ne contient que les mariages en cours (on ne garde pas l'historique). Si l'association conservait l'historique (les mariages passés), les cardinalités des deux rôles seraient 0:n et l'identifiant de l'association serait : Personne.Num (époux) + date ou Personne.Num (épouse) + date ; car il n'est pas exclu que les deux personnes se remarient une deuxième fois ensemble après avoir divorcé.



4.2. Identifiant d'une entité faible

Une entité faible est une entité dont aucun sous-ensemble d'attributs ne constitue un identifiant (elle n'a pas d'identifiant qui lui soit interne), et qui est lié par une association binaire de cardinalité 1:1 à un autre TE dont il dépend.

Dans l'exemple ci-dessous, Exemple (qui représente un exemplaire d'un livre) est une entité faible (NumE n'est pas un identifiant) dépendant de l'entité Livre,



L'identifiant d'une entité faible (qui est le même que celui de l'association) est constitué de l'identifiant de l'entité dont elle dépend et d'un (ou plusieurs) attributs de l'entité faible. L'identifiant de Exemple (et de « est un ») est : ISBN + N°ex
NB : ISBN signifie « International Standard Book Number », c'est un numéro qui identifie tout nouveau livre édité.

5. Contraintes d'intégrité

Les concepts de base du modèle entité-association (entité, association, attribut) permettent d'exprimer directement un certain nombre de contraintes de type structurel mais ne suffisent pas à donner une image complète de la réalité. Le modèle doit être enrichi par d'autres composantes, graphiques ou non.

Si dans une base de données les données ne satisfont pas les contraintes d'intégrité, il y a une « erreur » dans la base de données : on dit alors qu'elle est incohérente ou non intègre.

5.1. Contraintes structurelles

5.1.1. Intégrité d'entité

Cette contrainte stipule que la valeur de l'attribut identifiant pour une occurrence d'un type d'entité doit être unique, stable et renseigné

5.1.2. Intégrité référentielle

Pour toute occurrence d'une association, les occurrences des entités intervenant dans la relation doivent obligatoirement exister. Par exemple, pour une occurrence de l'association ASSURE entre les entités PERSONNE et VOITURE, les occurrences des deux entités doivent exister préalablement.

5.1.3. Cardinalités

Ce type de contraintes a été étudié précédemment

5.2. Contraintes logiques

Ce sont des contraintes exprimant des liens logiques entre des associations : par exemple exclusion, inclusion des occurrences d'une association par rapport à une autre association.

Exemple d'une contrainte d'exclusion :

Soient deux entités *Personne* et *Cours* et deux associations *Etudiant* et *Enseignant*. Il existe une contrainte d'exclusion entre les deux associations car une *Personne* ne peut être *Etudiant* et *Enseignant* dans le même *Cours*

Ces contraintes seront détaillées et modélisées lors de l'étude de la modélisation conceptuelle des données par la méthode Merise.

5.3. Contraintes de valeurs

5.3.1. Contraintes de valeurs d'un attribut

Elles limitent le domaine de valeurs possibles pour une propriété : âge <130 , code compris entre 1 et 1000.

5.3.2. Contraintes entre valeurs d'attributs

Elles expriment le fait que la valeur d'un attribut dépend d'autres éléments : la date d'un sinistre doit être postérieure à la date de signature du contrat ; si une *Personne* participe à l'association *Mariage*, alors son état civil doit être « marié ».

Certaines contraintes peuvent concerner l'évolution dans le temps d'une donnée : le kilométrage d'une voiture ne peut que croître.

En conclusion on peut dire qu'un schéma entité-association est un ensemble d'entités et d'associations avec leurs attributs, et des contraintes d'intégrité associées :

$$\boxed{\text{schéma conceptuel EA} = (\{TE\}, \{TA\}, \{CI\})}$$

6. Conception d'un schéma EA

6.1. Approche par le dictionnaire des données

Il s'agit de recenser de façon exhaustive les données de base du domaine étudié puis d'analyser les dépendances entre elles : les **dépendances fonctionnelles** .

On distingue 2 types de dépendances fonctionnelles:

DF simple :

On dira qu'une DF est simple si sa source n'est composée que d'un seul attribut.

NumEmp → Nom, Salaire

Une DF simple caractérise une *entité* dont la source est l'identifiant et dont les propriétés sont constituées par le but de la DF.

DF composée :

On dira qu'une DF est composée si sa source est composée par la réunion de plusieurs attributs.

NumFour, NumArt → Qté_livrée

Une DF composée caractérise une *association entre entités* dont la source est l'identifiant et dont les propriétés sont constituées par le but de la DF.

6.2. Construction progressive du diagramme EA

Le concepteur étudie l'existant et les besoins de l'entreprise en recensant les fiches, formulaires, bordereaux,... utilisés jusqu'à présent dans l'entreprise et en interviewant les personnes de l'entreprise sur les informations qu'elles utilisent et dont elles aimeraient disposer. Le concepteur établit ainsi progressivement la liste des entités, des associations, des attributs et des règles d'intégrité et de gestion. Le diagramme entité-association est ainsi construit progressivement en précisant pour chaque type sa définition.

Exemple : dans une bibliothèque, la bibliothécaire explique, « On ne peut pas emprunter de livre si on en a déjà emprunté trois, ou si on est en retard pour en rendre un. » C'est une règle de gestion, dont on peut extraire les informations suivantes pour le schéma de la base de données :

- Il y a (au moins) deux entités, Lecteur et Livre.
- Il y a une association Emprunte entre Livre et Lecteur. Cette association a une cardinalité 0:3 pour le lecteur.
- Il faut mémoriser la date d'emprunt. C'est un attribut qui dépend de Livre et de Lecteur, donc c'est un attribut de l'association Emprunte.
- Le fait de ne pas pouvoir emprunter plus de trois livres simultanément sera assuré automatiquement par le SGBD grâce à la cardinalité maximale 3 de Lecteur pour l'association Emprunte.

6.3. Vérification du diagramme

Une fois le diagramme entité-association établi, plusieurs types de vérification sont effectuées :

- vérification syntaxique : il s'agit de vérifier que les règles du modèle EA sont respectées.
- par jeu d'essai : le concepteur vérifie grâce à un échantillon de données que le diagramme permet effectivement de stocker les informations nécessaires à l'entreprise.
- complétude par rapport aux traitements : le concepteur vérifie que le diagramme contient tous les types d'informations nécessaires à l'exécution des traitements prévus.
- par les utilisateurs : le concepteur présente le diagramme accompagné des définitions aux personnes qui utiliseront la base de données et vérifie que les informations contenues correspondent bien aux besoins.

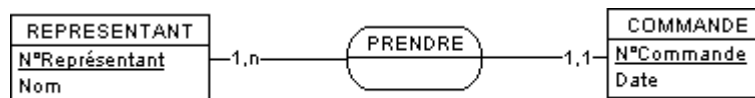
Chaque oubli, erreur, modification,... détecté lors des vérifications entraîne une mise à jour du diagramme et relance les différentes phases de vérification.

7.2. Contrainte d'intégrité fonctionnelle

Une Contrainte d'Intégrité Fonctionnelle (en abrégé : CIF) se définit par le fait que l'une des entités participant à l'association est complètement déterminée par la connaissance d'une ou plusieurs autres entités participant dans cette même association.

CAS DE L'ASSOCIATION DE DIMENSION 2 :

La CIF consiste simplement en une cardinalité 1,1 sur l'une des pattes :



La connaissance d'une commande détermine celle du représentant qui l'a prise : il n'y en a qu'un.

Remarques:

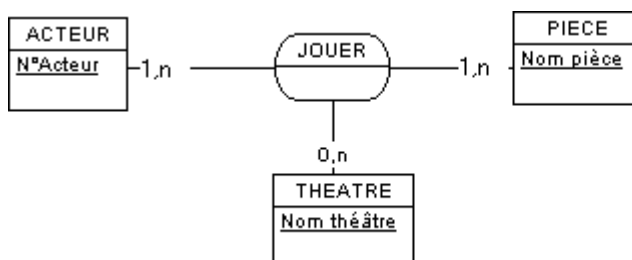
Certains auteurs proposent d'écrire explicitement CIF dans l'association. Nous le déconseillons pour 2 raisons :

- Cela mélange le modèle conceptuel avec le modèle logique que nous verrons plus tard. Et dans le modèle conceptuel "CIF" est déjà écrit avec "1,1". Il est inutile d'en rajouter.
- On perd le nom de l'association et le schéma devient moins lisible pour les utilisateurs.

CAS D'UNE ASSOCIATION DE DIMENSION SUPERIEURE A 2 :

Supposons que l'on veuille réaliser un annuaire des spectacles de théâtre de l'agglomération caennaise qui fournisse la liste des théâtres, des pièces qui y sont à l'affiche ainsi que des acteurs qui jouent dans ces pièces.

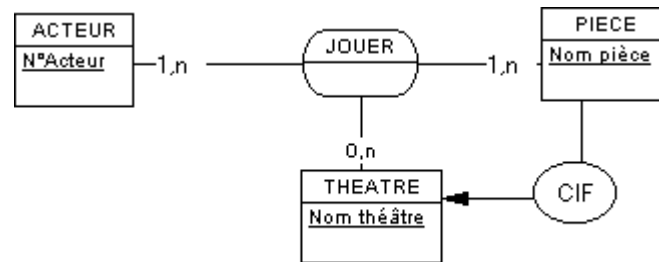
On peut proposer le modèle suivant :



Supposons maintenant qu'il existe une règle selon laquelle les théâtres ont l'exclusivité des pièces représentées. Autrement dit, une pièce est jouée dans un théâtre et un seul.

Nous sommes en présence d'une CIF : la connaissance de la pièce implique celle du théâtre (le seul qui soit autorisé à la mettre à l'affiche)

On pourrait dessiner cette règle comme ceci :

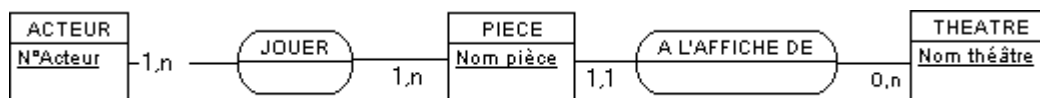


Mais, puisque connaissant la pièce, on peut en déduire le théâtre, on peut détacher l'entité THEATRE de l'association JOUER : Si on sait dans quelles pièces jouent les acteurs, on pourra toujours retrouver *LE* théâtre associé à chaque pièce.

Plus généralement :

Dans le cas d'une association de dimension supérieure à 2 et lorsqu'il y a une CIF, l'entité déterminée peut être détachée de l'association initiale pour rester associée avec la seule entité déterminante.

Ce qui donne dans notre exemple la simplification suivante :

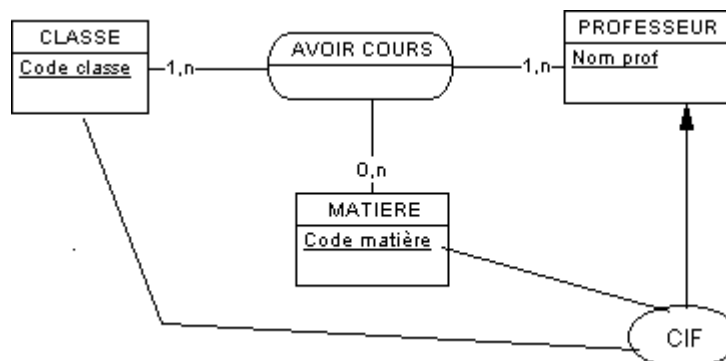


CAS OU DEUX ENTITES DETERMINENT ENSEMBLE UNE AUTRE ENTITE.

Dans un lycée, les professeurs donnent des cours à des classes dans certaines matières.

Mais on sait encore que dans une matière, une classe n'a qu'un seul professeur.

Ceci peut s'écrire ainsi : CLASSE * MATIERE -> PROFESSEUR



Cette CIF ne permet pas une simplification du modèle conceptuel mais aura une conséquence sur le modèle logique (relationnel) : quel sera l'identifiant (la clé) de l'association AVOIR COURS ?