



Conception des Bases de données

Introduction aux Bases de données

Qu'est ce qu'une base de données ?

- Collection de données (structurées ou non structurées) suivant une réalité modélisée

Où trouve-t-on des bases de données ?

- web: sites marchands, réseaux sociaux, ...
- finance: applications financières, gestion de comptes, ...
- économie : e-commerce (amazon), services de ventes/achats, ...
- industrie: gestion de centrales nucléaires, chaînes de production, ...
- transports: réservation de billets, gestion de trains/avions, ..
- science: données d'expérimentation,...
- services publiques: impôts, police, open-data, ...

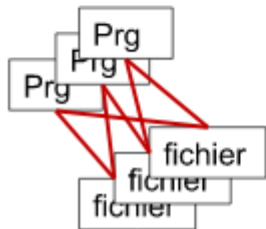
Quels sont les types de Bases de Données ?

- Les BD relationnelles (prédominantes : données de gestion)
- Les BD objet, NoSQL (orientées web, orientées documents, données non structurées)

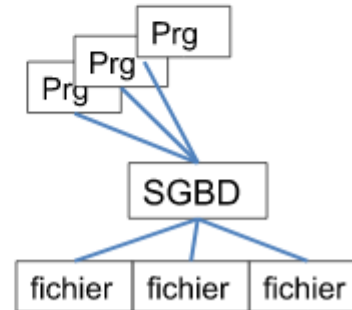
Interêts d'un SGBDR

SGBD :

- Contient les BDs ainsi que leur description et contraintes (méta-données stockes dans le catalogue du SGBD)
- Centralisation de l'information, représentation de relations complexes entre les données, permet de trouver et de mettre à jour efficacement des données reliées
- Contrôle de la redondance des données
- Contrôle des niveaux d'accès aux données (sécurité des données)
- Stockage persistant pour les données des applications, structures de stockage et interrogation efficace des données
- Abstraction des données, isolation entre les données et les traitements
- Plusieurs vues des mêmes données, interfaces utilisateur multiples
- Accès simultané aux données centralisées par plusieurs applications, contrôle de concurrence, partage de données
- Définition de contraintes d'intégrité et automatisation de leur maintenance (triggers)



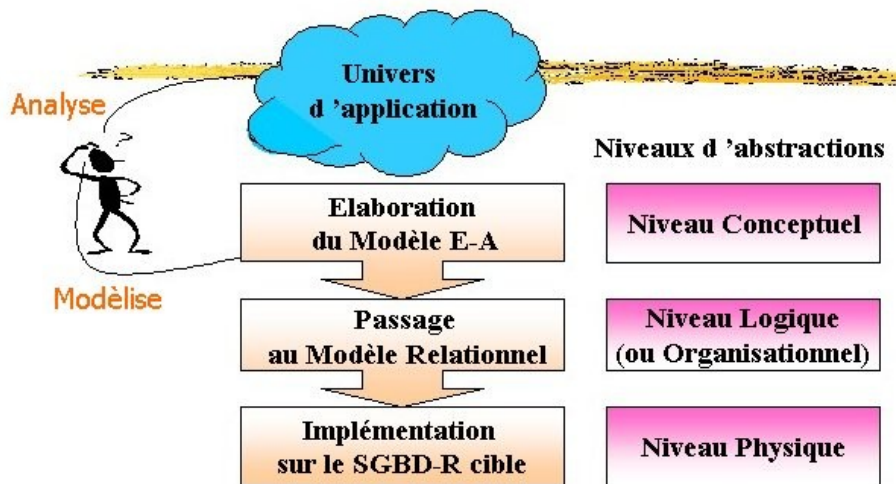
(*) Système de Gestion de Bases de Données
Database Management System (DBMS)



Une architecture à 3 Niveaux

Les méthodes de conception préconisent une démarche en étapes et font appel à des **modèles** pour représenter les **objets** qui composent les systèmes d'information, les **relations** existantes entre ces objets ainsi que les **règles sous-jacentes**.

La démarche de Conception



La modélisation se réalise en trois étapes principales qui correspondent à trois niveaux d'abstraction différents :

I. Niveau conceptuel : représente le contenu de la base en termes conceptuels, indépendamment de toute considération informatique.

II. Niveau logique relationnelle : résulte de la traduction du schéma conceptuel en un schéma propre à un type de BD.

III. Niveau physique : est utilisé pour décrire les méthodes d'organisation et d'accès aux données de la base.

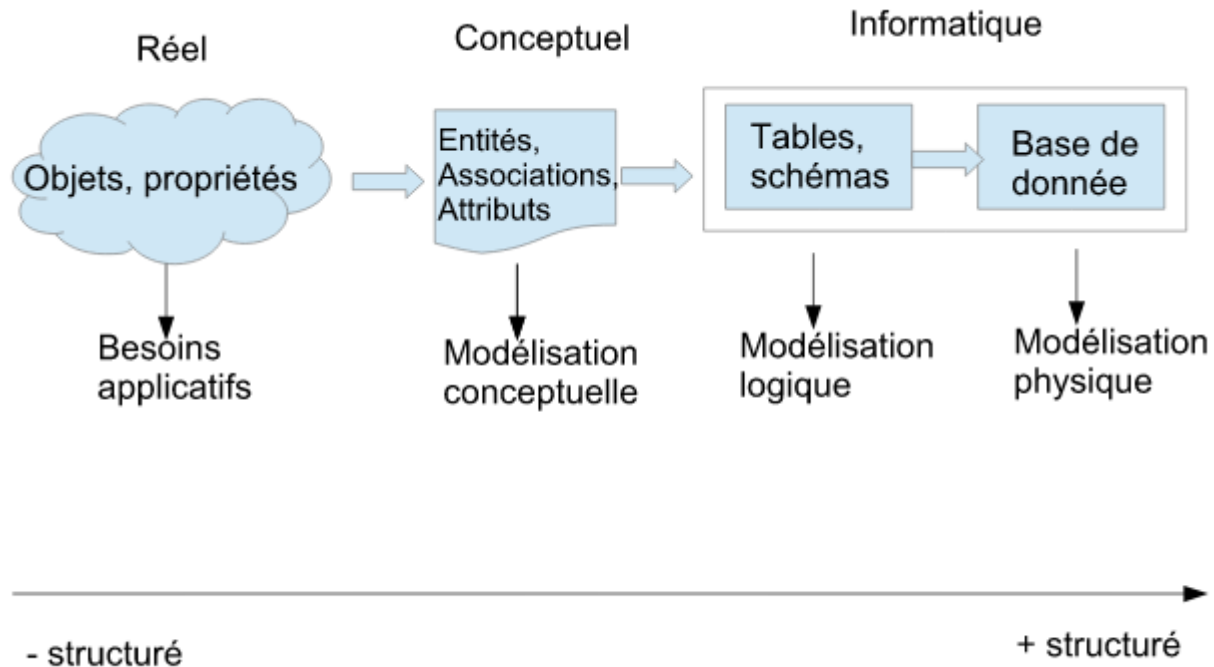
Comment construire une Base de données ?

- 1 Analyse des besoins
observer le monde réel
identifier les informations pertinentes
- 2 Modélisation des données
formaliser les besoins
- 3 Implantation des données et des procédures
d'interrogation et de mise à jour
BD proprement parlé



```
...provider
SelectSQL = * Database provider *
QuerySQL = * select id, name, email, phone
QuerySQL = * where id between 1 and 100
SelectQuery = SelectSQL + QuerySQL + GroupBy
Execute Query: Commit Transaction: select on the
Form Navigation
IF KeyAscii = 13 Then Execute Query
IF KeyAscii Like "F" And Not...
Not Chr(KeyAscii) Like "F" And Not...
```

Comment construire une Base de données ?



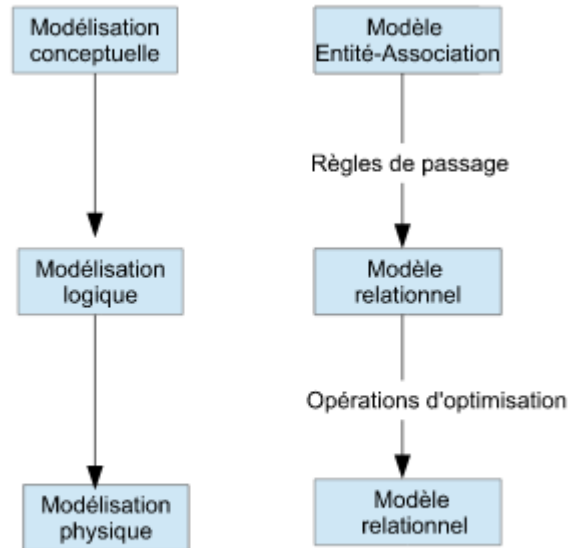
Modèle de données

Modèle entité-association

- Description haut niveau des données
 - entité et les liens entre elles (associations)
- ⇒ concepteur, client

Modèles relationnel

- Représentation logique des données
 - Concept de tables
- ⇒ concepteur, développeur



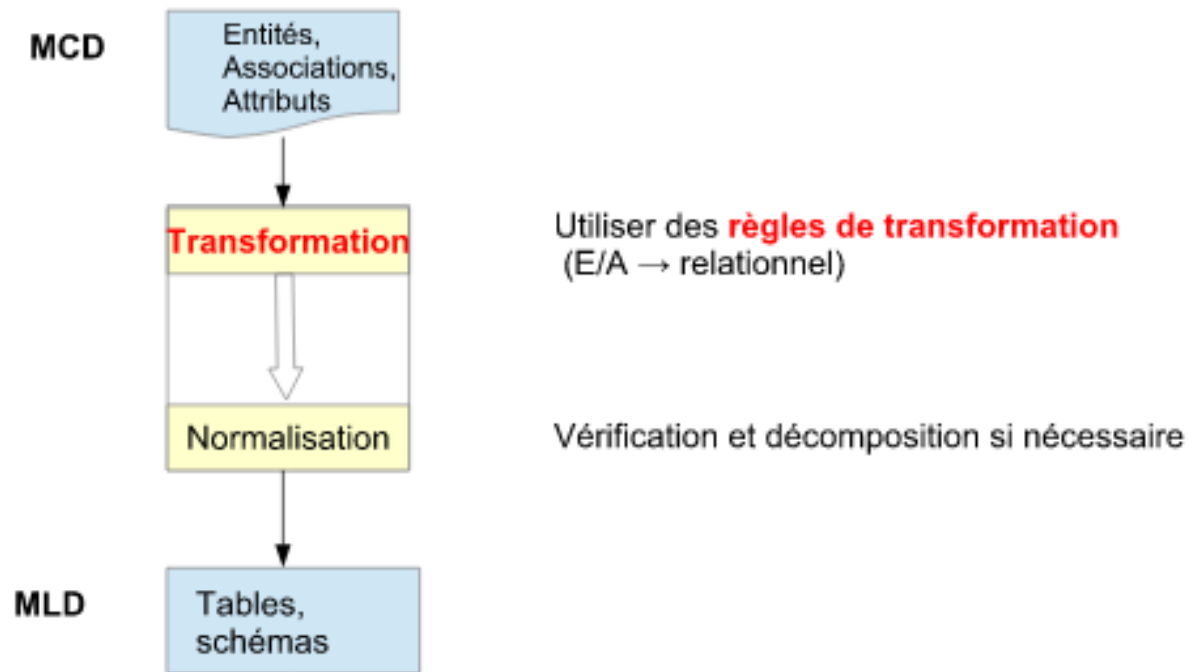
Modèle de données

- **Modèle conceptuel des données (MCD):** description de l'application dans un langage de haut niveau (Entité-Association) qui ne tient pas compte du SGBD
- **Modèle logique des données (MLD):** description des données dans un formalisme compatible avec un SGBD (schemas, tables, colonnes, clés primaires et étrangères)
- **Modèle physique des données (MPD):** implémentation du modèle logique dans le SGBD (affiner le MLD en un schéma pour un SGBD spécifique), utilisation de SQL (create TABLE..), types des attributs, index, dénormalisation

Modèle de données

Transformation MCD

MLD





Modélisation Conceptuelle schéma Entité-Association ou Entité-Relation

Démarche

Analyse des besoins

- Discussion **informelle** avec les futurs utilisateurs
 - Identifier les objets du monde réel et des liens entre eux
 - Identifier les opérations sur ces objets et les éventuelles évolutions
- Document technique décrivant les données de l'application

Etablissement du schéma conceptuel

- Langage de **haut niveau** (ex. le modèle entité-association)
 - Décider des données devant être stockées, de leur propriétés et des relations entre elles
 - Définir les contraintes à respecter
- Schéma dans un langage de haut niveau (Entité-Association)

Etude de cas: BD d'une Université

Les besoins :

- Gérer les inscriptions des étudiants à des modules
- Gérer l'affectation des tuteurs à des étudiants
- Gérer le planning des salles

Les objets à modéliser :

- Les étudiants Les modules
- Les tuteurs
- Les salles

Les liens entre les objets (scénario) :

- Les étudiants s'inscrivent à un ou plusieurs modules pour une année universitaire
- Le cours d'un module a lieu dans une salle donnée ; il débute à une heure connue et se déroule pendant une durée connue.

Le Modèle Entité Association

- **Principe** : Transcrire les besoins en terme de *classes d'entités* et de *classes d'associations*
- Les *entités* = les objets du monde réel
- Une classe d'entités = ensemble d'entités possédant les mêmes propriétés
- Les *associations* = les relations liant les entités
- Une classe associations = ensemble d'associations reliant des entités de la même classe
- Les *attributs* = les propriétés qui renseignent certaines informations sur une entité ou une associatio

Les éléments de base du modèle ER (Entité-Relation) ou E-A (Entité-Association)

- 1- Les entités
- 2- Les attributs
- 3- Type de relation :
cardinalités
- 4- L'identifiant

Le Modèle Entité Association: Entité

Exemple d'entités concrètes :

- L'artiste XXX est une entité
- La salle 24-34/208 est une entité
- Le cours "Introduction aux bases de données" est une entité

Exemple d'entités abstraites (qui ne correspondent pas à des objets physiques) :

- Le virement n° XXX ayant eu lieu le 20/01/20 est une entité
- Le compte en banque d'un client est une entité
- Un contrat d'assurance est une entité

Classe d'entités : Permet de décrire un ensemble d'entités de même type (ayant les mêmes caractéristiques)

Exemple de classes d'entités :

- La classe Étudiant décrit l'ensemble des étudiants de l'université (tout étudiant a un nom, prénom, adresse, matricule)
- La classe Module décrit toutes les modules de l'université (tout module a un code, un intitulé et un niveau)
- La classe Prof décrit tous les Profs de l'application (tout prof a un matricule et un nom)
- etc

Le Modèle Entité Association: Attribut

Attribut :

- * Modélise une propriété/caractéristique d'une classe d'entités, et peut être obligatoire ou facultatif
- * Possède un nom et un domaine de valeurs (=ensemble de valeurs possibles)
- * Est atomique ,il prend une seule valeur à la fois pour chaque entité

Exemple d'attributs :

- Le nom, le prénom, l'adresse et le matricule sont des attributs de la classe Étudiant
- Le nom est une chaîne de 1 à 20 caractères et l'attribut correspondant est Nom.
- Le matricule est un nombre entier avec 4 chiffres
- L'attribut prénom ne peut pas contenir plusieurs chaînes de caractères (e.g"Jean", "Louis") mais une seule chaîne de caractères (e.g "Jean, Louis")
- Le nom de l'étudiant de matricule 1234 (=une entité) ne peut pas être à la fois "Dupont" et "Martin"

Le Modèle Entité Association: Relation

Les relations : représentent les liens existants entre les entités.

- Contrairement aux entités, les relations n'ont pas de relations propres. Les relations sont caractérisées, comme les entités, par un nom et éventuellement des attributs.

- La dimension ou degré de la relation est le nombre d'entités impliquées dans une relation. Les relations peuvent être de :

1- **Dimension I** (ou réflexive) : elle ne concerne qu'une entité type dont elle relie deux éléments.

Ex : La relation Mariage relie deux éléments de l'entité PERSONNE

2- **Dimension II** : c'est la plus fréquente.

Ex : la relation entre la commande et le client est une relation binaire ou de dimension 2.

3- **Dimension III** : une relation est ternaire dans la mesure où elle ne peut être décomposée en deux ou trois relations binaires équivalentes.

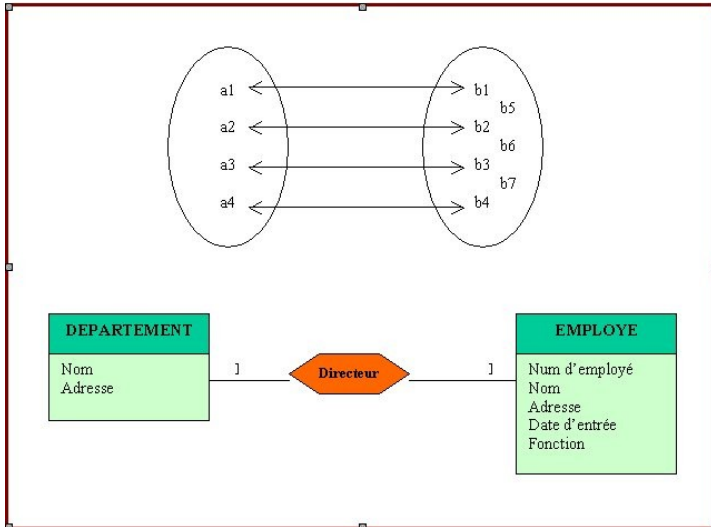
Ex : une location de voiture représente une relation entre un véhicule, une personne et une date.

=> De manière générale, une relation peut être caractérisée par n dimensions.

Le Modèle Entité Association: Cardinalité

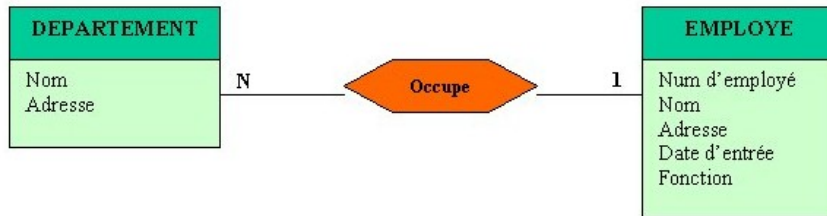
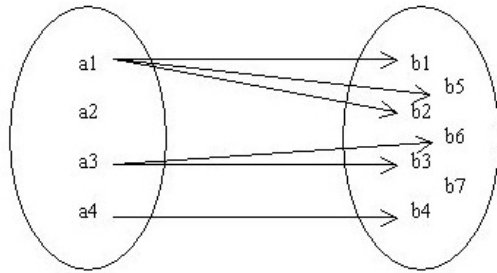
Cardinalité : la description complète d'une relation nécessite la définition précise de la participation des entités. La cardinalité est le nombre de participation d'une entité à une relation.

Cardinalité un à un : si et seulement si un employé ne peut être directeur que dans un seul département et un département n'a qu'un seul employé comme directeur.

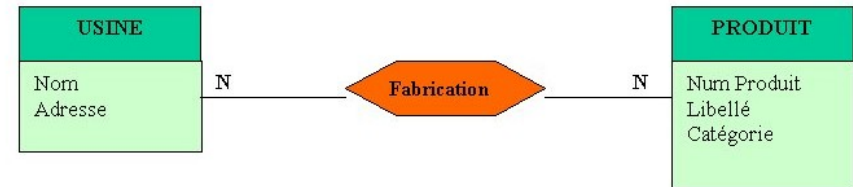
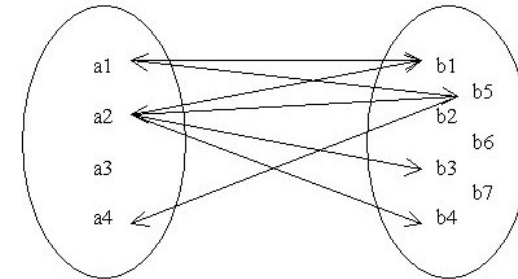


Le Modèle Entité Association: Cardinalité

Cardinalité un à plusieurs : un département peut occuper plusieurs employés qui réalisent différentes fonctions mais chaque employé ne fait partie que d'un seul département.



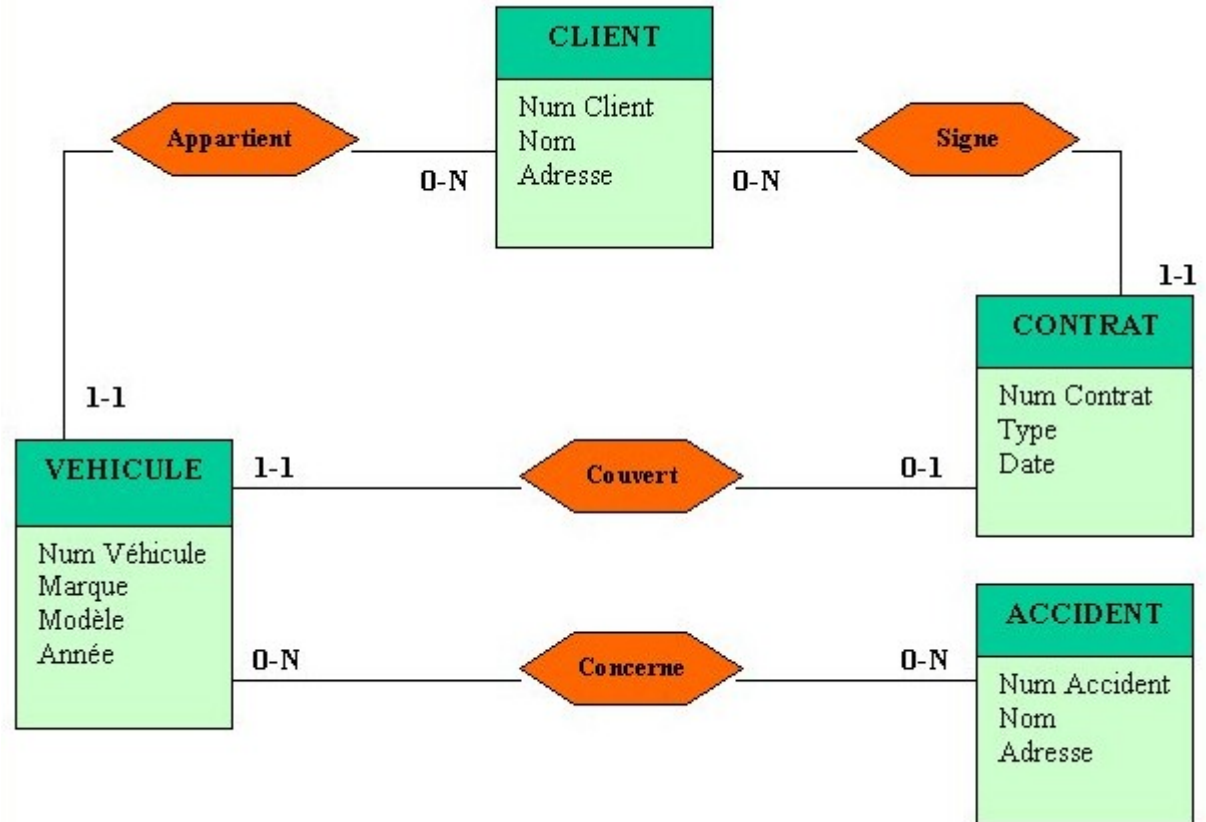
Cardinalité plusieurs à plusieurs : un type de produit peut être fabriqué en plusieurs usines et une usine donnée peut fabriquer plusieurs types de produits.



Le Modèle Entité Association: Cardinalité

Les cardinalités maximales et minimales traduisent les contraintes propres aux entités et relations. Dans un schéma conceptuel, elles sont représentées comme suit :

- 0-1 aucune ou une seule
- 1-1 une et une seule
- 0-N aucune ou plusieurs
- 1-N une ou plusieurs



Le Modèle Entité Association: Identifiant

Identifiant :

* Un attribut ou un sous ensemble d'attributs d'une classe d'entités permettant de déterminer une et une seule entité à l'intérieur de l'ensemble.

* Permet de distinguer les entités de la même classe (deux entités de la même classe ne peuvent pas avoir le même identifiant)

* Toute classe d'entités doit avoir au moins un identifiant (si plusieurs identifiants potentiels → en choisir un seul)

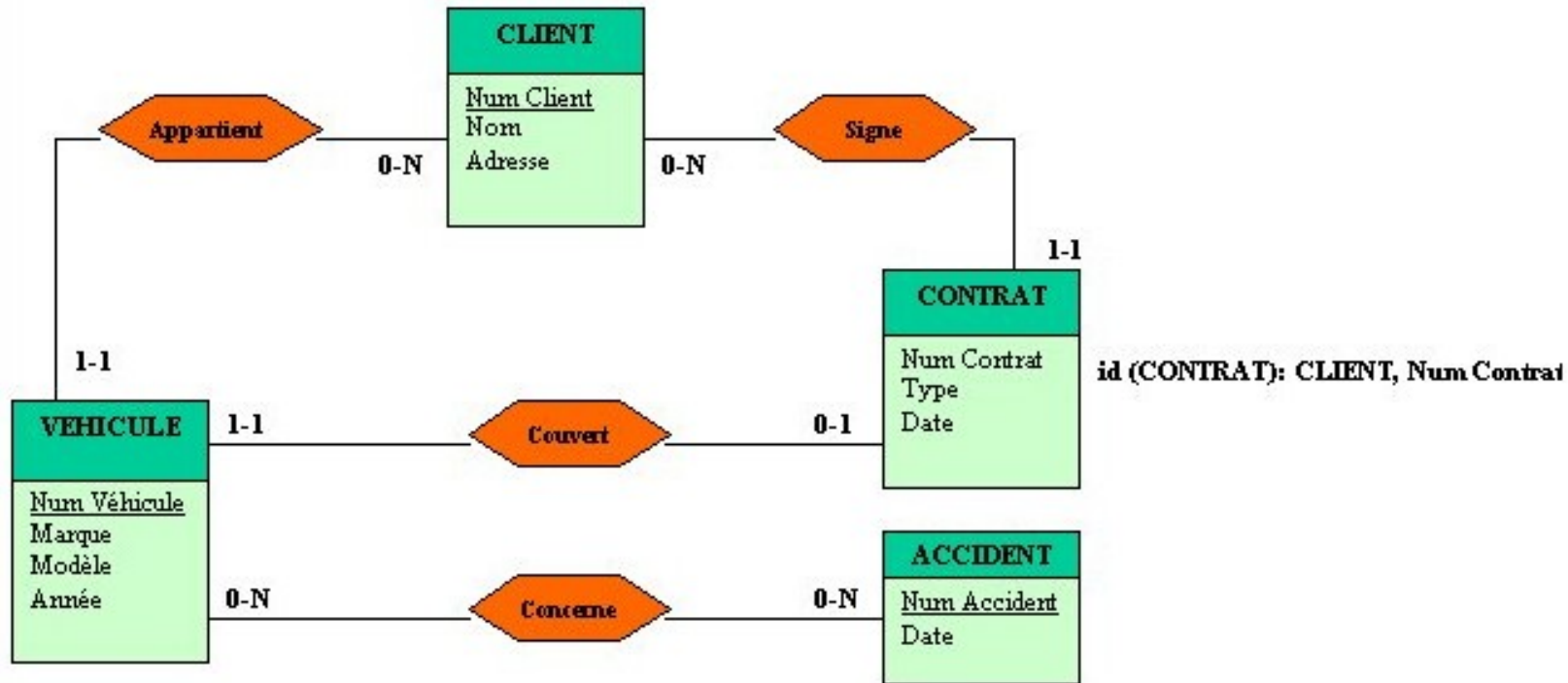
Graphiquement les identifiants sont les attributs soulignés. L'entité faible aura un identifiant composé de l'identifiant de l'entité dont elle dépend et d'un autre attribut.

* Un identifiant peut être :

- Naturel (construit à partir des attributs de la classe)

- Artificiel (rajouté aux attributs de la classe, lorsque les attributs de la classe ne permettent pas de définir un identifiant)

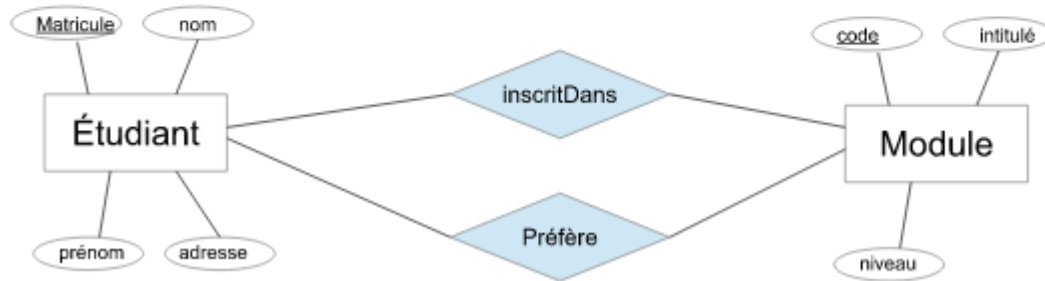
Le Modèle Entité Association: Identifiant



Le Modèle Entité Association

Classe d'associations

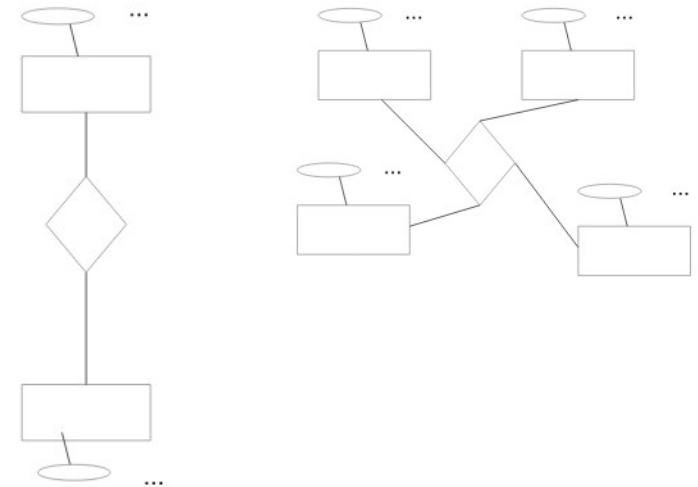
- L'ensemble d'associations de même type (qui relient les mêmes ensembles d'entités et qui ont la même sémantique)
- Désignée généralement par un verbe
- Exemple (représentation graphique) :



Types de classes d'association

Associations binaires (relient deux entités)

Associations N-aires (relient plus de deux entités)



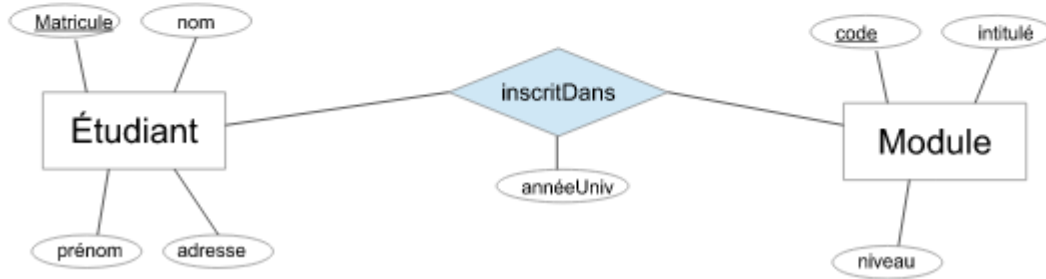
Retour à la BD d'une Université

Attributs d'une classe d'associations

Identifiant: L'ensemble des identifiants des classes d'entités qu'elle relie

Autres attributs: l'ensemble des valeurs des attributs des associations de cette classe

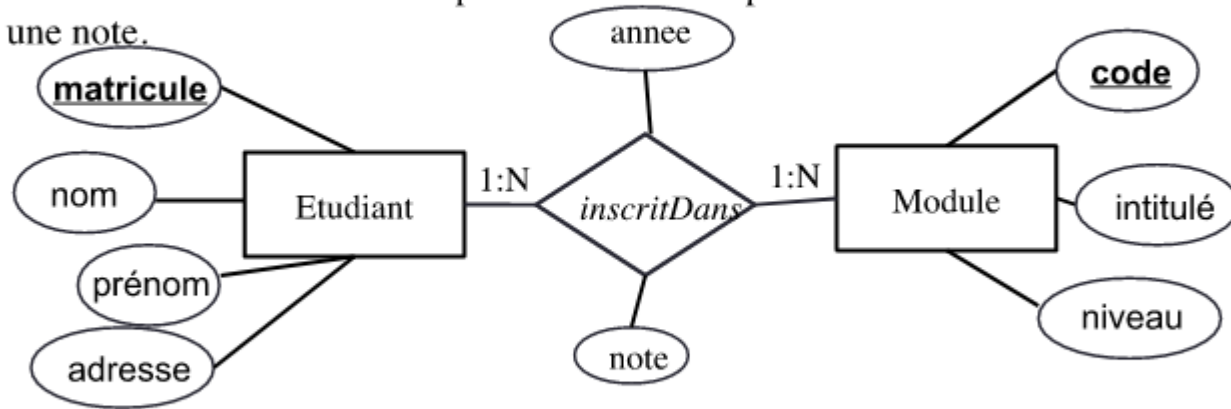
Exemple: Etudiant *inscritDans* Module pour une année



Retour à la BD d'une Université

Attribut d'une classe d'association

Un étudiant s'inscrit à un ou plusieurs modules pendant une année et obtient une note.

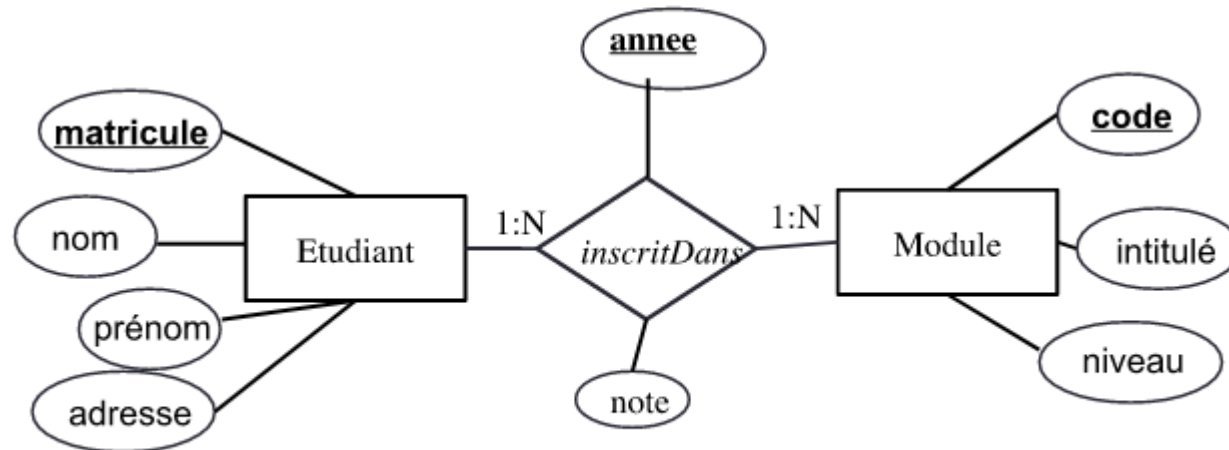


etudiants	modules	Année	note
1753	LI341	2012	9
0148	LI345	2012	13

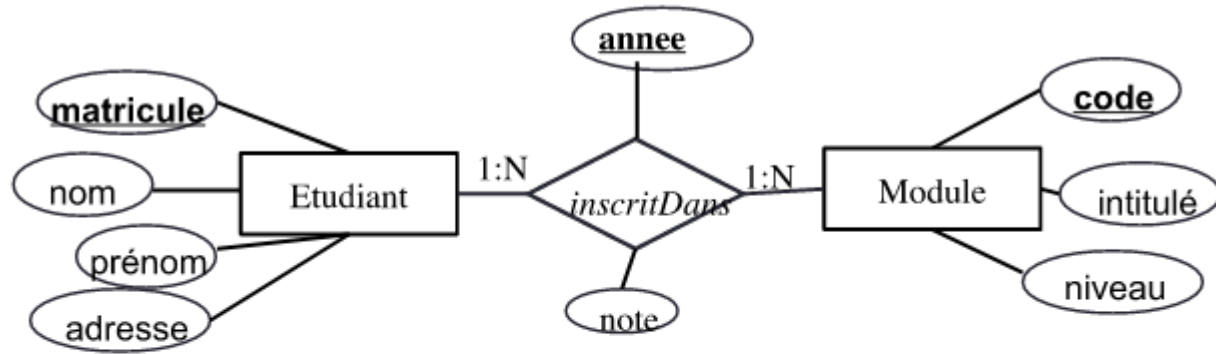
Retour à la BD d'une Université

Identifiant d'une classe d'associations

Un étudiant s'inscrit à un ou plusieurs modules pendant une année ou plusieurs années et obtient à chaque inscription une note.



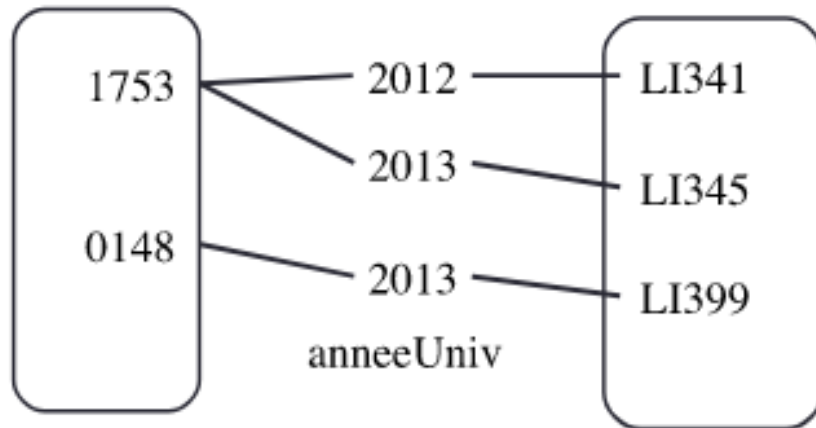
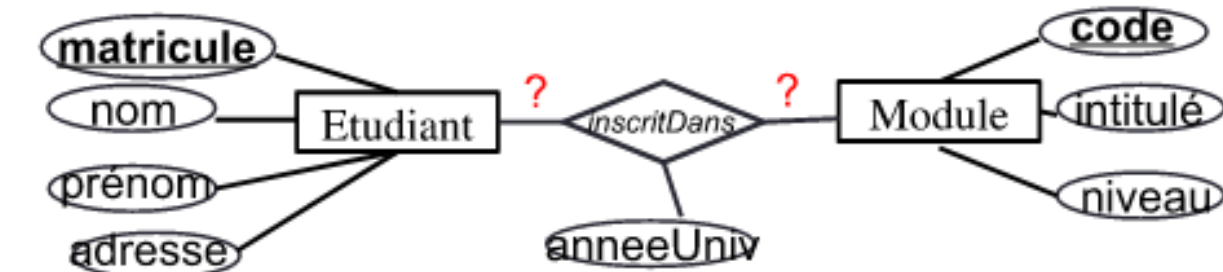
Retour à la BD d'une Université



etudiants	modules	Annee	note
1753	LI341	2012	9
1753	LI341	2013	10.5
0148	LI345	2012	13

- **Entité** sous-entend classe d'entités
- **Association** sous-entend classe d'associations
- Au niveau E/A, on se préoccupe seulement des classes, pas des instances particulières

Retour à la BD d'une Université



Les Etudiants

Les Modules

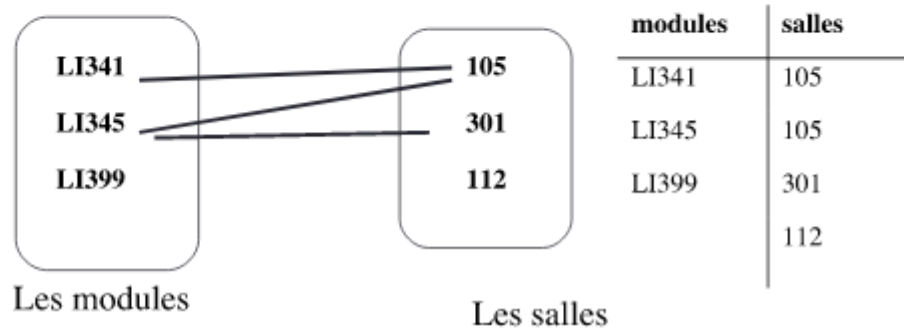
etudiants	modules	anneeUniv
1753	LI341	2012
1753	LI345	2013
0148	LI399	2013

Retour à la BD d'une Université

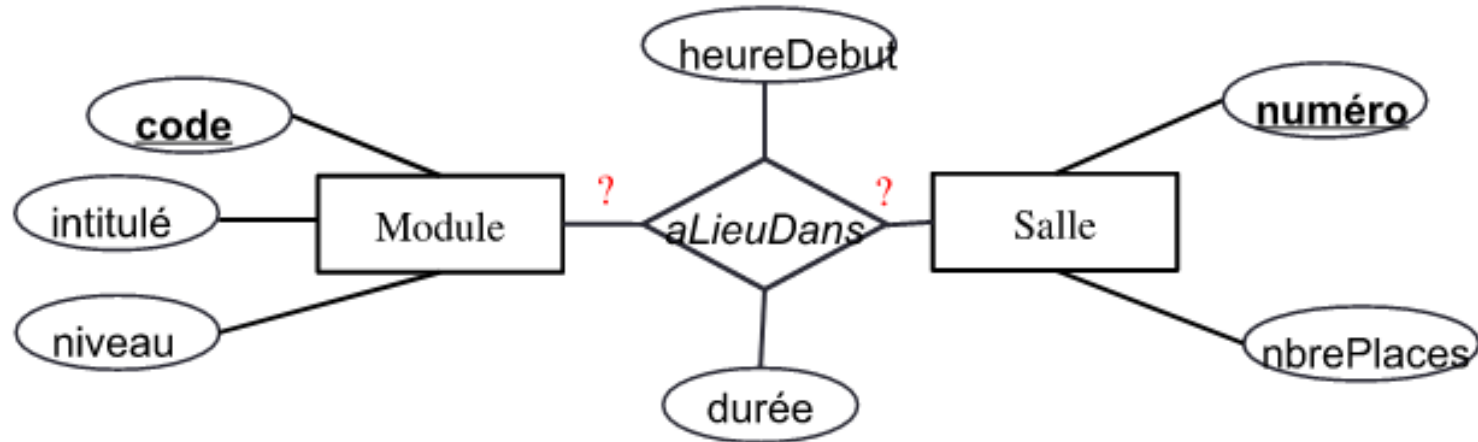
Les cardinalités d'associations

- Module *aLieuDans* Salle
 - Un module a lieu dans une et une seule salle
 - Une salle peut être utilisée pour plusieurs modules ou rester inoccupée

L'association « Module *aLieuDans* Salle »



Retour à la BD d'une Université



modules	salles	heureDebut	duree
LI341	105	830	120
LI345	105	1030	90
	214		

Retour à la BD d'une Université

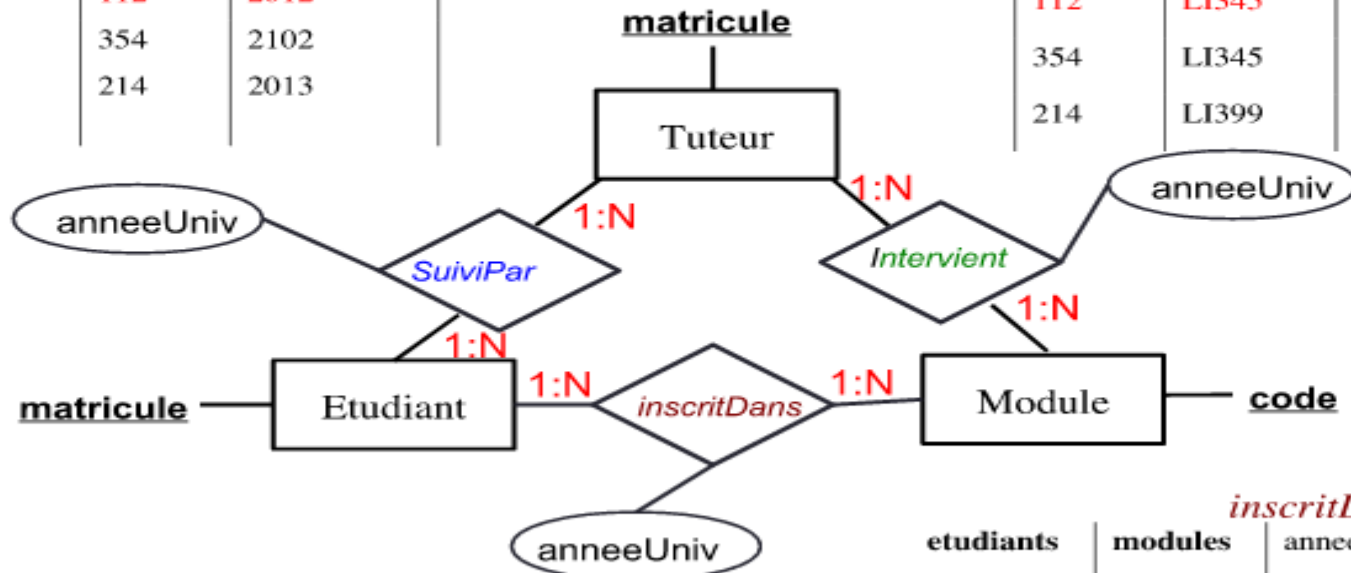
Trois associations binaires

SuiviPar

etudiants	tuteurs	anneeUniv
1753	112	2012
1753	354	2102
0148	214	2013

Intervient

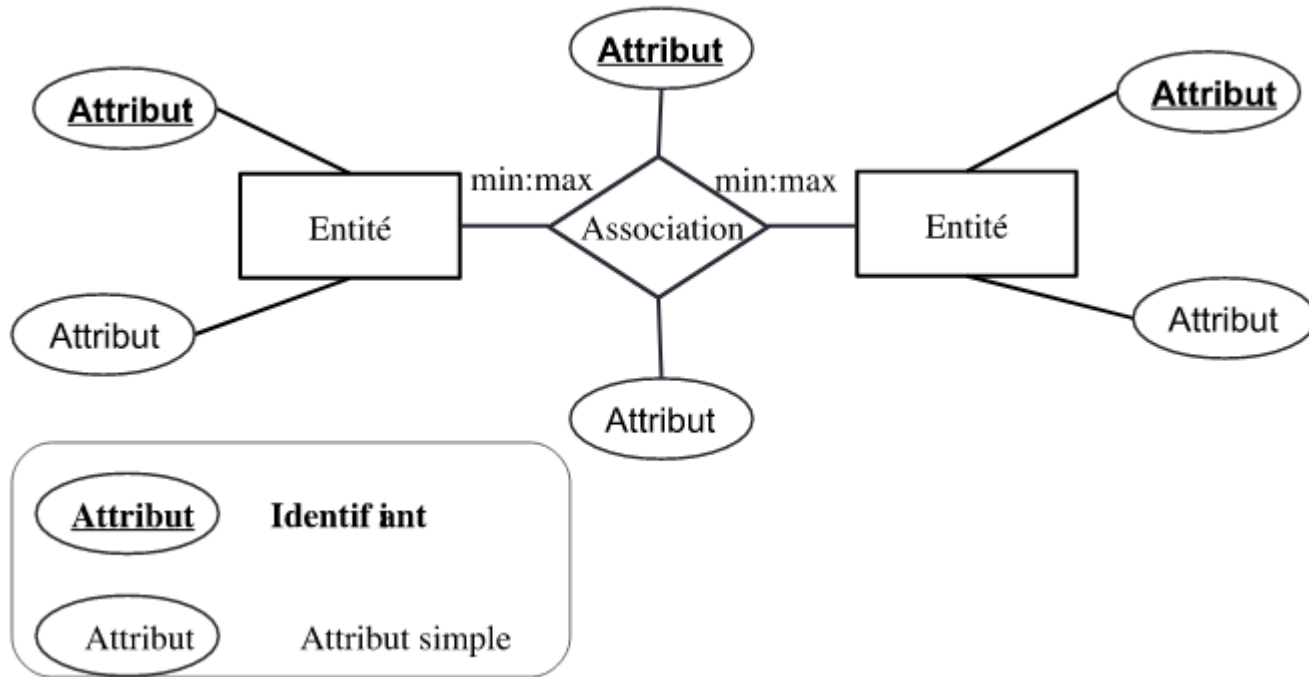
tuteurs	modules	anneeUniv
112	LI345	2012
354	LI345	2102
214	LI399	2013



- Un étudiant peut être inscrit dans un module et être suivi par un tuteur qui n'intervient pas dans le module !
- On peut aussi enlever l'inscription d'un étudiant à un module alors que le tuteur existe toujours !

etudiants	modules	anneeUniv
1753	LI341	2012
1753	LI345	2102
0148	LI399	2013

Le Modèle Entité Association



Choix de Conception

Analyse des besoins produit une spécification
peu précise

→ plusieurs choix de conception possibles

Questions fréquentes :

1. Un objet du monde réel peut-il être modélisé par une entité ou par un attribut ?
2. Un objet du monde réel peut-il être modélisé par une entité ou par une association ?
3. Un attribut décrit-il une association ou une entité ?

Choix de Conception

Entité ou attribut ?

Question. Pour renseigner l'adresse d'un étudiant

- 1) rajouter un attribut adresse à l'entité étudiant ? Ou
- 2) introduire une nouvelle entité, adresse, ayant comme attributs numéro, voie, code postal ?

Réponse. Décision relativement facile à prendre si l'on connaît l'application et son évolution.

Entité ou association ?

Règle générale : toute action impliquant deux entités donne lieu à une association.

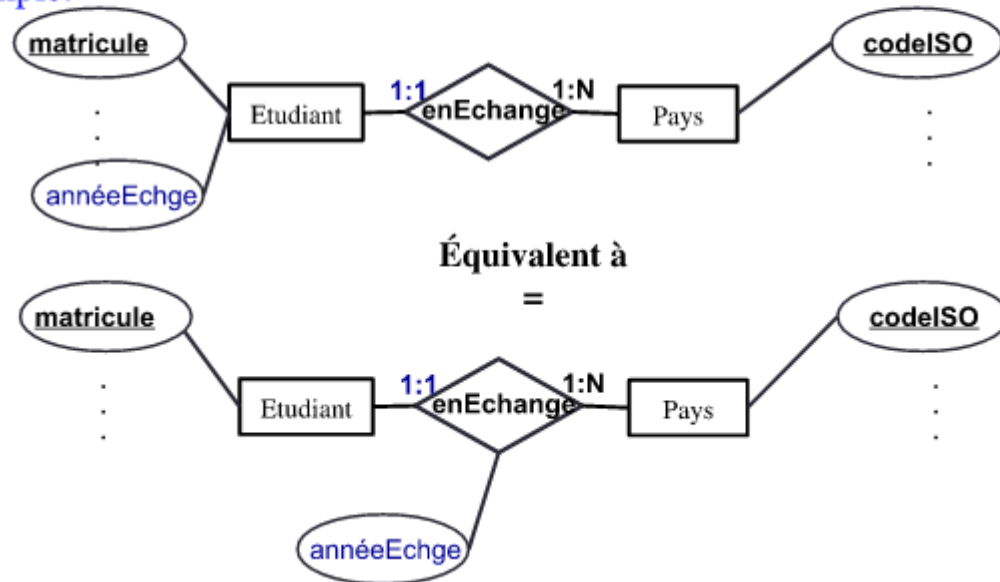
Exemple: (le cours d'un) module a lieu dans une salle → association
ALieuDans

Choix de Conception

Attribut d'une association ou d'une entité? Le choix dépend des cardinalités

- Cardinalité **1 à plusieurs** : les deux alternatives sont équivalentes puisque l'une des deux entités participe une seule fois dans l'association
- Cardinalité **plusieurs-à-plusieurs** : la sémantique diffère selon le cas où l'attribut est au niveau de l'entité ou de l'association

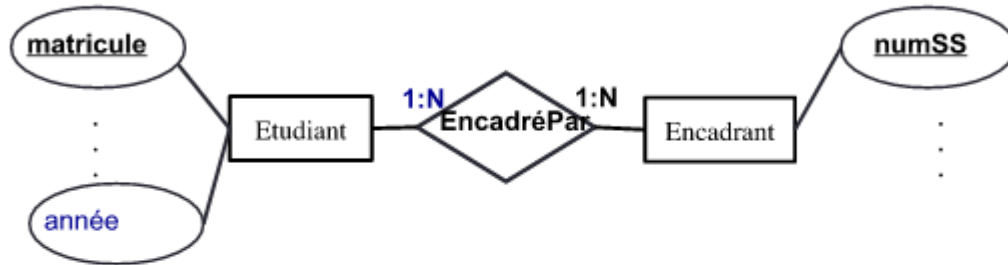
Exemple:



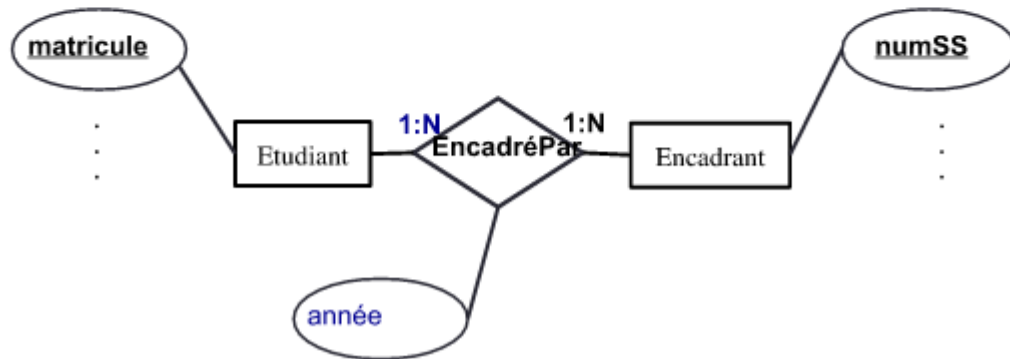
Choix de Conception

Attribut d'association ou d'entité?

Cardinalité plusieurs-à-plusieurs



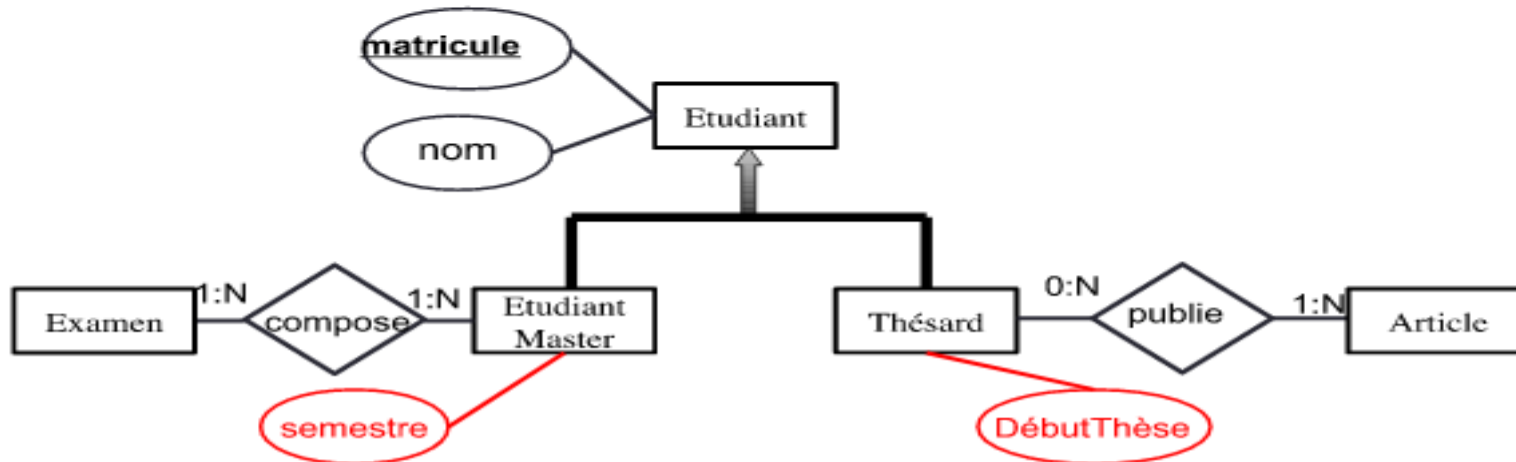
Différent de



Choix de Conception

Spécialisation

- Utile lorsque les objets à modéliser partagent certaines propriétés et possèdent d'autres propriétés propre à eux
- **Principe** : créer une entité avec les propriétés en commun dont vont *hériter* des propriétés plus spécifiques
- **Exemple** : il peut y avoir deux types d'étudiants
 - 📦 *Etudiants en master* passent des examens
 - 📦 *Etudiants en thèse* publient des articles scientifiques




Choix de Conception

La construction d'un schéma conceptuel peut se réaliser de la manière suivante :

1. Déterminer la liste des entités.
2. Pour chaque entité :
 - a) établir la liste de ses attributs ;
 - b) parmi ceux-ci, déterminer un identifiant.
3. Déterminer les relations entre les entités.
4. Pour chaque relation :
 - a) dresser la liste des attributs propres à la relation ;
 - b) vérifier la dimension (binaire, ternaire, etc.) ;
 - c) définir les cardinalités.
5. Vérifier le schéma obtenu, notamment :
 - a) supprimer les transitivités ;
 - b) s'assurer que le schéma est connexe ;
 - c) s'assurer qu'il répond aux demandes.
6. Valider avec les utilisateurs.

Application

Voir Projet 1 Bibliothèque



Modélisation Logique relationnelle

Passage du modèle E/A vers le modèle relationnel

Exemple de table de base de données

SALARIE			
Matricule	Nom	Grade	Salaire
100	Müller	cadre	12'000
101	Roche	employé	4'500
102	Chapuis	assistant	4'000

⇒ **Nom table**

⇒ **Attributs**

⇒ **Tuples**

⇒ **Domaines**

Table SALARIE (Code : entier,
Nom : chaîne de caractères,
Grade : {cadre, employé, assistant},
Salaire : [12'000-4'000])

Degré de la table : 4

Clé Primaire et Clé Candidate

- Chaque relation doit posséder *au moins une clé*
- Une relation a *au moins une clé candidate* (*chacun des attributs est renseigné, pas de valeurs NULL*)
 - on choisit *une seule* comme *clé primaire*
- Exemple: matricule est clé primaire, {*nom, prénom, dateNaiss*} est clé candidate

Chaque ligne a une valeur de matricule et de l'ensemble {*nom, prénom, dateNaiss*} différente

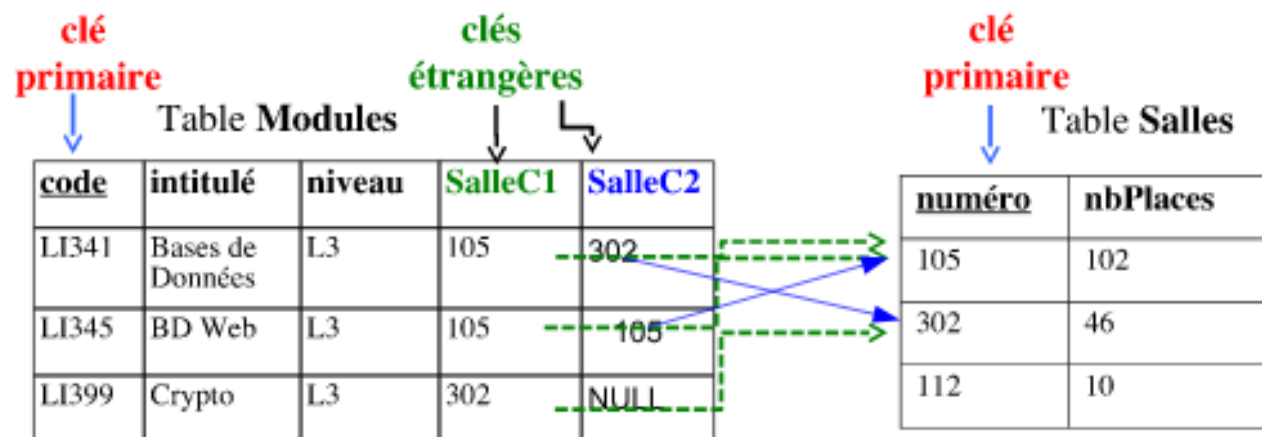
<u>matricule</u>	nom	prénom	dateNaiss	adresse
1753	Smith	Joe	1992-01-12	11 CP NYC
9832	Smith	Dan	1989-04-03	22 Rd NJ
4755	Smith	Joe	1994-11-29	7 HW NJ
6842	Roy	Ian	1992-05-18	NULL

- Notation: la **clé primaire** est soulignée, les **clés candidates** sont mentionnées en langage naturel

Clé étrangère

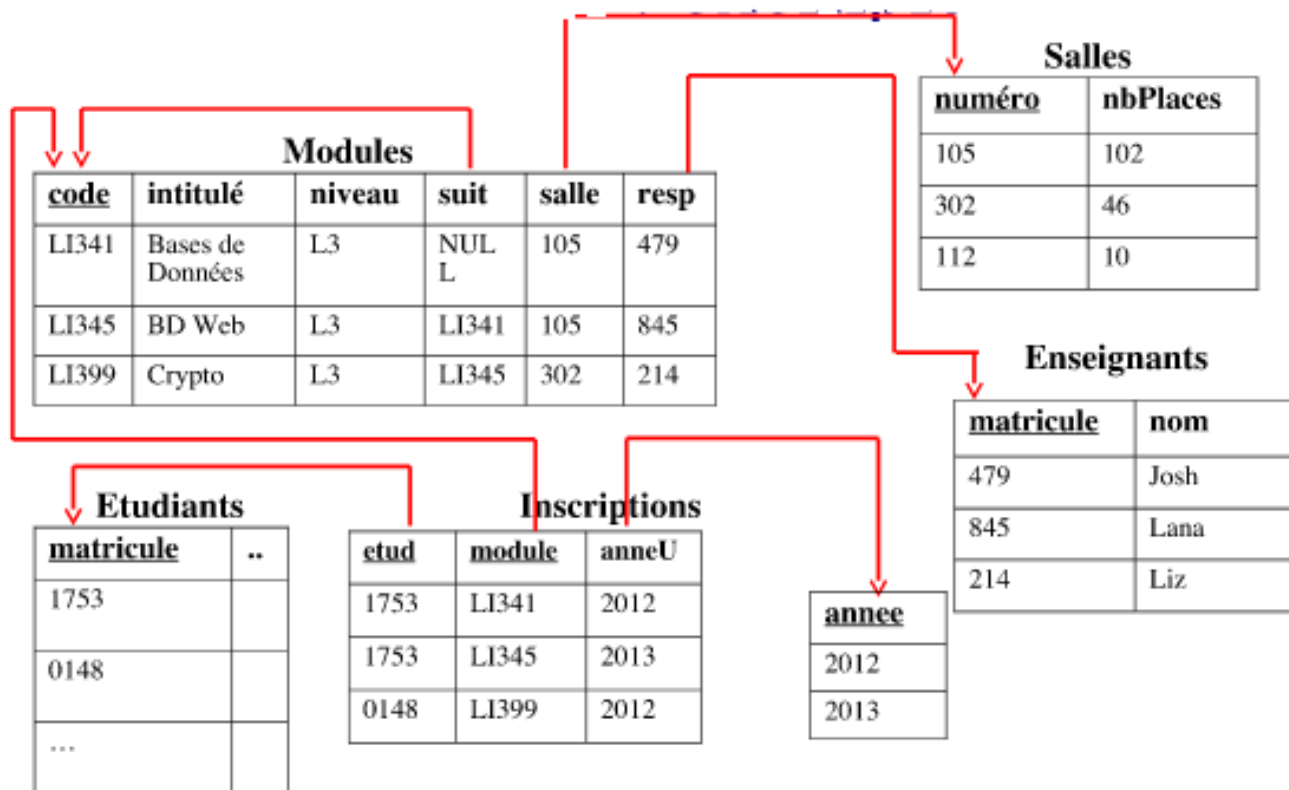
Clé étrangère:

- *sous-ensemble d'attributs* dont les valeurs proviennent des clés candidates de la même table ou d'autre table
- mécanisme de référencement des n-uplets



- Le nom d'une clé étrangère n'est pas nécessairement le même que celui de la clé référencée

Clé étrangère



Des clés étrangères peuvent composer la clé candidate/primaire d'une relation:

- Exemple: la clé {**etud**, **module**} de la table **Inscriptions**

La théorie de la normalisation

Permet de définir formellement la qualité des tables au regard du problème posé par la redondance des données. La théorie de la normalisation s'appuie sur la dépendance fonctionnelle.

Principe: décomposer une relation en plusieurs, en fonction des dépendances fonctionnelles, afin d'éliminer les anomalies (redondances).

DFs & Normalisation: les Dfs guident la normalisation. Une décomposition sans perte des données est une décomposition qui préserve les DFs.

La théorie de la normalisation

Les formes normales caractérisant les tables relationnelles :

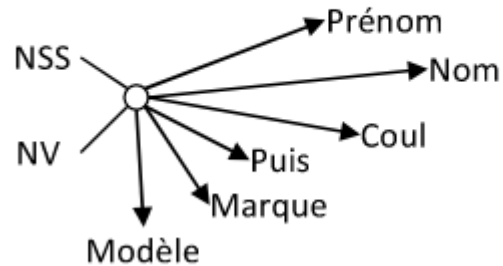
* **Première forme normale** : si elle ne contient que des attributs atomiques.

1FN : DFs

- tout attribut dépend fonctionnellement de la clé
- la relation ne contient que d'attributs atomiques

1FN : tous dépendent de la clé

R (NSS, NV, Prénom, Nom, Coul, Puis, Modèle, Marque)



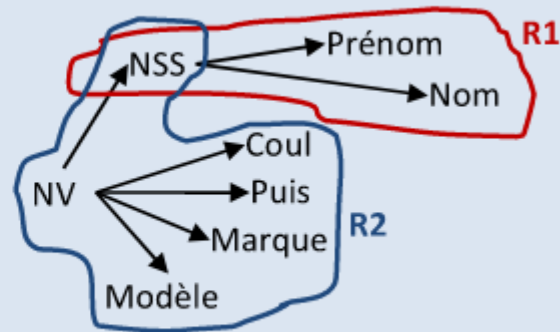
La théorie de la normalisation

* **Deuxième forme normale** : si elle ne contient que des attributs atomiques et, si de plus, il n'existe pas de dépendance fonctionnelle entre une partie d'une clé et une colonne non clé de la table.

2 FN : seulement DFEs

- être en 1FN
- tout attribut dépend de **toute** la clé
- uniquement des **DFEs** entre les attributs non-clé et la clé

2FN : uniquement des DFEs



R1 (NSS, Prénom, Nom)

R2 (NV, NSS, Coul, Puis, Modèle, Marque)

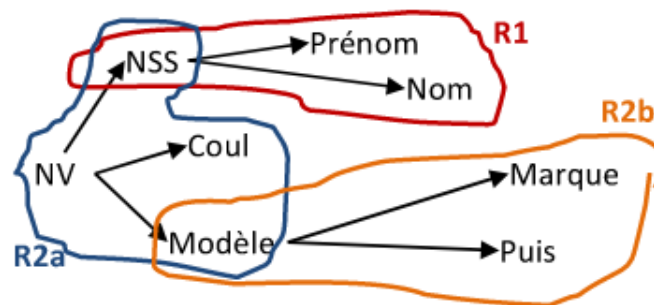
La théorie de la normalisation

* **Troisième forme normale** : si elle ne contient que des attributs atomiques, s'il n'existe pas de dépendance fonctionnelle entre une partie d'une clé et une colonne non clé de la table et si, de plus, aucune dépendance fonctionnelle entre les colonnes non clé.

3 FN : DFs Élémentaires et directes

- être en 2FN
- il n'existe aucune DF entre les attributs non-clé
- uniquement des DF élémentaires et directes entre les attributs clés et les attributs non-clé

3FN : uniquement des DF élémentaires et directes



R1 (NSS, Prénom, Nom)

R2a (NV, NSS, Coul, Modèle)

R2b (Modèle, Puis, Marque)

Ainsi, plus une table est normalisée moins elle comporte de redondances et donc de risques d'incohérence sémantiques dans les schémas relationnels.

Schéma Relationnel

Schéma d'une relation:

- Nom de la relation + Liste de ses attributs avec leur domaines (nombre, chaîne de caractères, date...) + Clés des tables et contraintes d'intégrité

Instance d'une relation:

- Ensemble des n-uplets de la table

Schéma d'une base de données:

- Ensemble des schémas des relations qui la composent

Clé primaire

1753	Smith	Joe	11 CP NYC
2410	Hersh	Dan	22 Rd NJ
0148	Clay	Maissa	7 HW NJ

<u>matricule</u>	nom	prénom	adresse
numérique	caractères	caractères	caractères

Schéma Relationnel

- **Schéma d'une BD** = ensemble des schémas de relation
 $S = \{R1, R2, \dots, Rn\}$ où **Ri** est un schéma de relation
- **Schéma de relation** = ensemble des attributs avec leurs domaines respectifs et les contraintes
 $R(\underline{A1:D1}, A2:D2, \dots, Am:Dm)$, A1 est clé primaire: relation d'arité m

- **Exemple:**

Etudiants(matricule : Number, nom : Varchar, prenom : Varchar, adresse : Varchar)

Modules(code : Number, intitule : Varchar, niveau : Varchar, salle : Number)

Salles(numero : Number, capacite : Number)

Schéma de la Base de Données: {Etudiants, Modules, Salle}

★ Number=numérique, Varchar=chaîne de caractères de longueur variable

- **Convention de notation:**

- Clé primaire : soulignement
- Clés étrangères : astérisque et désignation de la table référencée
- On omet les domaines des attributs

- **Exemple:**

Etudiants(matricule, nom, prenom, adresse, collaborateur*)

→ collaborateur fait référence à (la clé primaire de) Etudiants

Modules(code, intitule, niveau, salle*)

→ salle fait référence à Salles

Salles(numero, capacite, précédente *, suivante*)

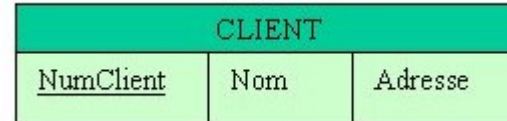
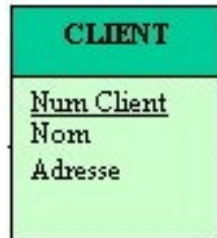
→ précédente et suivante font chacune référence à Salles

Du Modèle E/A au modèle Relationnel

Les règles principales de transformation d'un schéma conceptuel Entité-Relation en un schéma relationnel sont :

Règle I : Toute entité est traduite en une table relationnelle dont les caractéristiques sont les suivantes :

- * le nom de la table est le nom de l'entité ;
- * la clé de la table est l'identifiant de l'entité ;
- * les autres attributs de la table forment les autres colonnes de la table.



Du Modèle E/A au modèle Relationnel

Règle II: Toute relation binaire plusieurs à plusieurs est traduite en une table relationnelle dont les caractéristiques sont les suivantes :

- * le nom de la table est le nom de la relation ;
- * la clé de la table est formée par la concaténation des identifiants des entités participant à la relation ;
- * les attributs spécifiques de la relation forment les autres colonnes de la table.

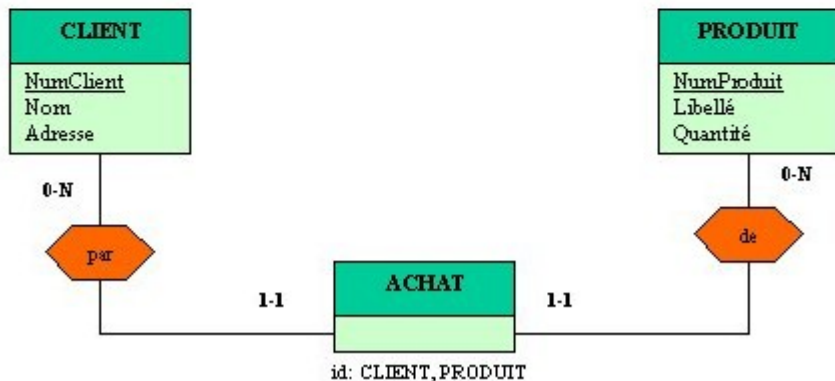
Une contrainte d'intégrité référentielle est générée entre chaque colonne clé de la nouvelle table et la table d'origine de cette clé.

Du Modèle E/A au modèle Relationnel

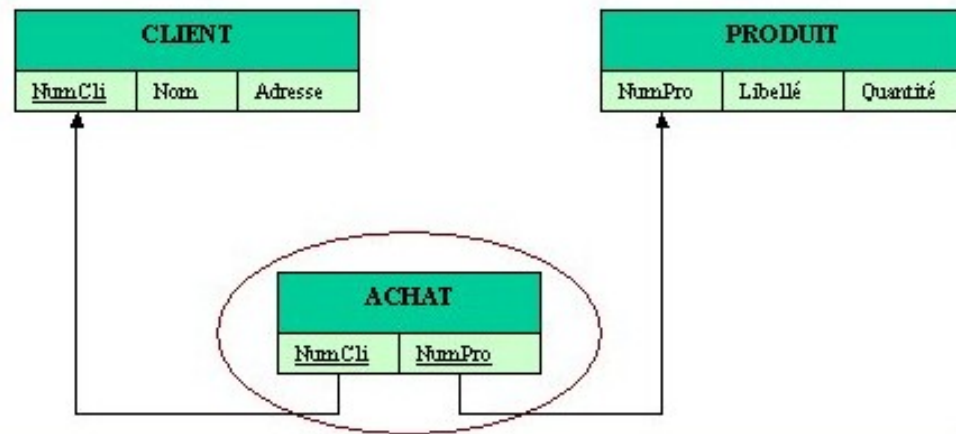
Schéma conceptuel :



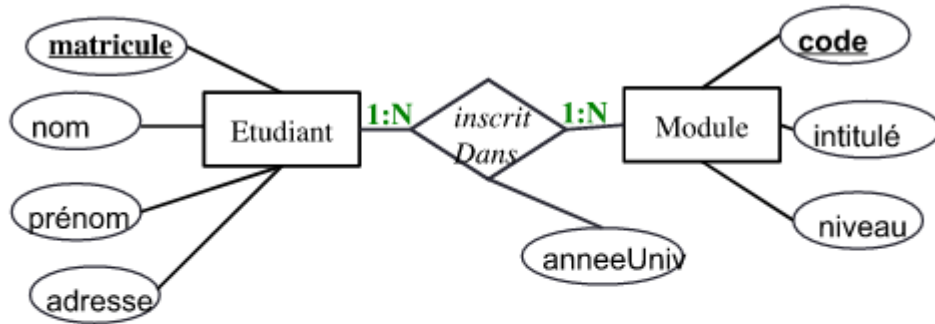
Transformation du schéma conceptuel :



Traduction:



Exemple: Association n-aire(x,n)



Etudiant(matricule, nom, prenom, adresse)

Module(code, intitulé, niveau)

Inscriptions(matricule*, code*, anneeUniv)

Du Modèle E/A au modèle Relationnel

Règle III: Toute relation binaire un à plusieurs est traduite :

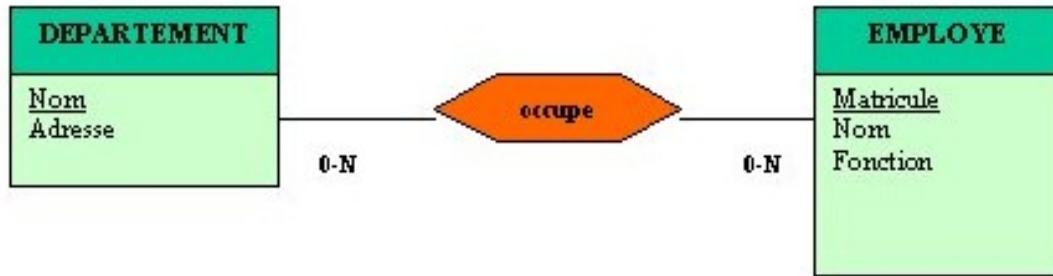
1- soit par un report de clé : l'identifiant de l'entité participant à la relation côté N est ajoutée comme colonne supplémentaire à la table représentant l'autre entité. Cette colonne est parfois appelée clé étrangère. Le cas échéant, les attributs spécifiques à la relation sont eux aussi ajoutés à la même table ;

2- soit par une table spécifique dont les caractéristiques sont les suivantes :

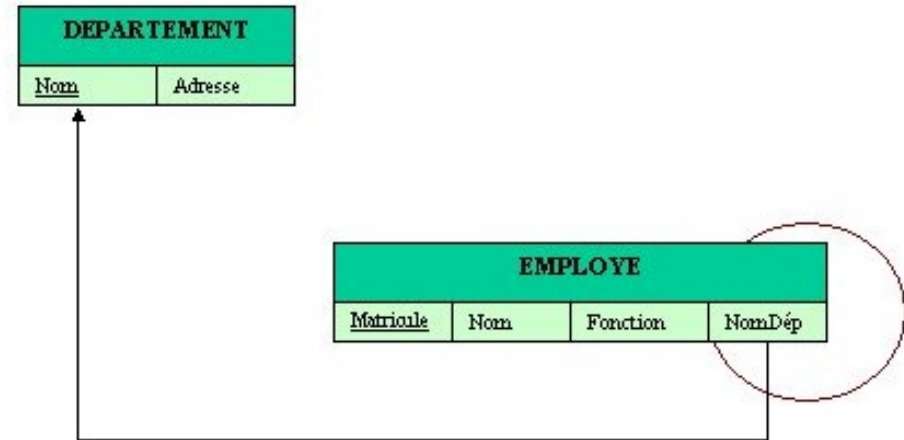
- * le nom de la table est le nom de la relation ;
- * la clé de la table est l'identifiant de l'entité participant à la relation côté 1 ;
- * les attributs spécifiques de la relation forment les autres colonnes de la table.

Du Modèle E/A au modèle Relationnel

Schéma conceptuel :



Traduction :



Du Modèle E/A au modèle Relationnel

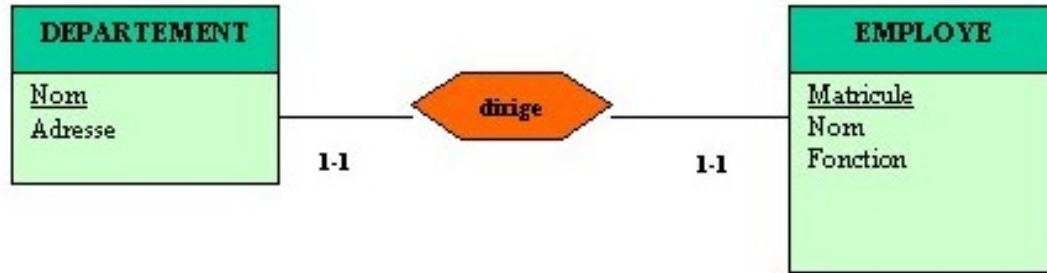
Règle IV: Toute relation binaire un à un est traduite, au choix, par l'une des trois solutions suivantes :

- 1- fusion des tables des entités qu'elle relie (choix1) ;
- 2- report de clé d'une table dans l'autre (choix2) ;
- 3- création d'une table spécifique reliant les clés des deux entités (choix3).

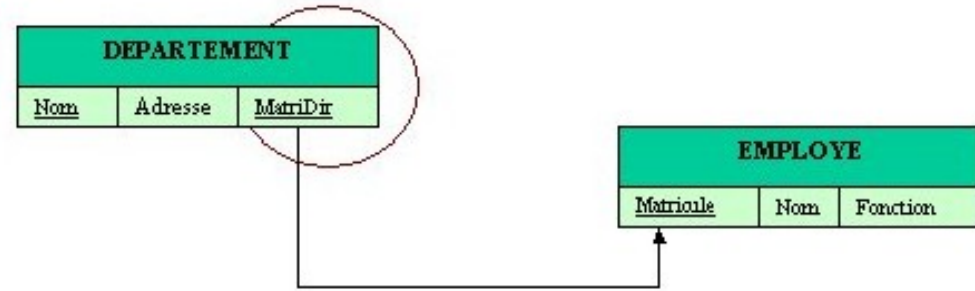
Les attributs spécifiques de cette relation sont ajoutés à la table résultant de la fusion (choix1), reportés avec la clé (choix2), ou insérés dans la table spécifique (choix3).

Du Modèle E/A au modèle Relationnel

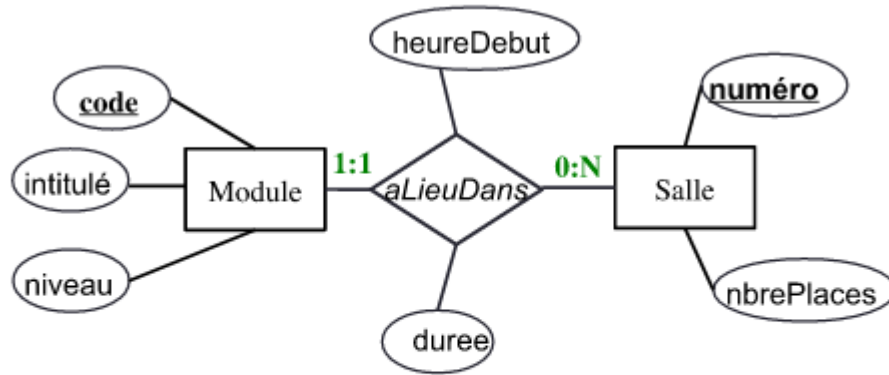
Schéma conceptuel :



Traduction (choix 2) :



Exemple: Association n-aire (x,y) et (1,1)



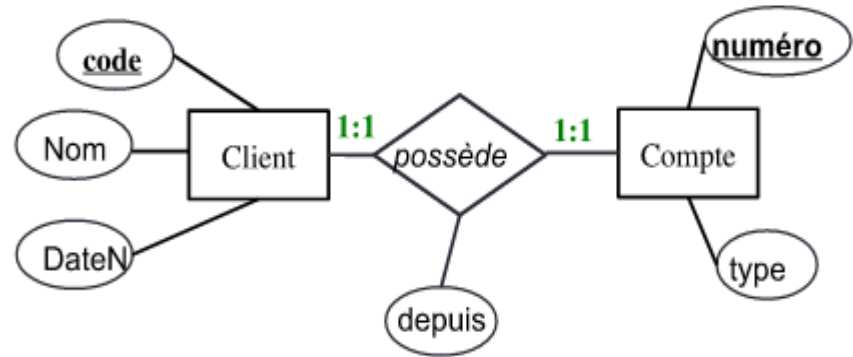
Rappel : attributs de l'association {code, numéro, heureDebut, durée}

Module(code, intitulé, niveau, numéroSalle*, heureDebut, duree)

NuméroSalle référence numéro de la table Salle

Salle(numéro, nbrePlaces)

- Si plusieurs cardinalités 1:1, modifier la table de seulement une des relations correspondantes



Compte(numéro, type)

Client(code, Nom, DateN, NumeroCompte*, depuis)

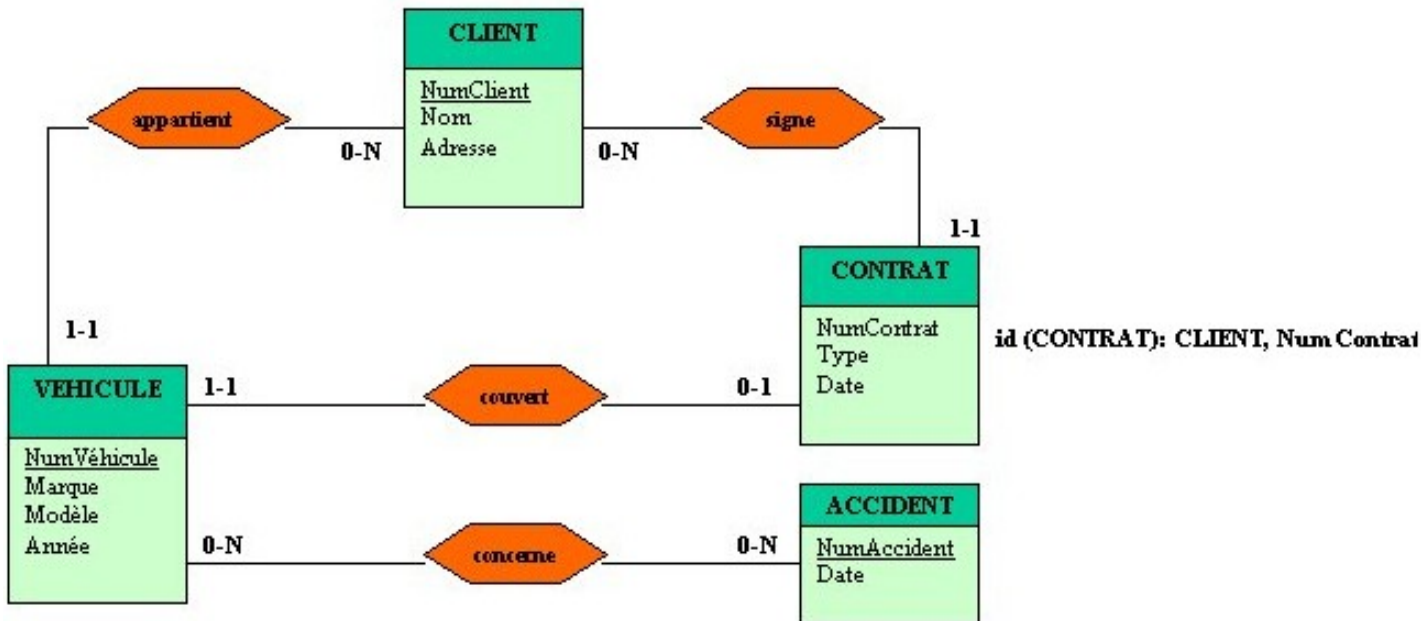
NumeroCompte référence numéro de la table Compte

- La table *Compte* peut être enlevée si l'entité *Compte* n'est associée à aucune autre entité, tous ses attributs seront stockés dans la relation *Client*

Application

Schéma conceptuel :

Convertir ce Modèle Conceptuel en Modèle Relationnel

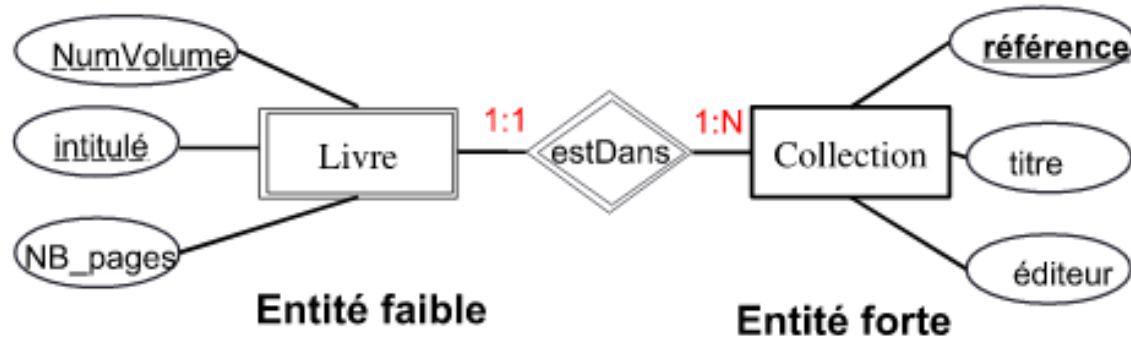


Règles de transformation des entités faibles

- Créer une relation pour l'entité faible
- **Clé de la relation** correspondante à une entité faible = concaténation de l'**identifiant de l'entité faible** et celui de l'entité dont elle dépend
 - les attributs constituant l'identifiant de l'entité forte constituent une clé étrangère (qui fait aussi partie de la clé primaire)

Transformation similaire à celle pour le Cas 1, en incluant en plus la clé étrangère dans la clé primaire

Règles de transformation des entités faibles : Exemple



Collection (référence, titre, éditeur)

Livre (référence*, numVolume, intitulé, NB_pages)

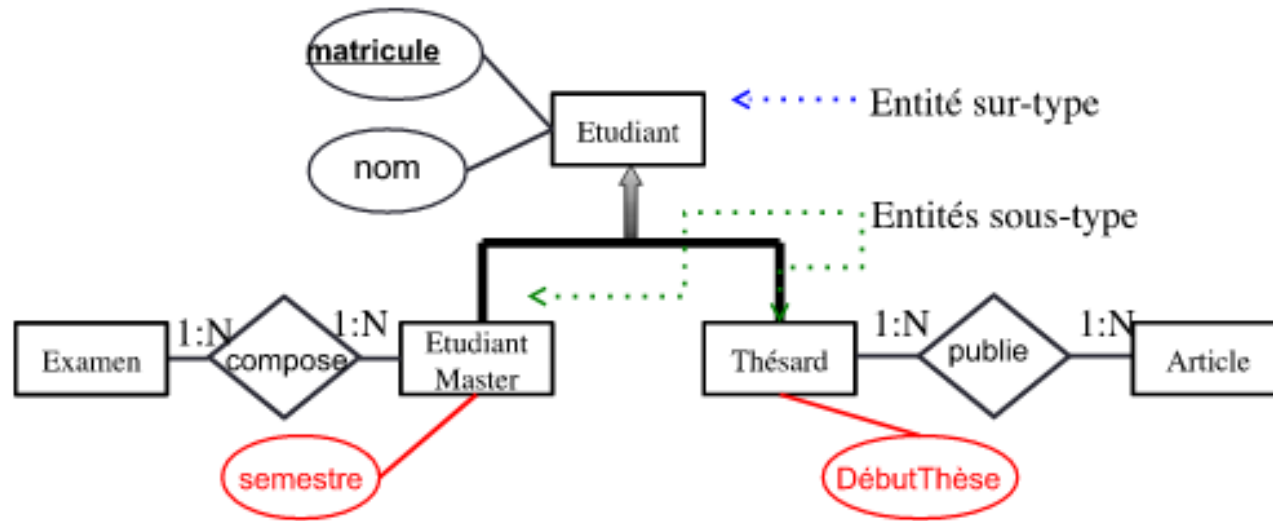
référence est une clé étrangère qui fait référence à la clé primaire de Collection

Clé primaire Livre: {référence, numVolume, intitulé}

Règle de transformation de la spécialisation

- Transformation de la spécialisation:
 - Créer une relation (table) pour chaque entité sous-type, les attributs de la relation sont ceux de l'entité sous-type
 - *Clé de la relation*: l'identifiant de l'entité sur-type
- **Cas particulier**, si l'entité sur-type est abstraite (il n'existe pas d'instance de cette entité dans l'application):
 - supprimer la table correspondante à l'entité sur-type
 - rajouter tous ses attributs dans toutes les tables correspondantes aux entités sous-type

Règle de transformation de la spécialisation: Exemple



Etudiant(matricule, nom)

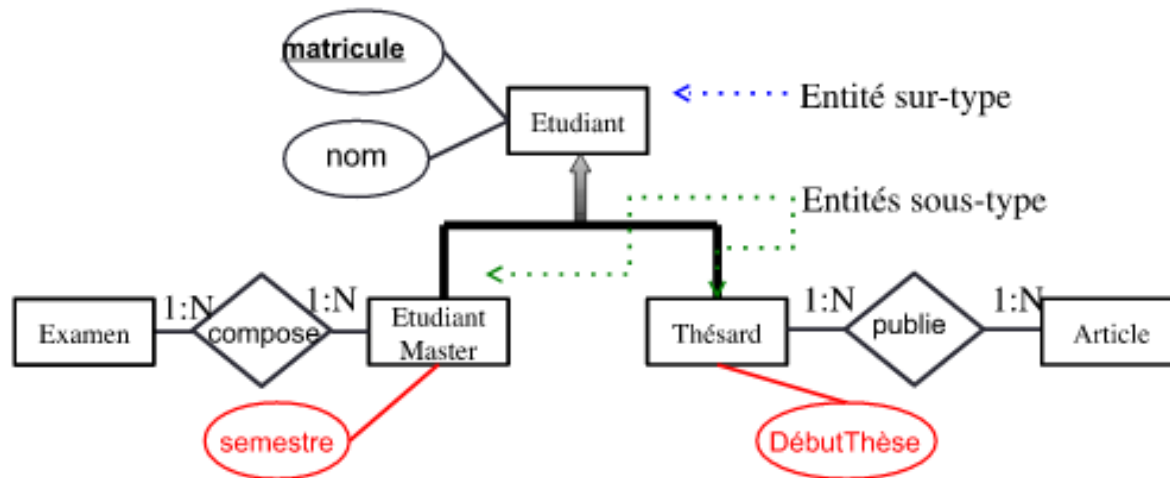
EtudiantMaster(matricule*, semestre)

Thésard(matricule*, debutThese)

Dans **EtudiantMaster** et **Thésard**
matricule est clé primaire et étrangère
(référence matricule de la table **Etudiant**)
en même temps!

Règle de transformation de la spécialisation: Exemple

- Il n'existe pas d'étudiants autres que les étudiants inscrits en master ou en thèse : on supprime la table **Etudiant**



Etudiant(matricule, nom)

EtudiantMaster(matricule, nom, semestre)

Thésard(matricule, nom, debutThese)

Dans **EtudiantMaster** et **Thésard**
matricule est la clé primaire, elle n'est pas clé étrangère.