

Conception et normalisation

1. Modélisation des données

Un modèle de données est un ensemble de concepts utilisés pour décrire la structure d'une base de données. Par structure de base de données, nous entendons, les types de données, les relations, et les contraintes qui définissent le gabarit de la base de données. Tout modèle peut être exprimé à divers niveaux de précision, conceptuel, logique physique et externe. Cette définition sera retenue lorsque nous présenterons la partie « Présentation des approches traditionnelles pour l'intégration des données »

Les techniques de modélisation de données permettent d'une part de comprendre le fonctionnement d'un système et la façon dont les données sont organisées et d'autres part de construire un système d'information qui reflète la réalité.

Il existe plusieurs niveaux de représentation (ou de modélisation des données).

Niveau conceptuel : le niveau conceptuel ou le schéma conceptuel de la base de données décrit la réalité du système d'information d'une organisation indépendamment du SGBD d'implantation. Le modèle utilisé à ce niveau peut-être le modèle d'Entité/Association (modèle d'Entité/Relation). Ce niveau est de la responsabilité de l'analyste concepteur ou de l'administrateur de la base de données

Niveau interne (logique et physique) : le niveau interne spécifie comment les données sont représentées ou stockées sur les supports de stockage. Ce niveau est complètement pris en charge par les SGBD. (SGBDR Oracle)

Niveau externe : ce niveau correspond à la vision de tout ou une partie du schéma conceptuel par un groupe d'utilisateurs concernés par une application ou une partie de l'application. Il s'agit de décrire à l'aide de schémas externes (vues) la façon dont seront perçues les données par les programmes.

2. Le modèle Conceptuel : Formalisme d'oracle.

Dans le formalisme d'oracle on met en évidence les entités avec leurs propriétés et les identifiant en respectant le formalisme suivant :

Caractéristiques

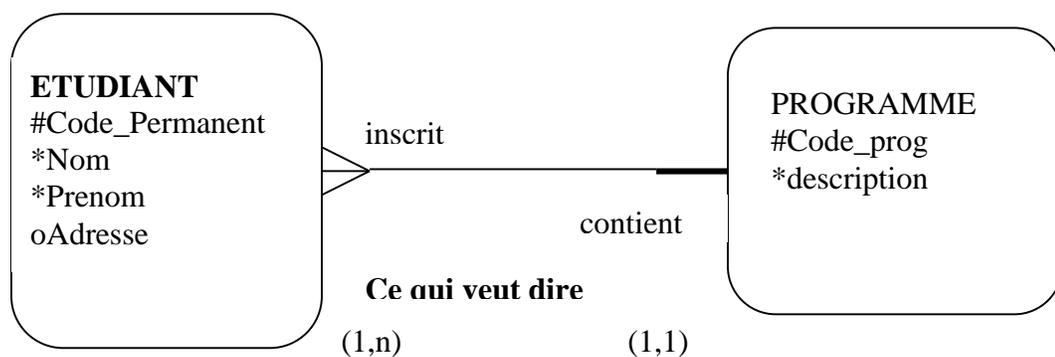
- ✓ **Les relations ne sont pas porteuses de propriétés** (contrairement au modèle Merise)
- ✓ Les relations ont des noms indiquant leurs rôles dans le système.
- ✓ Dans une entité l'identifiant est précédé du symbole #
- ✓ Dans une entité les attributs obligatoires sont précédé du symbole *
- ✓ Dans une entité, les attributs optionnels sont précédés du symbole o
- ✓ Les cardinalités sont représentées par des lignes avec les terminaisons possibles suivantes :

Représentation des cardinalités

	Optionnelle	signification	Obligatoire	signification
Unique	-----	(0,1)	=====	(1,1)
Multiple	----- <	0,n)	===== <	(1,n)

Pour une entité dans le modèle, la ligne près de l'entité indique l'**optionalité** et la ligne au bout indique la **multiplicité**.

Exemple 1 et explication



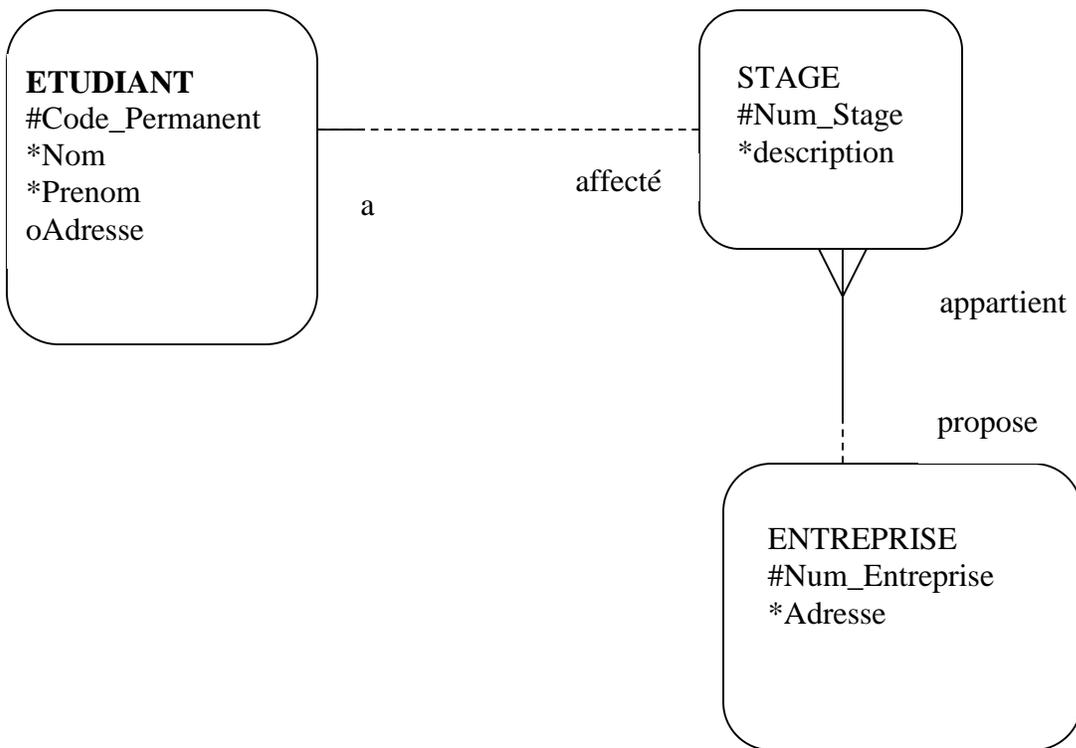
Pour la relation *inscrit*

La ligne étant continue du côté de l'entité Etudiant, donc il s'agit d'une obligation. Comme il n'y a pas de patte d'oie du côté de l'entité Programme, alors il s'agit de l'unicité. On peut lire alors la relation : Un étudiant est inscrit (le nom de la relation) à un seul programme.

Pour la relation *contient*

Du côté de l'entité Programme, la ligne est continue, il s'agit d'une obligation. Du côté de l'entité Etudiant, la terminaison de la relation est une patte d'oie, donc il s'agit de la multiplicité. On lira la relation de la façon suivante. Un programme contient 1 à plusieurs étudiants.

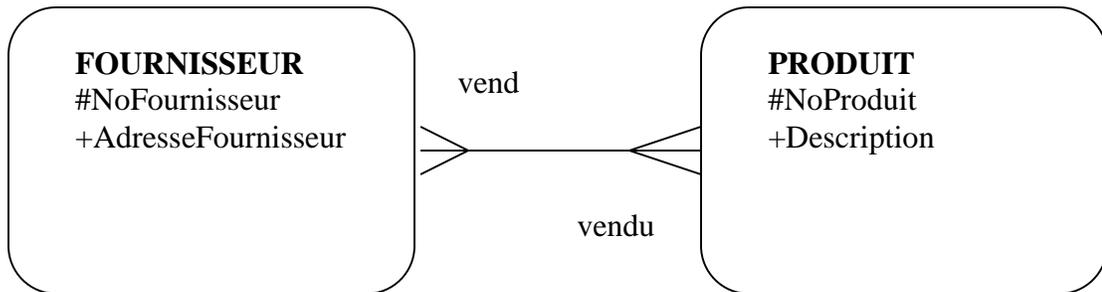
Exemple 2



Pour la relation *a*, on peut la lire de la façon suivante : un étudiant a 0 ou 1 stage. La relation *affectée* se lie de la même manière. En effet un stage est affecté à 0 ou 1 étudiant.

Une entreprise propose 0 ou plusieurs stages, c'est ainsi que nous pouvons lire la relation *propose*. Par contre, un stage appartient 1 et une seule entreprise.

Exemple 3



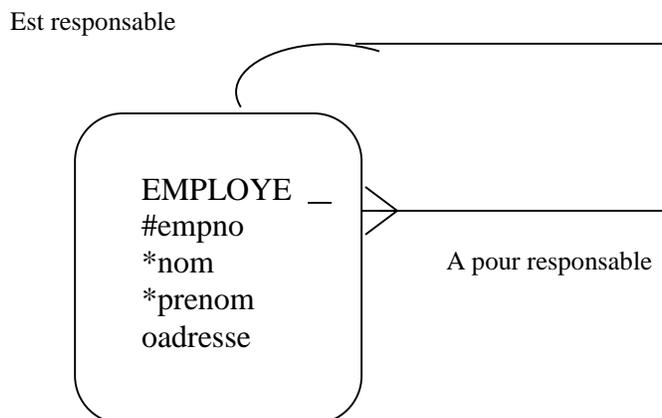
Attention : dans le formalisme d'oracle, les cardinalités signifient la même chose que dans le modèle MERISE. MAIS elles sont placées de l'autre côté de l'entité.

Remarque : Toutes les relations décrites plus haut sont des relations binaires simples.

Sauf cas contraire (si le système l'exige), il est fortement recommandé de modéliser notre base de données en utilisant des relations binaires

Exemple d'une relation réflexive :

Il existe des systèmes qui exigent qu'une relation soit reliée à la même entité. Par exemple dans une entreprise si nous voulons représenter les employés avec leur responsable immédiat il est clair que nous avons besoin d'une relation sur l'entité **EMPLOYE** avec l'entité **EMPLOYE**. La représentation sera la suivante



Autres concepts

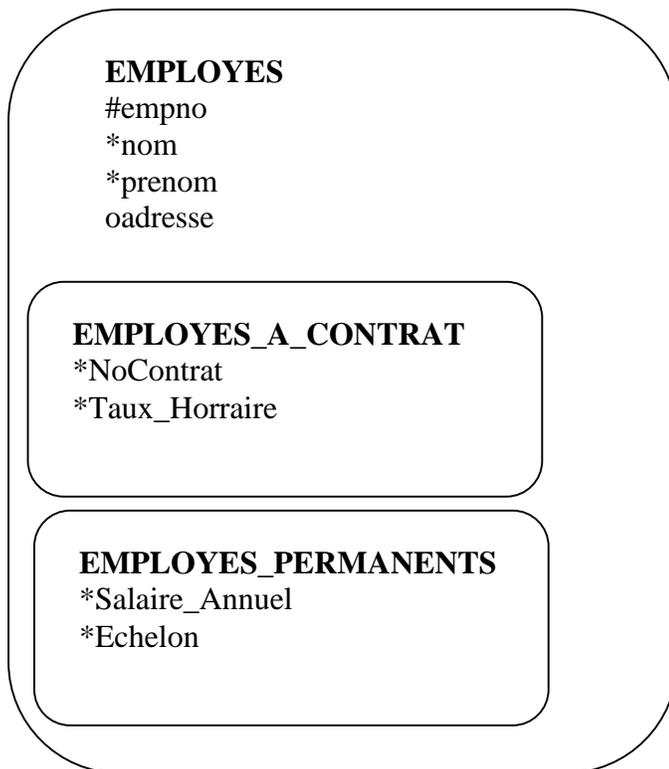
Généralisation d'entité.

La généralisation d'entité consiste à regrouper sous la même entité plusieurs entités ayant les mêmes caractéristiques.(un peu comme le concept d'héritage en POO.).

On appelle entité *surtype*, l'entité qui englobe plusieurs entités.

On appelle entité *sous-type* une entité à l'intérieur de l'entité surtype.

Exemple



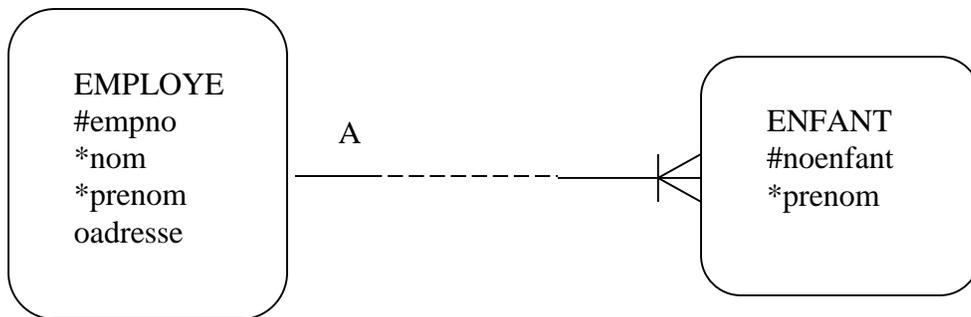
Dans le modèle précédent, l'entité EMPLOYÉS est l'entité *surtype* . Les entités EMPLOYÉS_PERMANENTS et EMPLOYE_A_CONTRAT sont des entités sous-type. Chacune d'elles possèdent ses propres propriétés et les propriétés de l'entité surtype EMPLOYÉS.

Ce type de modèle fait penser au concept d'héritage dans la POO.

Identifiant Relatif

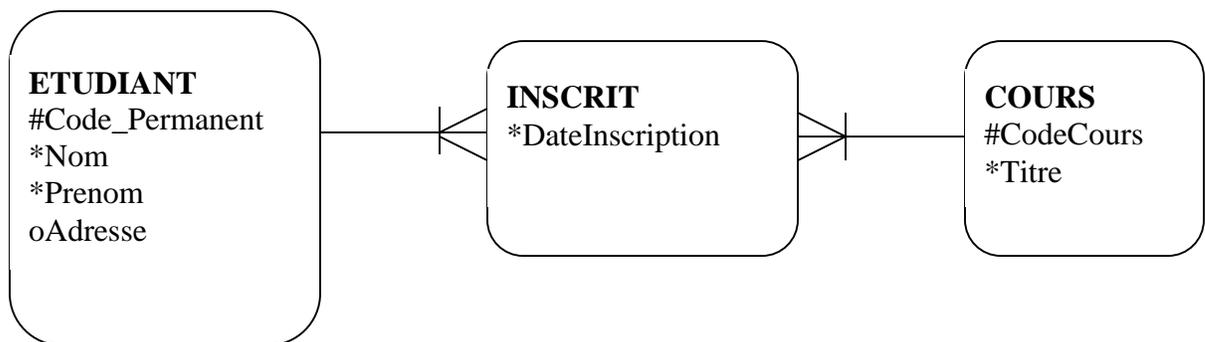
Il existe des entités dont les occurrences ne peuvent être identifiées par l'identifiant de l'entité en question, on dit alors que l'entité a un **identifiant faible**. Pour identifier parfaitement les occurrences, nous devons utiliser l'identifiant d'une autre entité.

Exemple 1



L'identifiant de l'entité **ENFANT** est faible. L'entité **ENFANT** est entièrement dépendante de l'entité **EMPLOYE**, donc pour identifier l'enfant, nous avons besoin de connaître l'identifiant de l'employé à qui appartient l'enfant.

Exemple 2



Vérification et normalisation du modèle entité/relation

La vérification

On dit qu'un modèle conceptuel (entité/relation) est vérifié s'il répond aux conditions suivantes :

- Tous les attributs non calculés sont présents dans une entité ou une relation dans le cas du modèle de MERISE)
- Aucun attribut n'est redondant
- Toutes les entités ont un identifiant
- Tous les attributs sont élémentaires (non décomposable ou se traitent comme un tout, exemple l'attribut adresse peut-être décomposable)

La normalisation du modèle Conceptuel

Les règles de normalisation du modèle conceptuel sont les mêmes que celle du modèle relationnel et permettent d'assurer :

- La non redondance des données
- L'intégrité des données
- La facilité de mise à jour

Lors de la normalisation d'un modèle de données (peu importe le niveau de modélisation, de conceptualisation ou de représentation) nous devons avoir au moins la *3eme forme normale*.

Important : Lorsque le modèle conceptuel est normalisé, le mod