

Rappel sur les bases de données

1) Généralités

1.1 Base de données et système de gestion de base de données: définitions

Une base de données est un ensemble de données stockées de manière structurée permettant ainsi d'être utilisées par un ou plusieurs programmes informatiques. Les données doivent être consultables selon n'importe quel critère (Par exemple, trouver les livres de sciences fiction écrit par des auteurs de moins de 25 ans entre 1925 et 1930). De plus la base de données doit permettre la mise à jour et la consultation des données par plusieurs utilisateurs.

Pour cela, il ya les systèmes de gestion de bases de données. Un système de gestion de base de données (SGBD) est un ensemble de programmes permettant à plusieurs utilisateurs d'accéder à une base de données en leur donnant l'illusion qu'ils sont les seuls à travailler sur ces données. Un SGBD doit permettre entre autres la recherche, la mise à jour, l'insertion et la suppression de données.

1.2 Niveaux de données

On peut distinguer trois niveaux à un SGBD :

-Niveau physique (le système de gestion de fichier) qui correspond à l'organisation et le stockage physique des données

-Niveau conceptuel (SGBD interne) qui gère les données stockées dans les fichiers, les liens existants entre ces données. C'est un niveau conceptuel qui permet de décrire les données, leurs propriétés et comment y accéder.

-Niveau vue (SGBD externe) qui définit la mise en forme et la présentation des données aux programmes et aux utilisateurs ayant accès à la base de données.

Cette décomposition en couche permet d'avoir une indépendance entre les données et les traitements.

1.3 Propriétés

De manière générale, le SGBD doit avoir les caractéristiques suivantes :

-Indépendance physique : si la manière dont sont stockées les bases de données est modifiée (c'est à dire le gestionnaire de fichiers) il n'est pas nécessaire de modifier le niveau interne et externe.

-Indépendance logique : le SGBD interne peut être modifié sans remettre en cause le niveau physique.

-Manipulabilité : des personnes ne connaissant pas la base de données et la manière dont elle est stockée en machine doivent pouvoir manipuler les données.

- Rapidité des accès : le SGBD doit pouvoir répondre aux requêtes le plus rapidement possible.

- Limiter les redondances : le SGBD doit éviter au maximum de stocker des informations redondantes afin d'éviter des pertes en espace mémoire et aussi pour éviter des erreurs (comme par exemple la mise à jour d'une donnée mais pas de ses copies éventuelles).

- Vérification de l'intégrité : les données doivent être cohérentes entre elles. Par exemple, si une donnée fait référence à une autre alors cette dernière doit être aussi présente dans la base de données. C'est le cas pour la bibliothèque : un emprunt fait toujours référence à un livre et ce livre doit exister dans la base de données pour que celle-ci reste cohérente.

- Partageabilité des données : le SGBD doit permettre l'accès simultané aux données par plusieurs utilisateurs.

- Sécurité des données : le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données mais aussi d'assurer un bon fonctionnement après une panne. Pour cela des mécanismes de sauvegardes doivent être mis en place.

1.4 Modèle de données

Pour définir le niveau conceptuel et les vues du SGBD, nous avons besoin d'un modèle de données qui va décrire de manière abstraite les données. Il existe différents modèles de SGBD. Cependant il est possible de distinguer deux types de modèles : les modèles conceptuels et les modèles logiques. Les modèles conceptuels permettent d'analyser de manière graphique les besoins de l'application. Les modèles logiques permettent eux de modéliser l'application sous un langage facilement implémentable. Nous présentons dans les sections suivantes un modèle conceptuel : le schéma entité-association et le modèle logique relationnel où les données sont enregistrées dans des tableaux à deux dimensions.

2) Schéma entité-association (appelé parfois entité-relation)

2.1 Définitions

Entité : « objet » du monde réel que l'on peut identifier et qui possède un ensemble de propriétés évaluées qui la décrivent.

Par exemple : le cours de mathématiques de L3 est une entité décrit par les propriétés matière = mathématiques, niveau = L3

Classe d'entités : c'est un regroupement d'entités de même nature. Elle est identifiée par un nom et par une liste de propriétés.

Par exemple : le cours de mathématiques de L3 et le cours de physique de L1 sont deux entités appartenant à la classe d'entités « cours » ayant pour propriétés « matière » et « niveau ».

Association : lien logique entre plusieurs entités. Elle peut avoir des propriétés évaluées.

Par exemple : le professeur Albert Dupont donne le cours de mathématique du jeudi après midi. L'entité « professeur Albert Dupont » est donc associée à l'entité « cours de mathématique du jeudi matin »

Classe d'associations : c'est un lien entre deux ou plusieurs classes d'entités. Elle est identifiée par nom (qui correspond souvent à un verbe) et possède une liste de propriétés.

Par exemple : « donne » est une classe d'associations qui va associer la classe d'entités « professeur » et la classe d'entités « cours »

Attribut : propriété d'une association ou d'une entité. Elle est caractérisée par un nom et un type élémentaire (varchar, int ...).

Par exemple : « matière » est l'attribut de l'entité « cours », elle a pour type un « varchar »

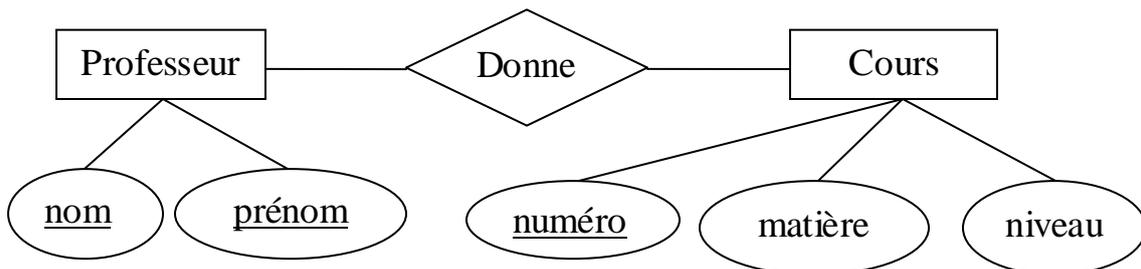
Identificateur : un identificateur est un attribut ou un ensemble d'attributs permettant de distinguer deux entités appartenant à la même classe d'entités.

Par exemple : « nom » et « prénom » sont les identificateurs de la classe d'entités « professeur ». De même on pourrait imaginer que cours a un numéro lui permettant de l'identifier

Par abus de langage, on appelle entité une classe d'entités. De la même façon, on appelle association une classe d'associations.

2.2 Représentation graphique

Une classe d'entités est représentée par un rectangle avec son nom à l'intérieur. Une classe d'associations est représentée par un losange avec son nom à l'intérieur et est relié par une arête à chaque classe d'entités qu'elle relie. Les attributs sont eux représentés par des cercles contenant leurs noms et sont reliés à la classe d'entité ou la classe d'association qu'ils décrivent. Le ou les attributs identificateurs sont soulignés.



2.3 Les cardinalités

Les cardinalités d'une classe d'associations indiquent pour chaque entité le nombre de fois minimum et maximum où elle participe à l'association.

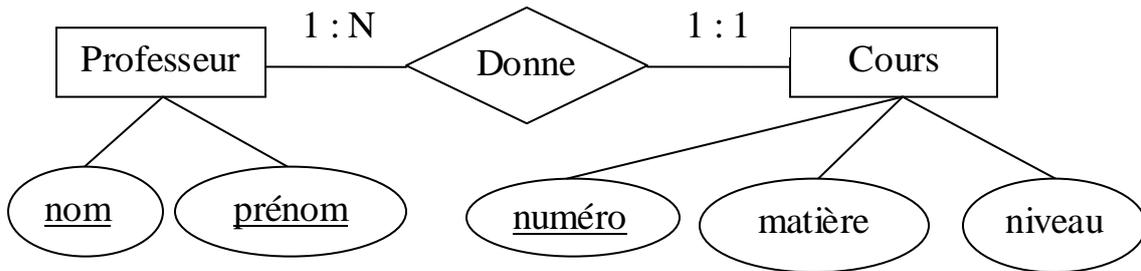
- La cardinalité **minimum** traduit donc le nombre de fois minimum où une entité participe à la classe d'association. Sa valeur peut être de 0 ou 1.

Par exemple : l'entité « le professeur Albert Dupont » peut participer au minimum 1 fois à la classe d'association « donne ».

- La cardinalité **maximum** traduit donc le nombre de fois maximum où une entité participe à la classe d'association. Sa valeur peut être de 1 ou N.

Par exemple : l'entité « le professeur Albert Dupont » peut participer au maximum N fois à la classe d'association « donne », car ce professeur peut enseigner plusieurs cours.

Le couple est représenté graphiquement sur l'arête reliant la classe d'entité à la classe d'association par min : max



3) Modèle relationnel

3.1 Définition

Domaine : Ensemble de valeurs caractérisés par un nom. Il correspond au type.

Par exemple : l'ensemble des entiers ou

Produit cartésien : Le produit cartésien d'un ensemble de domaines : est l'ensemble des n-upplets tel que . On le note

Par exemple : Le produit cartésien de et de est

Relation : Sous-ensemble du produit cartésien de plusieurs domaines. Une relation est toujours caractérisée par un nom.

Par exemple, on peut définir avec les domaines et

{Dupont, Durand, Dumas} la relation professeur = {(Albert, Dupont), (Paul, Dumas), (Paul, Durand)}

Une relation peut être représentée sous forme tabulaire. On appelle alors **attribut** la colonne d'une relation caractérisée par un nom.

Par exemple, la relation professeur possède deux attributs : prénom et nom.

Prénom	Nom
Albert	Dupont
Paul	Dumas
Paul	Durand

Un schéma de relation est défini par le nom de la relation suivi de la liste des attributs et de la définition de leur domaine.

Par exemple : professeur (prénom : {Pierre, Paul}, nom : {Dupont, Durand, Dumas})

Note : On peut omettre la définition des domaines si ceux-ci sont implicites (du fait du nom de l'attribut)

Un schéma de BD relationnelle est un ensemble de schémas de relation

Une base de données relationnelle est composée d'un ensemble de **tables** à deux dimensions. Chaque table correspond à une **relation** où les **colonnes** sont les **attributs** de la relation. Chaque ligne contient un n-upplet correspondant à un enregistrement de la table.

Attention : Une relation est un ensemble par définition il ne peut donc y avoir deux n-upplets (ou deux enregistrements) identiques.

3.2 Clé

Surclé d'une relation : ensemble d'attributs dont la connaissance des valeurs permet d'identifier un n-upplet unique de la relation considérée. Il peut exister plusieurs surclé. Une relation étant un ensemble : l'ensemble des attributs de la relation forme toujours une surclé.

Clé d'une relation : est une surclé minimale. Par conséquent si on retire un attribut de la clé alors le reste des attributs ne forme plus une surclé.

Lorsqu'il existe plusieurs clés, on en choisit une que l'on nomme **clé primaire**, elle est représentée dans le schéma de relation en soulignant le nom des attributs concernés par la clé.

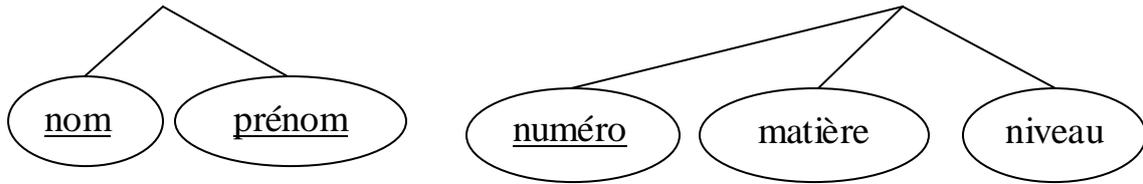
Contraintes référentielles : Une contrainte d'intégrité référentielle définit qu'un attribut (ou groupe d'attributs) ne peut prendre comme valeur qu'une valeur existant comme clé (primaire ou non) d'une autre relation. En d'autres termes, elle interdit à un n-upplet de référencer un n-upplet inexistant. On appelle ces attributs **clé étrangère**. Ils sont symbolisés par le signe #.

3.3 Passage d'un schéma E-A à un schéma relationnel

- Pour chaque classe d'entité du schéma E-A créez une relation ayant le même nom et dont les attributs correspondent aux propriétés de la classe d'entités. La clé de ce schéma est l'identificateur de la classe d'entité.

Par exemple : Pour le schéma E-A ci-dessous, on peut créer deux relations : professeur(nom, prénom) et cours (matière, niveau, numéro)





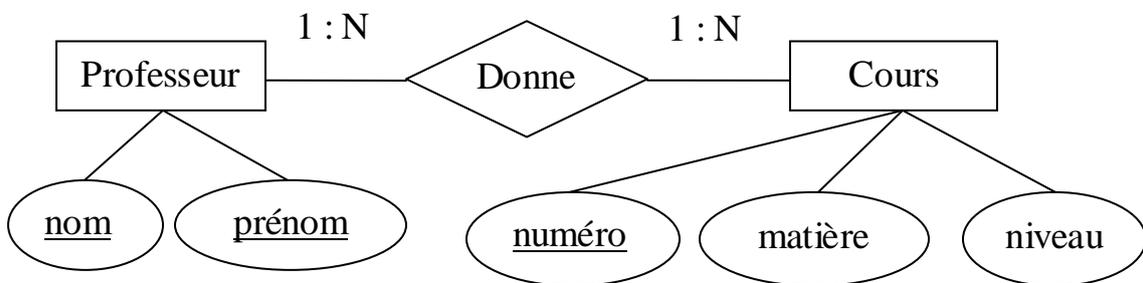
-Pour chaque classe d'associations dont au moins une classe d'entité à une cardinalité 1-1, rajoutez dans le schéma de cette classe d'entités la clé de l'autre classe d'entité.

Par exemple, professeur(nom, prénom) et cours (matière, niveau, numéro, #nom, #prénom) où les attributs nom et prénom réfèrent au professeur qui donne le cours. nom et prénom doivent absolument référencer un professeur existant dans la base, ils formant une clé étrangère.

-Pour les autres classes d'associations, créez un schéma de relation ayant le même nom dont les attributs seront formés des propriétés de la classe d'association et des identificateurs des classes d'entités participant à la relation. La clé d'une telle relation est constituée de tous les attributs correspondant aux identificateurs des classes d'entités participant à la relation.

Par exemple : si l'on modifie le schéma en permettant qu'un même cours soit assuré par plusieurs professeurs. Nous aurons le schéma relationnel suivant :

- professeur (nom, prénom)
- cours (matière, niveau, numéro)
- donne(#nom, #prénom , #numéro)



4) Quelques liens utiles

<http://www-inf.int-evry.fr/COURS/BD/>

<http://laurent-audibert.developez.com/Cours-BD/>

<http://www-poleia.lip6.fr/%7Egancarsk/PolyLicBD2003.pdf>

Il en existe plein d'autres, il suffit de chercher.