

Auto-formation
MLD
Modèle Logique des
Données

Objectifs

Le modèle logique des données est une représentation du modèle conceptuel en termes d'organisation des données : il se réfère à l'état de l'art technologique c'est-à-dire des possibilités du moment en matière de logiciels et de matériels.

Généralités

Actuellement, on assiste à une mutation des logiciels de gestion de bases de données dites hiérarchiques ou navigationnelles vers des systèmes relationnels.

Le modèle relationnel présente deux aspects fondamentaux :
une algèbre permettant de manipuler des tables
une démarche de conception permettant de définir une collection de tables.

Concept de base du relationnel

La **table**, concept central du modèle, **qui** peut être définie comme un tableau de données.

Les colonnes de ce tableau sont appelées **les attributs** de la relation.

Le nombre de colonnes ou d'attributs de la table est le **degré** de la table

Chaque attribut peut prendre des valeurs dans un **domaine**

Les lignes de ce tableau, occurrence de la table, seront appelées **tuples** ou **n-uplets**.

Exemple

LIVRES
<u>N°</u>
Titre
Auteur
Editeur
Série

Donnera en formalisme relationnel

LIVRES (N°; Titre; Auteur; Editeur; Série)

Nom de la table : LIVRES

Degré de la table : 5

Attribut : N°; Titre; Auteur; Editeur; Série

Domaine de N°; Entier, 3 de long, ≠ 0

Domaine de Titre et Série; Caractère , 30 de long

Domaine de Auteur et Editeur; Caractère , 20 de long

En représentation en extension de la table

LIVRES	N°	Titre	Auteur	Editeur	Série
tuple 1	12	Tintin au Congo	Hergé	Casterman	Tintin
tuple 2	23	Raymond Calbuth	Tronchet	Fluide Glacial	Calbuth
tuple 3	345	Le comptoir de Judas	Bourgeon	Glénat	Les passagers du vent
tuple 4	24	Mafalda a un petit frère	Quino	Glénat	Mafalda
tuple 5	15	Tintin au Tibet	Hergé	Casterman	Tintin

Problèmes liés à la conception de schéma relationnel

Considérons par exemple une table P concernant des propriétaires de véhicule et les attributs de cette table

- nom : nom du propriétaire
- date : date d'acquisition du véhicule
- tél : dernier téléphone du propriétaire
- n° immat : numéro d'immatriculation du véhicule
- marque : marque du véhicule
- type : type du véhicule
- cv : puissance fiscale du véhicule
- coul : couleur du véhicule

Soit une extension de la table P

P	nom	date	tél	n° immat	marque	type	cv	coul
tuple 1	Durand	10/2/88	23323223	3344RF45	Renault	R25	9	bleu
tuple 2	Dupont	8/10/88	62625255	7787FG56	Peugeot	405GR	7	vert
tuple 3	Pagnol	7/7/89	76453434	554FG22	Volvo	245	8	blanc
tuple 4	Pagnol	21/4/90	76453434	667TG22	Peugeot	305	6	gris
tuple 5	Duval	15/8/90	78256852	129DR75	Renault	R25	9	blanc
tuple 6	Martin	10/7/90	77885234	777DG34	Ford	Sierra	7	rouge

Cette table pose, dans son utilisation , un certain nombre de problèmes liés à la redondance des données, ou lié à la nécessité d'avoir des attributs pour lesquels on accepte des valeurs nulles.

Données redondantes

La table fait apparaître une personne autant de fois qu'elle possède un véhicule.

tuple 3	Pagnol	7/7/89	76453434	554FG22	Volvo	245	8	blanc
tuple 4	Pagnol	21/4/90	76453434	667TG22	Peugeot	305	6	gris

Si Mr Pagnol change de N° de téléphone, il faut s'assurer que la mise à jour s'effectue bien sur ces deux véhicules.

Une autre redondance est lié à la correspondance marque, type,cv

tuple 1	Durand	10/2/88	23323223	3344RF45	Renault	R25	9	bleu
tuple 5	Duval	15/8/90	78256852	129DR75	Renault	R25	9	blanc

Les valeurs nulles

La table concerne à la fois des personnes et des véhicules. On pourrait vouloir garder des tuples concernant des voitures sans propriétaire ou des propriétaires sans véhicules. Dans ce cas on aurait des tuples de la forme

voiture sans propriétaire

tuple x				6234HH3 4	Peugeot	305	6	gris
---------	--	--	--	--------------	---------	-----	---	------

propriétaire sans véhicule

tuple y	Lenoir		54532134					
---------	--------	--	----------	--	--	--	--	--

Conception de schémas relationnels

Dans les systèmes relationnels, redondances et valeurs nulles sont à éviter, car elles introduisent des incohérences potentielles et compliquent l'exploitation et la manipulation des tables.

Une bonne façon de constituer un ensemble de « **bonnes tables relationnels** » est de les dériver d'un modèle conceptuel des données exprimé en formalisme entité-association, c'est l'approche MERISE.

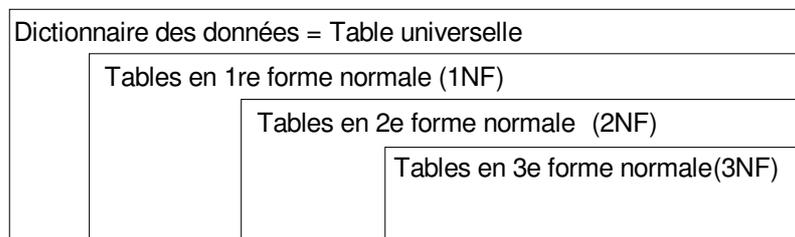
Une autre façon est basée sur le processus de normalisation proposé par Codd.

PRINCIPE DE LA NORMALISATION

On part d'une table universelle dont le schéma se compose de la totalité des attributs (dictionnaire de données), sur laquelle on applique un algorithme de normalisation.

La Normalisation se présente alors comme une décomposition de cette table de départ en plusieurs tables normalisées.

Codd a proposé trois formes normales, auxquelles ont été ultérieurement été ajoutée d'autres formes normales comme les 4^e, 5^e formes normales ou la forme de Boyce-Codd.



La première décomposition s'effectue sur **la notion de dépendance fonctionnelle**.

Un attribut B d'une table T est fonctionnellement dépendant d'un autre attribut A de T, si, à tout instant, chaque valeur de A n'a qu'une valeur associée de B : on note $A \rightarrow B$

La 0^{ème} forme normale impose que chaque table soit identifiée par une clé primaire et que tous les autres attributs de la table soient en dépendance fonctionnelle avec la clé primaire.

COMMANDE
<u>n° commande</u>
date
n° client
nom
n° article
désignation
qté commandée

La 1^{ère} forme normale a pour objet d'éliminer les groupes répétitifs dans une table. La démarche est la suivante

Sortir le groupe répétitif de la table initial.

Transformer le groupe répétitif en table, trouver sa clé et rajouter dans la clé de cette nouvelle table la clé primaire de la table initiale.

COMMANDE
<u>n° commande</u>
date
n° client
nom
n° article
désignation
qté commandée

groupe répétitif

devient

COMMANDE
<u>n° commande</u>
date
n° client
nom

ARTICLE COMMANDE
<u>n° commande</u>
<u>n° article</u>
désignation
qté commandée

La 2^{ème} forme normale concerne les tables à clé primaire composée (composée de plusieurs attributs). La règle impose que les attributs non-clé primaire dépendent de la totalité de la clé primaire. Tout attribut qui ne dépendrait que d'une partie de la clé primaire doit être exclu de la table. Le processus est le suivant.

Regrouper dans une table les attributs dépendant de la totalité de la clé, et conserver cette clé pour cette table.

Regrouper dans une autre table les attributs dépendant d'une partie de la clé, et faire de cette partie de clé la clé primaire de la nouvelle table.

ARTICLE COMMANDE
<u>n° commande</u>
<u>n° article</u>
désignation
qté commandée

devient

ARTICLE
<u>n° article</u>
désignation

ARTICLE COMMANDE
<u>n° commande</u>
<u>n° article</u>
qté commandée

La 3^{eme} forme normale a pour objet l'élimination des dépendances transitives au sein d'une table. (Rappel : La transitivité $A \rightarrow B$, $B \rightarrow C$ alors $A \rightarrow C$). La démarche est

Conserver dans la table initiale les attributs dépendant directement de la clé

Regrouper dans une table les attributs dépendant transitivement; l'attribut de transition reste dupliqué dans la table initiale, et devient la clé primaire de la nouvelle table.

COMMANDE
<u>n° commande</u>
date
n° client
nom

devient

COMMANDE
<u>n° commande</u>
date
n° client

CLIENT
<u>n° client</u>
nom

La forme normale de Boyce-Codd (BCNF) permet d'éviter des redondances dues à l'existence de dépendances fonctionnelles autres que celles de la clé vers des attributs non clés.

Soit un exemple dans lequel un employé est affecté (un certain nombre d'heures) à un certain nombre de projets effectués dans un certain nombre d'unités de fabrication (Une unité de fabrication ne traite qu'un projet donné).

AFFECTER
<u>n° employé</u>
<u>n° projet</u>
nb heures
unité fabrication

devient

AFFECTER
<u>n° employé</u>
<u>unité fabrication</u>
nb heures

UNITE FABRICATION
<u>unite fabrication</u>
n° projet

DERIVATION DU MODELE ORGANISATIONNEL

Il s'agit d'exprimer les tables à partir du schéma entité-association issu du MOD.

Cette transformation est entièrement algorithmique (voir utilisation de AMC Désigner).

Le modèle obtenu est obligatoirement en deuxième forme normale. Il n'est pas nécessairement en troisième forme, le choix des individus n'étant pas guidé sur la redondance minimum.

REGLES DE TRANSFORMATIONS

1) ENTITE :

Toute entité est transformée en table. Ses propriétés deviennent les attributs de la table. L'identifiant devient la clé primaire unique de la table.

CLIENT
<u>n° client</u>
nom
adresse
code postal
ville

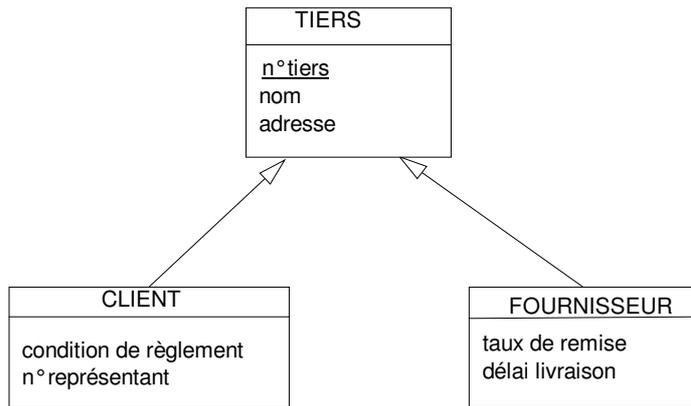
devient

CLIENT
<u>n° client</u>
nom
adresse
code postal
ville

Schéma relationnel : CLIENT (n° client, nom, adresse, code postal, ville)

2) SPECIALISATION:

On exprime les sous-types par des tables spécifiques, avec comme clé primaire celle de la table du surtype. La table du surtype est exprimé comme une entité quelconque.



devient

Schéma relationnel

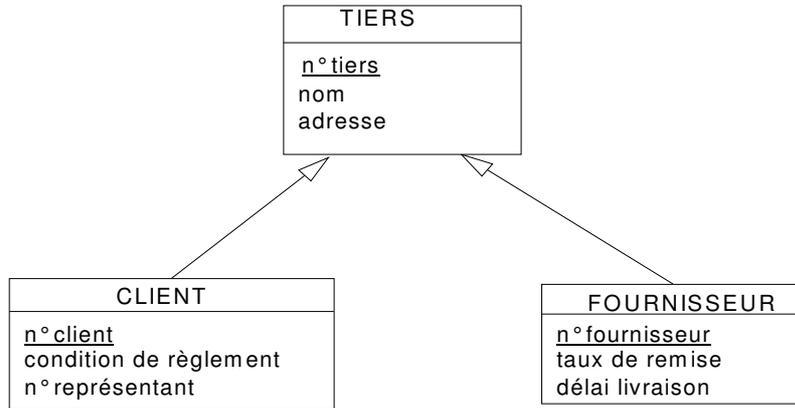
TIERS (n° tiers, nom, adresse)

CLIENT (n° tiers, condition règlement, n° représentant)

FOURNISSEUR (n° tiers, taux de remise, délai de livraison)

3) GENERALISATION:

On exprime les sous-types par des tables spécifiques, avec leur clé primaire, et la table du surtype avec sa clé primaire.



devient

TIERS
<u>n° tiers</u>
nom
adresse

CLIENT
<u>n° client</u>
n° tiers
code règlement
n° représentant

FOURNISSEUR
<u>n° fournisseur</u>
n° tiers
taux remise
délai livraison

Schéma relationnel

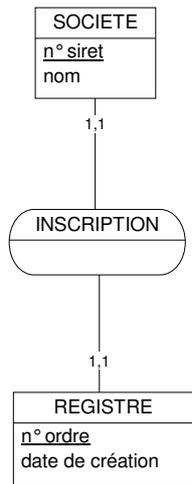
TIERS (n° tiers , nom ,adresse)

CLIENT (n° client , n° tiers, condition règlement, n° représentant)

FOURNISSEUR (n° fournisseur, n° tiers, taux de remise, délai de livraison)

4) CARDINALITE (1,1)-(1,1)

On choisit l'une des entités qui hérite de l'ensemble des attributs de l'autre entité, la clé primaire restant celle de l'entité.



devient

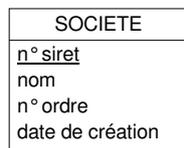
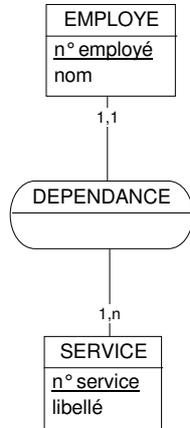


Schéma relationnel

SOCIETE (n° SIRET , nom , n° ordre, date de création)

5) CARDINALITE (1,1)-(0,n) ou (1,1)-(1,n) ou (1,1)-(0,1)

On duplique la clé de la table issue de l'entité à cardinalité (0,n) ou (1,n) ou (0,1) dans la table issue de l'entité à cardinalité (1,1) où elle devient une clé externe



devient

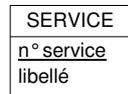
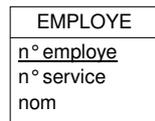


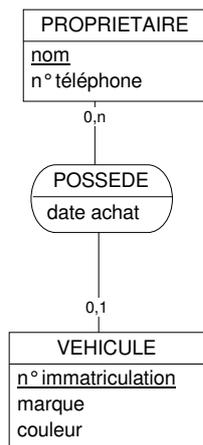
Schéma relationnel

SERVICE (n° service , libellé)

EMPLOYE (n° employé , n° service , nom)

6) CARDINALITE (0,1)-(0,n) ou (0,1)-(1,n) ou (0,1)-(0,1)

On crée une table avec comme clé primaire l'identifiant de l'entité à cardinalité (0,1) ; l'identifiant de l'autre entité devenant clé externe de cette table.



devient

Schéma relationnel

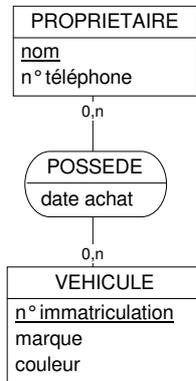
PROPRIETAIRE (nom , n° téléphone)

POSSEDE (n° immatriculation , nom , date achat)

VEHICULE (n° immatriculation , marque , couleur)

7) CARDINALITE (0,n)-(0,n) ou (1,n)-(1,n) ou (0,n)-(1,n)

On crée une table avec comme clé primaire une clé composé des identifiants des deux entités.



devient

Schéma relationnel

PROPRIETAIRE (nom , n° téléphone)

POSSEDE (n° immatriculation , nom , date achat)

VEHICULE (n° immatriculation , marque , couleur)

Exercice de synthèse 7 (sur MLD)

Le texte qui suit reprend l'exercice de synthèse 4 ,mais enrichi de plusieurs paragraphes que vous modéliserez afin de modifier le MCD d'origine.
Puis vous en déduirez le MLD

1 - Le service de formation d'une entreprise souhaite gérer ses actions de formation. Afin d'adresser des convocations aux employés, on enregistre les agents avec leur numéro, leur nom, prénom ainsi que l'établissement auquel ils appartiennent (l'entreprise est implantée dans différents lieux géographiques). De cet établissement on connaît le code, le nom et l'adresse.

Un certain nombre de cours sont offerts aux employés qui peuvent s'y inscrire. Ces cours sont connus dans un catalogue, où ils figurent avec un code et un intitulé. Tous les cours sont animés par des employés internes à l'entreprise. On s'assurera que les personnes affectées à l'animation de cours ne sont pas sujettes à une inscription à ce même cours. Personne dans cette entreprise n'ayant le don d'ubiquité.

On désire établir des convocations aussi bien pour le personnel inscrit que pour le personnel enseignant. A ce sujet, il faut savoir que tout cours est planifié plusieurs fois par an et que l'on parle plus volontiers de session. Une session est repérée par un numéro et a un intitulé (printemps, été,...). La planification consistera à décider quels cours seront offerts dans une session,et pour chacun d'entre eux quelle en sera la date et la durée,puis de les inscrire au catalogue. Les inscriptions ou animations ou convocations se font par référence aux cours et aux sessions.

Lors de la convocations on fait savoir la liste des ouvrages (dont on donne le numéro, la référence et l'intitulé) qu'il est conseillé de consulter pour un cours donné. Par ailleurs, c'est à ce moment là que l'on réservera les salles (dont on connaît le numéro et le nombre de place).

Ultérieurement on veut s'assurer que les personnes inscrites ont été présentes ou absentes.

Autres exercices de synthèse sur le MLD

Reprendre les MCD des exercices de synthèse 1, 2 et 6 et construire les MLD correspondants.

REMARQUES IMPORTANTES SUR LE MODELE LOGIQUE

Parmi les avancées majeures des années 90, l'architecture « CLIENT/SERVEUR », impose d'étudier au niveau logique la répartition des données.

SCENARIO 1

Les données sont stockés et gérées sur une seule machine.

Le MLD est global et part du MOD global

SCENARIO 2

Les données sont stockées et gérées sur plusieurs machines.

Un MLD global décrit les données permanentes dans un modèle unique indépendant de la localisation des données.

Un MLD réparti précise la structure logique des données permanentes à implanter sur chacune des machines

Un à n MLD locaux correspondant à la structure des données sur un site local.

ATTENTION la répartition des données impose une SYNCHRONISATION à préciser et valider avec les traitements. On distingue

Les applications à données réparties non synchronisées

On se contente de copies rafraîchies des données hors sites.

Les applications à données réparties synchronisées en permanence

Mise à jour simultané en temps réel.