

Génie Logiciel

UE1

Bases de données

TRAORE Aboudou

Ingénieur de conception des techniques informatiques

Développement et intégration d'applications

Consultation et formation en informatique

traobou12@yahoo.fr

Ouagadougou, Octobre 2015

LEÇON II

**Modèles de représentation, dépendances fonctionnelles
et normalisation**

Sommaire

Présentation de la leçon

Objectif général

Objectifs spécifiques

Contenu de la leçon

Activités d'apprentissage

Présentation de la leçon

Dans ce cours nous parlerons des Modèles de représentation, des dépendances fonctionnelles et de la normalisation des relations.

Objectif général

L'objectif général est de pouvoir décrire et représenter l'aspect conceptuel des données tout en respectant les normes de normalisation.

Objectifs spécifiques

A la fin de la leçon, l'apprenant doit être à mesure faire un schéma conceptuel E/A, un schéma relationnel et de passer du modèle E/A à un schéma relationnel afin de faciliter la construction de la base de données.

Section I : Modèles de représentation

1. Le modèle Entité/Association

Le modèle EA en français (pour Entité Association), ER en anglais (pour Entity Relationship) est un formalisme retenu par l'ISO ¹ pour décrire et représenter l'aspect conceptuel des données à l'aide d'entités et d'associations.

Il est utilisé pendant la conception dans le cadre des bases de données. Il permet de spécifier la structure des informations qui vont être contenues dans la base de données et d'offrir une représentation abstraite indépendante du modèle logique qui sera choisi ensuite.

1.1 Concepts généraux

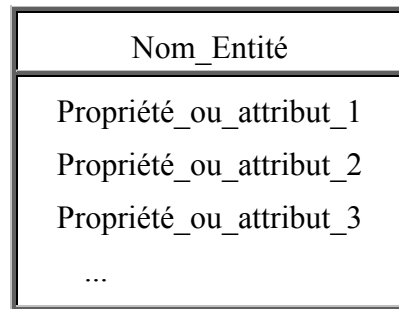
Le modèle EA permet de distinguer les *entités* qui constituent la base de données, et les *associations* entre ces entités. Ces concepts permettent de donner une structure à la base de données.

On distingue:

- Une **entité** : C'est la représentation d'un objet matériel ou immatériel. Par exemple un employé, un film, un bulletin de paie.

¹ L'ISO (International Organization for Standardization ou Organisation internationale de normalisation en français) → www.iso.org

Les **entités**, sont représentées par des rectangles. Chaque entité est caractérisée par un ensemble d'attributs parmi lesquels un ou plusieurs forment l'identifiant unique.



- **L'occurrence** d'entité: C'est la représentation de l'entité dans le monde réel. Autrement dit c'est une instance de l'entité.

Par exemple l'étudiant nommé *KARIM* est une instance ou occurrence de l'entité *Etudiant*.

- **Les attributs, propriétés, ou caractéristiques** d'une entité : Ce sont des données élémentaires relatives à une entité.

Par exemple, un numéro d'employé, une date de début de projet

Un attribut est caractérisé par:

- un nom ;
- un type ;
- et éventuellement une valeur initiale.

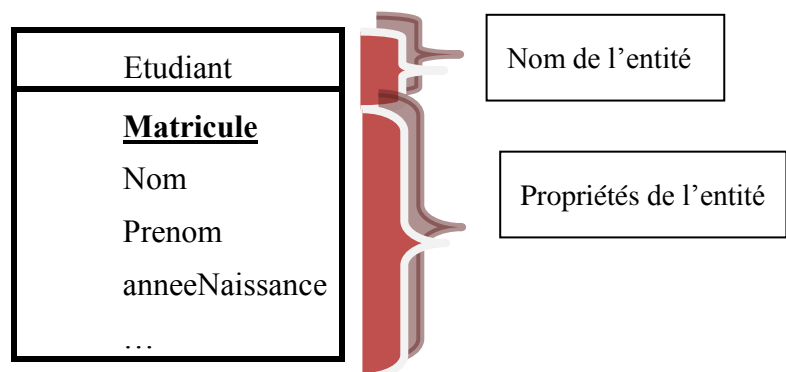
On distingue les attributs:

- **multi-values** qui sont constitués d'un ensemble de valeurs prises dans un même domaine ; une telle construction permet de résoudre le problème des numéros de téléphones multiples ;
- **composés** qui sont constitués par agrégation d'autres attributs ; un attribut adresse peut par exemple être décrit comme l'agrégation d'un code postal, d'un numéro de rue, d'un nom de rue et d'un nom de ville.
- **L'identifiant** : C'est une propriété ou groupe de propriétés qui sert à identifier de manière unique une entité (en gras et souligné dans l'exemple ci-dessous).

Choix de l'identifiant : L'identifiant d'une entité est choisi de façon à ce que deux occurrences de cette entité ne puissent pas avoir le même identifiant.

Par exemple, le numéro matricule de l'étudiant sera l'identifiant de l'entité Etudiant.

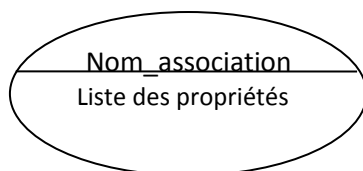
Exemple complet de représentation:



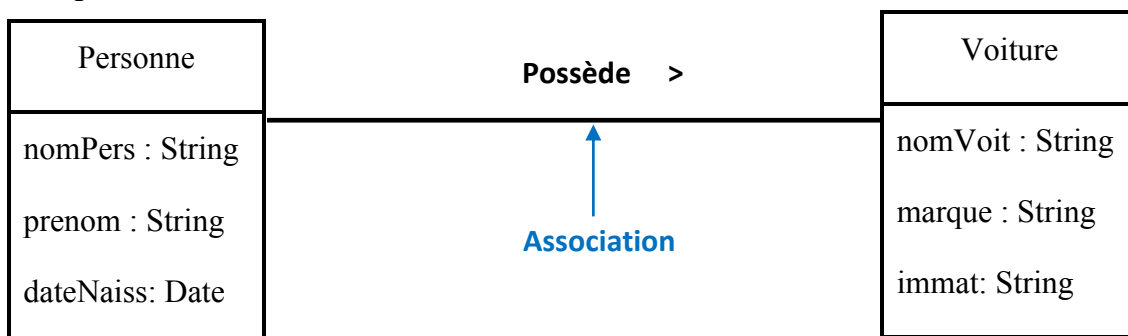
- **L'objet** : C'est la représentation d'une entité dans le monde réel (ou du monde virtuel pour les objets immatériels).
- Une **association**: Elle représente
 - une relation entre plusieurs entités;
 - les liens qui existent entre les objets dans le monde réel.

Une association peut avoir des propriétés particulières.

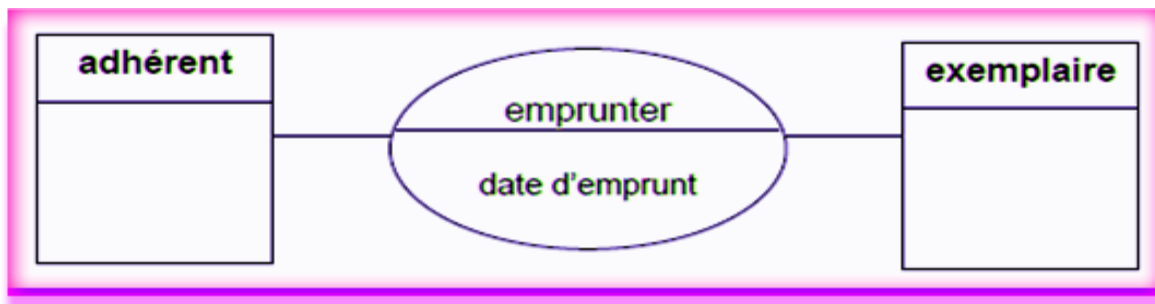
Les associations entre entités sont représentées par *des traits simples* entre ces rectangles ou par le symbole :



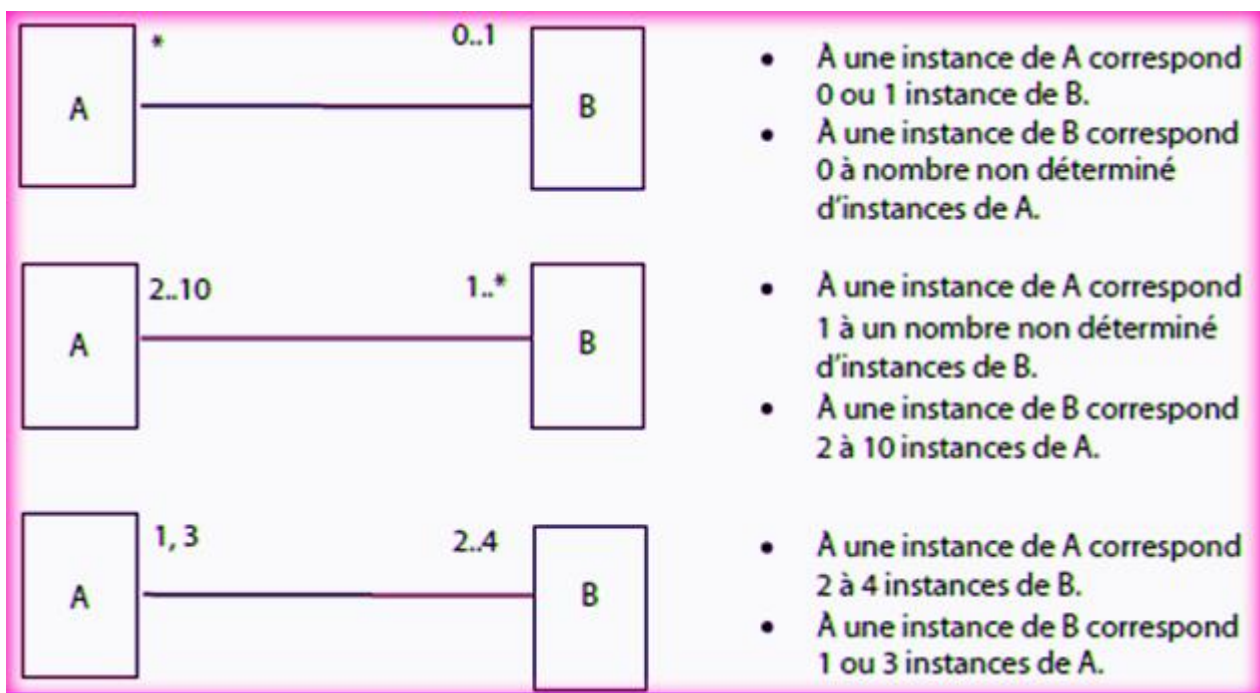
Exemples avec entités:



Ou



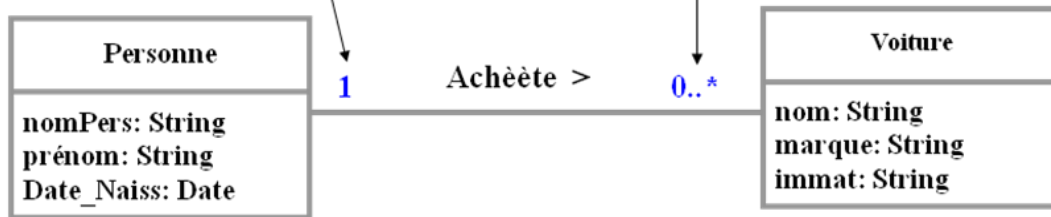
- **Les cardinalités:** c'est le nombre minimum et maximum de fois qu'une occurrence de l'entité participe aux occurrences de l'association (relation).
 - minimum: généralement 0 ou 1
 - maximum: généralement 1 ou n



Exemple 1 : La lecture se faisant selon la notation UML

Une voiture est achetée par une et une seule personne

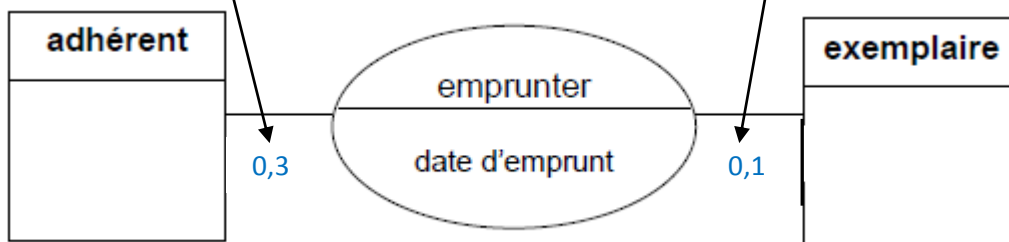
Une personne peut acheter 0 ou n voitures



Exemple 2 : La lecture se faisant selon la notation MERISE

Un adhérent peut emprunter 0 ou 3

Un exemplaire est emprunté par 0 ou une seule personne



- La cardinalité 0,3 indique qu'un adhérent peut être associé à 0, 1, 2 ou 3 livres, c'est à dire qu'il peut emprunter au maximum 3 livres.
- A l'inverse un livre peut être emprunté par un seul adhérent, ou peut ne pas être emprunté.

1.2 Entités faibles

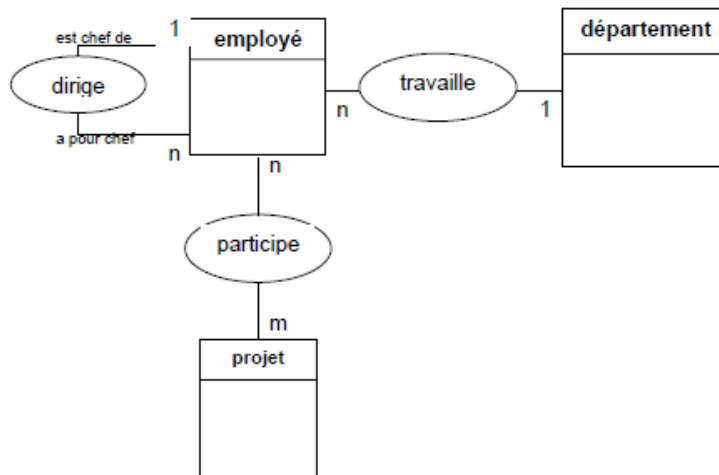
Lorsqu'une entité ne peut exister qu'en étroite association avec une autre, et est identifiée relativement à cette autre entité. On parle alors d'*entité faible*. Parallèlement l'autre entité est dite forte. Une entité faible est toujours identifiée par rapport à une autre entité.

Remarque: Selon la notion MERISE, l'entité faible est celle qui porte le plus faible cardinalité. Et c'est l'inverse pour la notation UML.

Exemple : Ici *Voiture* est l'entité faible et la *Personne* est forte.



1.3 Exemple de diagramme E/A



1.4 Avantage et inconvénients du modèle E/A

Le modèle conceptuel le plus courant. Parmi les avantages du modèle E/A, on a:

- Le modèle Entité/Association est *simple* et *pratique* ;
- Un modèle très abstraite qui **n'est pas associé à un langage** ;
- Il n'y a que 3 concepts : *entités*, *associations* et *attributs* ;
- Il est approprié à une représentation graphique intuitive, même s'il existe beaucoup de conventions ;
- Il permet de modéliser rapidement des structures pas trop complexes.

En outre, elle permet :

- l'analyse du monde réel ;
- la conception du système d'information ;
- la communication entre différents acteurs de l'entreprise.

Il y a malheureusement plusieurs inconvénients parmi lesquels on peut citer :

- Il est *non-déterminisme* : il n'y a pas de règle absolue pour déterminer ce qui est entité, attribut ou relation ;

- il ne propose pas d'opérations sur les données ;
- Il reste pauvre : il est difficile d'exprimer des contraintes d'intégrité, des structures complexes ;
- La conception de schéma reste en partie matière de bon sens et d'expérience : On essaie en général :
 - de se ramener à des associations entre 2 entités : au-delà, on a probablement intérêt à transformer l'association en entité ;
 - d'éviter toute redondance : une information doit se trouver en un seul endroit ;
 - enfin et surtout de privilégier la simplicité et la lisibilité, notamment en ne représentant que ce qui est strictement nécessaire.

2. Le modèle relationnel

Un *modèle de données* définit un mode de représentation de l'information selon trois composantes :

- des *structures de données* ;
- des *contraintes* qui permettent de spécifier les règles que doit respecter une base de données ;
- des *opérations* pour manipuler les données, en interrogation et en mise à jour.

2.1 Définitions

Schéma relationnel : Le schéma relationnel est l'ensemble des *relations* qui modélisent le monde réel.

Les relations représentent les entités du monde réel (comme des personnes, des objets, etc.) ou les associations entre ces entités.

Un schéma de relation est simplement le nom de la relation suivi de la liste des attributs, chaque attribut étant associé à son domaine.

La syntaxe est donc :

$R (A1:D1, A2:D2, \dots, An:Dn)$ avec R le nom de l'entité et Ai les noms des attributs et Di les domaines. Avec i allant de 1 à n .

Ou tout simplement $R (A1, A2, \dots, An)$. Autrement dit : **nom_de_l'entité** (liste des attributs de l'entité).

Exemple : Etudiant (Matricule, Nom, Prenom, anneeNaissance).

Domaine : Un *domaine de valeurs* est un ensemble d'instances d'un type élémentaire.

Exemple : les entiers, les réels, les chaînes de caractères, etc.

Clé d'une relation : La clé d'une relation est le plus petit sous-ensemble des attributs qui permet d'identifier chaque ligne de manière unique.

Nuplet (ou ligne ou tuple) : Un tuple est un enregistrement ou une ligne dans la représentation d'une relation sous forme de table.

Base de données relationnelles: Base de données dont le schéma est un ensemble de schémas de relations et dont les occurrences sont les tuples de ces relations.

Système de gestion de bases de données relationnel : C'est un logiciel supportant le modèle relationnel, et qui peut manipuler les données avec des opérateurs relationnels.

2.2 Passage d'un schéma conceptuel E/A à un schéma relationnel

On suppose que pour chaque entité, la clé est définie par un identifiant **idE**.

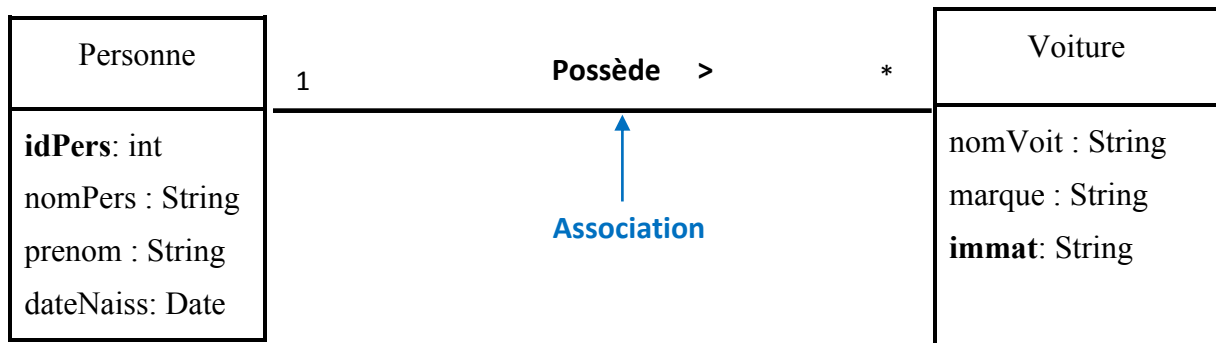
- **Règles générales de passage : entités**

Rappel : le schéma d'une relation est constitué du nom de la relation, suivi de la liste des attributs.

Alors, pour chaque entité du schéma E/A:

1. On crée une relation de même nom que l'entité ;
2. Chaque propriété de l'entité, y compris l'identifiant, devient un attribut de la relation ;
3. Les attributs de l'identifiant constituent la clé de la relation.

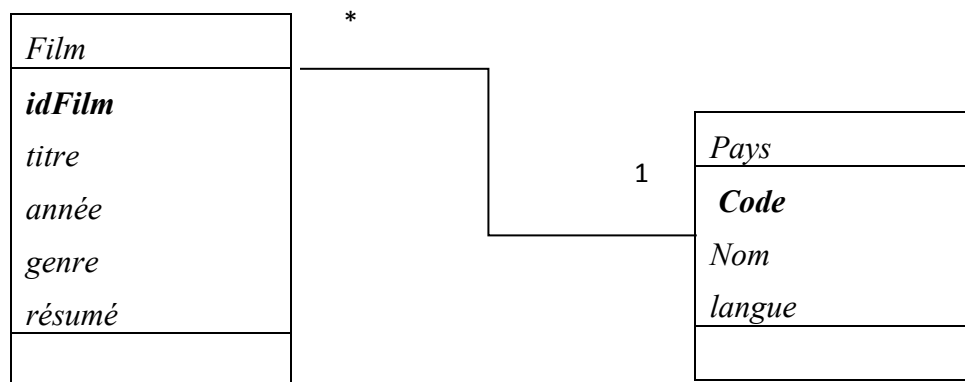
Exemple 1 : Soit le schéma E/A :



On a :

Personne(**idPers**,nomPers,prenom,dateNaiss) et Voiture(**immat**,nomVoit,marque)

Exemple 2 : Soit le schéma E/A :

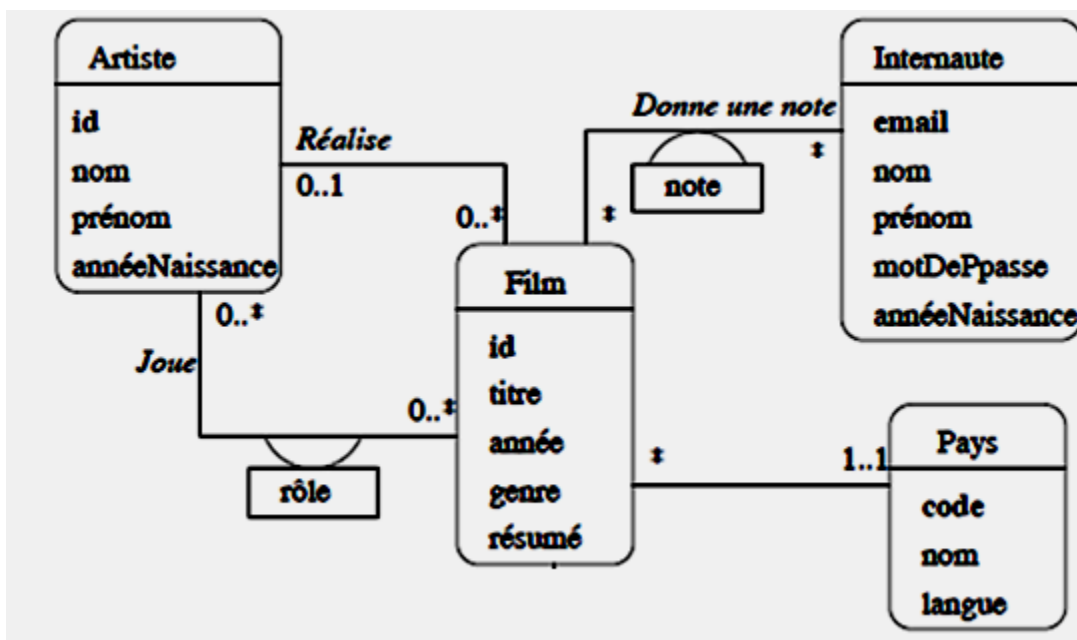


On a:

- Film (**idFilm**, titre, année, genre, résumé)
- Pays (**code**, nom, langue)

Exemple 3 : soit l'E/A

À partir de ce schéma E/A



on obtient les relations suivantes :

- Film (**idFilm**, titre, année, genre, résumé)
- Pays (**code**, nom, langue)
- Artiste (**idArtiste**, nom, prénom, annéeNaissance)
- Internaute (**email**, nom, prénom, région)

Remarque : On ne s'intéresse pas pour l'instant, aux liens entre les relations.

- **Règles de passage : associations de un à plusieurs**

Soit une association de un à plusieurs entre A et B.



Le passage au modèle logique suit les règles suivantes :

1. On crée les relations R_A et R_B correspondant respectivement aux entités A et B ;
2. On détermine l'entité faible (ici A) ;
3. L'identifiant de B devient un attribut de R_A ;
4. L'idée est qu'une occurrence de A « référence » l'occurrence de B qui lui est associée à l'aide d'une *clé étrangère*. Cette référence se fait de manière unique et suffisante à l'aide de l'identifiant.

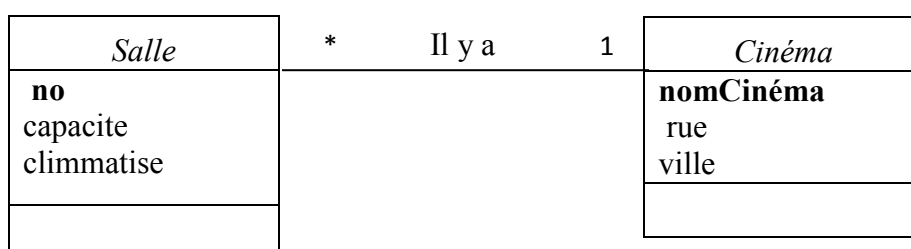
Exemple 1:

De l'exemple 3 ci-dessus, on obtient le schéma représentant l'association entre les types d'entité Film, Artiste et Pays. Les clés étrangères sont en italiques.

- Film (**idFilm**, titre, année, genre, résumé, *idArtiste*, *codePays*)
- Artiste (**idArtiste**, nom, prénom, annéeNaissance)
- Pays (**code**, nom, langue)

Exemple 2 :

Soit l'E/A suivant représentant l'association entre les types d'entité Cinéma et Salle



On a

- Cinéma (**nomCinéma**, numéro, rue, ville)
- Salle (**nomCinéma**, **no**, capacité)

Remarque : On note que l'identifiant d'une salle est constitué de l'identifiant du cinéma, et d'un numéro complémentaire permettant de distinguer les salles au sein d'un même cinéma. La clé étrangère est donc une partie de la clé primaire.

- **Règles de passage : associations binaires de plusieurs à plusieurs**

Soit une association binaire n-m entre A et B.

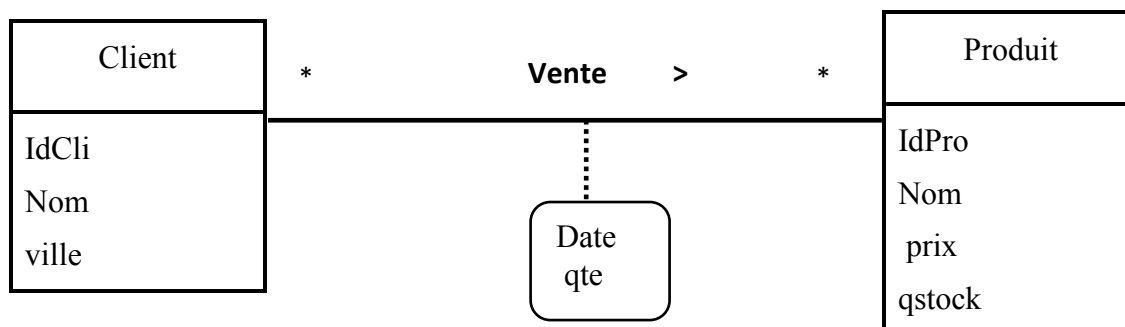


Le passage du modèle logique suit les règles suivantes :

1. On crée les relations R_A et R_B correspondant respectivement aux entités A et B ;
2. On crée une relation R_{A-B} pour l'association ;
3. La clé de R_A et la clé de R_B deviennent des attributs de R_{A-B} ;
4. La clé de cette relation est la concaténation des clés des relations R_A et R_B ;
5. Les propriétés de l'association deviennent des attributs de R_{A-B} ;

Exemple :

Soit le modèle E/A ci-dessous



On a le schéma relationnel correspondant suivant :

- CLIENT (IdCli, nom, ville)
- PRODUIT (IdPro, nom, prix, qstock)
- VENTE (IdCli, IdPro, date, qte)

Représentation des données sous forme de tables

CLIENT	IdCli	Nom	Ville
	X	Smith	Paris
	Y	Jones	Paris
	Z	Blake	Nice

PRODUIT	IdPro	Nom	Prix	Qstock
	P	Auto	100	10
	Q	Moto	100	10
	R	Velo	100	10
	S	Pedalo	100	10

VENTE	IdCli	IdPro	Date	Qte
	X	P		1
	X	Q		2
	X	R		3
	Y	P		4
	Y	Q		5
	Z	Q		6

2.3 Quelques avantages du model relationnel

- Simplicité de présentation: représentation sous forme de tables ;
- Possibilité de faire des opérations relationnelles : algèbre relationnelle ;
- Indépendance physique : optimisation des accès et stratégie d'accès déterminée par le système ;
- Indépendance logique : concept de VUES ;
- Conservation de l'intégrité : contraintes d'intégrité définies au niveau du schéma.

Section II : Modèles Dépendances fonctionnelles et normalisation

La normalisation permet de garantir que le schéma relationnel est un bon schéma. Elle consiste en différents stades de qualité permettant d'éviter certaines erreurs de conception qui génèrent de la redondance, la limitation ou la perte de données, l'incohérence ou l'effondrement des performances des traitements.

1. Dépendance fonctionnelle (DF)

1.1. Dépendance fonctionnelle

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation, et X et Y des sous-ensembles de $\{A_1, A_2, \dots, A_n\}$. On dit que X détermine Y (Y est déterminé par X , ou Y dépend fonctionnellement de X) si, et seulement si, des valeurs identiques de X impliquent des valeurs identiques de Y . On le note : $X \rightarrow Y$

1.2. Dépendance fonctionnelle élémentaire :

C'est une dépendance fonctionnelle de la forme $X \rightarrow Y$, où A est un attribut unique n'appartenant pas à X et où il n'existe pas X' inclus dans X tel que $X' \rightarrow A$.

2. Notion de clé

2.1. Clé

Soit E un type d'entité et A l'ensemble des attributs de E . Une clé de E est un sous-ensemble minimal de A permettant d'identifier de manière unique une entité parmi n'importe quelle extension de E .

2.2. Clé de relation

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation, et X un sous-ensemble de (A_1, A_2, \dots, A_n) , X est une clé si, et seulement si, :

$$2.3.1 \quad X \rightarrow (A_1, A_2, \dots, A_n)$$

$$2.3.2 \quad X \text{ est minimale}$$

3. Les différentes formes normales

3.1. Première forme normale (1 FN)

Une relation est en première forme normale si et seulement si tout attribut contient une valeur atomique (non multiple, non composée).

Prenons, par exemple, le pseudo schéma de relation suivant :

PERSONNE (idPersonne, nomPersonne, prénomPersonne, adresse, voitures)

Ce pseudo schéma de relation n'est pas 1 FN, car l'attribut adresse est composite et l'attribut voitures est multiple. Il faut le décomposer en :

PERSONNE (idPersonne, nomPersonne, prénomPersonne, numéroEtRue, codePostal, ville)

VOITURE (idVoiture, modèle, marque, propriétaire) où propriétaire est une clé étrangère qui fait référence au schéma relation PERSONNE.

3.2. Deuxième forme normale (2 FN)

Une relation est en deuxième forme normale si et seulement si :

- elle est en première forme normale (1 FN) ;
- tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé.

Une relation peut être en deuxième forme normale par rapport à une de ses clés candidates et ne pas l'être par rapport à une autre. Une relation avec une clé primaire réduite à un seul attribut, ou contenant tous les attributs, est forcément en deuxième forme normale.

Soit, par exemple, le schéma de relation suivant :

AFFECTATION (idPersonne, idEtablissement, nomPersonne, prénomPersonne, nomEtablissement)

Supposons qu'une personne puisse être affectée à plusieurs établissements et que dans un établissement sont affectées plusieurs personnes. Cette relation n'est pas en 2FN car, par exemple, nomPersonne ne dépend que de idPersonne. Pour normaliser cette relation, il faut la décomposer de la manière suivante :

PERSONNE (idPersonne, nomPersonne, prénomPersonne)

ETABLISSEMENT (idEtablissement, nomEtablissement)

AFFECTATION (idPersonne, idEtablissement) où idPersonne et idEtablissement sont des clés étrangères qui font respectivement référence aux schémas de relation PERSONNE et ETABLISSEMENT.

3.3. Troisième forme normale (3 FN)

Une relation est en troisième forme normale si et seulement si :

- elle est en deuxième forme normale ;
- tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non-clé.

3.4. Forme Normale de BOYCE-CODD (BCFN)

Une relation est en Forme normale de BOYCE-CODD si, et seulement si, les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut.

Activités d'apprentissage

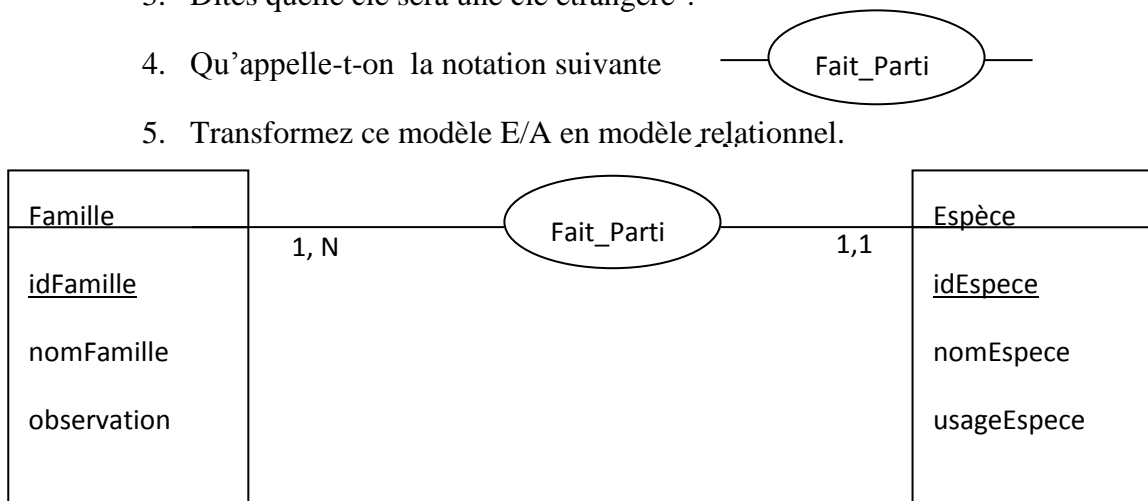
Partie A

1. Qu'est-ce qu'une base de données ?
2. A quoi sert-elle la clé primaire d'une table ?
3. donnez la définition d'une entité et une occurrence d'entité.
4. donnez la définition d'une occurrence d'entité.
5. Quand parle-t-on de clé primaire et quand parle-t-on d'identifiant dans un système informatique.
6. qu'est-ce qu'un état sous Access?

Exercice 2 :

A partir du modèle conceptuel(MCT) de données suivant, vous allez élaborer le modèle physique de données (MPD) correspondant. Pour cela :

1. Déterminez les clés primaires ?
2. Déterminez l'entité faible et l'entité forte ?
3. Dites quelle clé sera une clé étrangère ?
4. Qu'appelle-t-on la notation suivante
5. Transformez ce modèle E/A en modèle relationnel.



Exercice 3 :

A partir de ce schéma déterminer les relations possibles :

