

Chapitre 6

Modélisation conceptuelle de BD

Plan:

1. Le modèle Entités-Associations (E/A) (modèle conceptuel)
2. Traduction en modèle relationnel (modèle logique)

Pourquoi une modélisation préalable ?

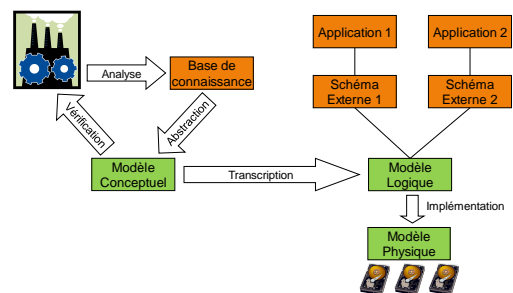
- Il est difficile de modéliser un domaine sous une forme directement utilisable par un SGBD.
 - Stockage physique
 - Cohérence/intégrité des données
 - Sécurité...
- Une ou plusieurs modélisations intermédiaires sont donc utiles (avant et après le modèle logique).

MERISE

- (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise)
- Langage de spécification très répandu en BD
- Concept clé: séparation des données et des traitements.
- Principe : hiérarchiser les préoccupations de modélisation qui sont de trois ordres : la conception, l'organisation et la technique.

=> 3 niveaux de représentation des données :
niveau **conceptuel**, niveau **logique**, niveau **physique**.

Conception de BD

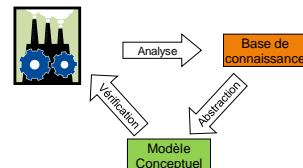


La base de connaissance



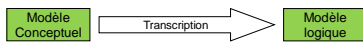
- Informations peu structurées
- Collectionnées sans outil formel
- Contient :
 - Les informations sur les données
 - Les informations sur les contraintes entre les données
 - Les règles de gestion et de transformation de données

Le modèle conceptuel des données



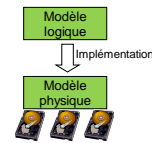
- Décrit les objets principaux, leurs caractéristiques et leurs relations grâce à une représentation schématisée des données.
=> Information structurée : modèle Entités-Associations.
- Représentation graphique
- Indépendant des choix « informatiques »
- Affiné par une méthode itérative

Le modèle logique des données



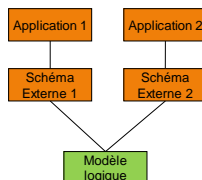
- Transcription du modèle conceptuel dans un formalisme adapté à une implémentation ultérieure, au niveau physique, sous forme de base de données.
- Propre à un type de base de données (relationnelle, réseau ou autre), dans notre cas BD relationnelle.

Le modèle physique des données



- Le modèle physique (ou interne) des données permet d'établir la manière concrète dont le système sera mis en place (pour un SGBD fixé)
 - ⇒ Informations nécessaires à l'accès aux données
 - Index
 - Chemin
 - Codage ...

Schémas externes



- Adaptation du modèle logique global aux différents points de vue de l'entreprise
 - Transformation de la structure de données
 - Contraintes supplémentaires

Modèles de données

- Modèle conceptuel
 - Pour la *conception* d'applications
 - Modèle *entité-association*, UML, ...
- Modèle logique
 - Pour la *définition* de BD supportées par un SGBD
 - Modèle hiérarchique, réseau, *relationnel*, objet
- Modèle physique
 - Pour l'*implantation* du SGBD
 - Organisation et stockage des données, indexes, ...

1. Un modèle conceptuel : le modèle Entités-Associations (E/A)

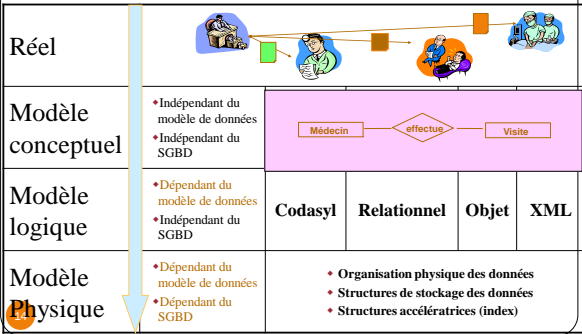
Pourquoi une modélisation conceptuelle?

- Permettre une **meilleure compréhension**
 - Le monde réel est trop complexes
 - Abstraction des aspects cruciaux du problème
 - Omission des détails
- Permettre une **conception progressive**
 - Abstractions et raffinements successifs
 - Possibilité de prototypage rapide
 - Découpage en modules ou packages
 - Génération des structures de données (et de traitements)

Elaborer un modèle conceptuel

- Isoler les **concepts fondamentaux**
 - Que vont représenter les données de la BD ?
 - Découvrir les concepts élémentaires du monde réel
 - Décrire les concepts agrégés et les sous-concepts
- Faciliter la **visualisation** du système
 - Diagrammes avec notations simple et précise
 - Compréhension visuelle et non seulement intellectuelle

Modélisation à plusieurs niveaux



Modèle Entités-Associations (E/A)

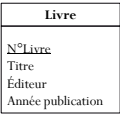
- Ensemble de concepts pour modéliser les données d'une application (d'une entreprise).
- Ensemble de symboles graphiques associés.
- Formalisé en 1976 par P.Chen, modèle très utilisé en France.
- Etendu vers E/A généralisé puis vers l'objet
 - Concepts initiaux des modèles E/A (Merise): Entité, Identifiant, Propriété, Association, Occurrence, cardinalité.
 - Extensions (MERISE/objet, MERISE/2): Héritage (Spécialisation, généralisation), Association d'associations, Contraintes d'intégrité Fonctionnelle.

Entité

- Un objet du monde réel qui peut être identifié et que l'on souhaite représenter
 - La **classe d'entité** correspond à une collection d'entités décrites par leur type commun (le format)
 - L'**instance d'entité** correspond à un élément particulier de la classe d'entité (un objet)
 - **Attention: on dit entité pour les deux !** Comprendre selon le contexte.
- Il existe généralement plusieurs entités dans une classe

Attributs et clé

- Attribut
 - Donnée élémentaire servant à caractériser les entités (et les associations).
- Clé primaire
 - Un ou plusieurs attribut(s) qui permettent d'identifier de manière unique une entité dans sa classe.
 - Soulignés dans la représentation graphique.



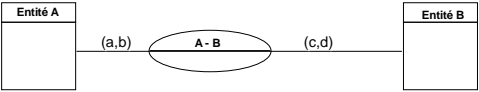
Association

- **Association** : lien existant entre entités.
 - Le nombre d'entités impliquées dans l'association donne sa dimension (souvent 2)
 - Dans une association chaque entité joue un rôle
- **Classe d'association** : ensemble d'associations qui ont une même sémantique définie sur les mêmes types d'entités.



Cardinalité

- Le schéma doit être complété par des informations quantitatives. Les quantités correspondent au **nombre de participations d'une entité** à une association.



- a) On fixe une occurrence de A, combien *au minimum* d'occurrence de B sont associés à A via A-B?
- b) *au maximum*
- c) On fixe une occurrence de B, combien *au minimum* d'occurrence de A sont associés à B par A-B?
- d) *au maximum*

Cardinalité minimum et maximum

- Les cardinalités minimum et maximum ne peuvent prendre que 3 valeurs :

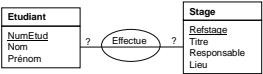
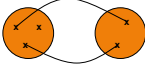
- mini { 0 Une occurrence de A peut ne pas être reliée, via l'association A-B, à une occurrence de B.
- 1 Une occurrence de A doit être reliée, via l'association A-B, à une occurrence de B.
- maxi { n Une occurrence de A peut être reliée, via l'association A-B, à plusieurs occurrences de B.

Cardinalités : exemple

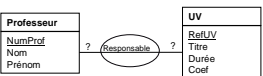
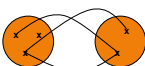


Associations et cardinalités

- Association 1-1

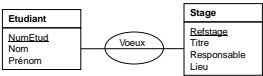
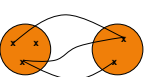


- Association 1-N

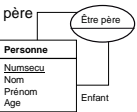


Associations et cardinalités

- Association N-M

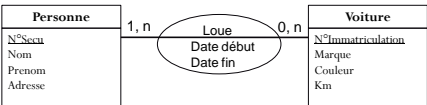


- Association réflexive



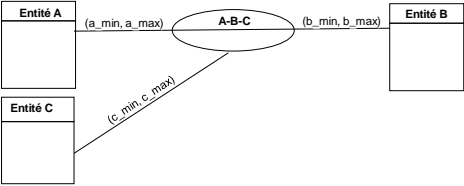
Associations et attributs

- Les associations peuvent aussi avoir des attributs.



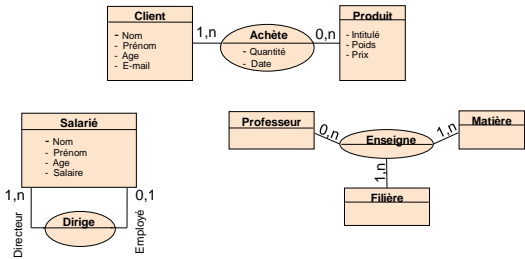
Cardinalité n-aire

- Exemple en dimension 3.



- a_min: une occurrence de A est fixé, le nombre correspond au nombre *minimum* de couple (BC) associé à A via A-B-C
- a_max: *maximum*

Exemples



(Equivalence : Entités-Association / UML)

Entité-Association	UML
Classe d'entité	Classe
Entité	Objet
Association	Relation
Cardinalité	Multiplicité
Modèle conceptuel de données	Diagramme de classes

Attention: les cardinalités E/A sont différentes des cardinalités UML !

Quelques règles

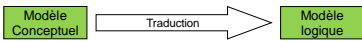
- Une entité ne peut être relié directement à un autre entité
- Il n'y a pas de restriction sur le nombre d'association entre entités
- Une association ne peut être construite directement sur une autre association. Il faut passer par une agrégation (Merise étendu)
- Il n'y a pas de restriction sur le nombre d'entités qui interviennent dans une association (de 1 à N)
- La dimension (ou ordre/degré) d'une association est le nombre d'entités distinctes qui interviennent dans une association.

Démarche

- Déterminer la liste des classes d'entités
- Pour chaque classe d'entité :
 - Etablir la liste de ses attributs
 - Parmi ceux-ci déterminer un identifiant si possible
- Déterminer les associations entre les entités
- Pour chaque association :
 - Dresser la liste des attributs propres à l'association
 - Vérifier la dimension
 - Ajouter les cardinalité
- Vérifier le schéma obtenu :
 - Supprimer les transitivités (triangle)
 - S'assurer que le schéma est connexe
- Valider avec les utilisateurs

2.Traduction en modèle relationnel (modèle logique)

Le modèle logique



- Le choix du modèle logique dépend du choix du SGBD.
- => Nous allons utiliser le modèle **relationnel**.

Traduction en modèle relationnel : Démarche

- Pour passer du modèle conceptuel au modèle logique relationnel on va appliquer un certain nombre de règles de transformation.
- Règle 1 : Toute entité est traduite en une table relationnelle.**
 - 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 l'entité forment les autres colonnes de la table.

Traduction en modèle relationnel : Démarche

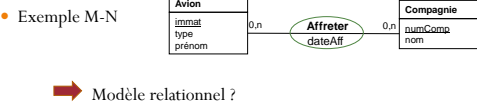
- Règle 2 : Toute Association M-N est traduite en une table relationnelle.**
 - Le nom de la table est le nom de l'association.
 - La clé de la table est formée par la concaténation des identifiants des entités participant à l'association et ceux de l'association elle-même.
 - Les attributs spécifiques de l'association forment les autres colonnes de la table.
- Remarque : une contrainte d'intégrité référentielle est générée entre chaque colonne clé de la nouvelle table et la table d'origine.

Traduction en modèle relationnel : Démarche

- Règle 3 : Toute Association 1-N est traduite.**
 - Par un report de clé : l'identifiant de l'entité participant à l'association côté n est ajouté comme colonne supplémentaire à la table représentant l'autre entité.
 - Cette colonne supplémentaire sera une *clé étrangère* (#).
 - Le cas échéant, les attributs de l'association sont aussi ajoutés à la même table.
- Remarque : génère une contrainte d'intégrité référentielle.

Traduction en modèle relationnel : Démarche

- Règle 4 : toute association 1-1 est traduite.**
 - Cas (0,1)–(1,1) : report de la clé d'une table dans l'autre.
➡ Report de la clé dans la table coté (1,1)
(Évite l'apparition de valeur nulle.)
 - Eventuellement cas (1,1) –(1,1) : fusion des entités reliées par l'association en une table.



Exemples

- Exemple 1-1



- Exemple 1-N

