

# Conception d'une base de données

Cyril GRUAU\*

17 octobre 2005 (corrigé le 13 juillet 2006)

## Résumé

*Ce support de cours regroupe quelques notions concernant le modélisation conceptuelle de système d'information par schéma entités-associations (via l'étude des dépendances fonctionnelles), la traduction en schéma relationnel et la démarche inverse (rétro-conception). Il présente également les extensions majeures du modèle conceptuel de données.*

**Mots-clef:** Merise, modèle conceptuel, entité, association, dépendance fonctionnelle, graphe de couverture minimale, schéma relationnel, traduction, rétro-conception, agrégation, identifiant relatif, héritage

Compléments apportés à l'édition de novembre 2003 :

- une ré-écriture complète des règles de normalisation ..... 10
- un nouveau paragraphe sur les dépendances fonctionnelles ..... 16
- une ré-écriture complète de la section sur les agrégations ..... 30
- idem pour les identifiants relatifs ..... 35
- et l'héritage ..... 38
- auxquels s'ajoutent de nouveaux exemples et donc de nombreuses figures illustratives ..... 42

## Remerciements

L'auteur tient à exprimer toute sa gratitude envers Frédéric Brouard pour son travail de correction sur ce document, ses judicieux conseils et son soutien en toutes circonstances.

---

\*Cyril.Gruau@ensmp.fr

## Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Modèle conceptuel de données (MCD)</b>	<b>4</b>
1.1 Schéma entités-associations . . . . .	4
1.1.1 Entités et associations . . . . .	4
1.1.2 Attributs et identifiants . . . . .	5
1.1.3 Cardinalités . . . . .	6
1.1.4 Associations plurielles . . . . .	7
1.1.5 Association réflexive . . . . .	7
1.1.6 Associations non binaires . . . . .	8
1.2 Règles de normalisation . . . . .	10
1.2.1 Les bonnes manières dans un schéma entités-associations . . . . .	10
1.2.2 Les formes normales . . . . .	14
1.3 Dépendances fonctionnelles . . . . .	16
1.3.1 Définitions et propriétés . . . . .	16
1.3.2 Graphe de couverture minimale . . . . .	17
1.3.3 Traduction vers un schéma entités-associations . . . . .	17
1.3.4 Gestion des dates et du caractère historique . . . . .	18
1.3.5 Dépendances plurielles et réflexives . . . . .	20
1.3.6 Associations sans attributs . . . . .	20
1.4 Méthodologie de base . . . . .	21
<b>2 Modèle logique de données (MLD)</b>	<b>22</b>
2.1 Systèmes logiques . . . . .	22
2.2 Modèle logique relationnel . . . . .	22
2.2.1 Tables, lignes et colonnes . . . . .	22
2.2.2 Clés primaires et clés étrangères . . . . .	22
2.2.3 Schéma relationnel . . . . .	23
2.3 Traduction d'un MCD en un MLDR . . . . .	24
<b>3 Modèle physique de données (MPD)</b>	<b>27</b>
3.1 Distinction entre MLD et MPD . . . . .	27
3.2 Optimisations . . . . .	27
<b>4 Rétro-conception</b>	<b>28</b>
4.1 Traduction inverse . . . . .	28
4.2 Cas particuliers . . . . .	29
<b>5 Compléments</b>	<b>30</b>
5.1 Agrégation . . . . .	30
5.1.1 Association de type 1 : n . . . . .	30
5.1.2 Association de type n : m . . . . .	32
5.1.3 Tables de codification ou tables de référence . . . . .	34
5.2 Identifiant relatif ou lien identifiant . . . . .	35
5.2.1 Résolution d'un problème sur le schéma relationnel . . . . .	35
5.2.2 Modèle conceptuel correspondant . . . . .	36
5.2.3 Discussion autour de la numérotation des exemplaires . . . . .	37
5.3 Héritage . . . . .	38
5.3.1 Sous-entité . . . . .	38
5.3.2 Utilisation de l'héritage pour séparer les informations complémentaires . . . . .	39
5.3.3 Spécialisation des associations . . . . .	40

<b>Conclusion</b>	<b>41</b>
<b>Références</b>	<b>41</b>
<b>Table des figures</b>	<b>42</b>
<b>Index</b>	<b>43</b>

## Introduction

Quand nous construisons directement les tables d'une base de données dans un logiciel de gestion des bases de données (Oracle, SQL Server, DB2, Access, MySQL, PostGre, ...), nous sommes exposés à deux types de problème :

- nous ne savons pas toujours dans quelle table placer certaines colonnes (par exemple, l'adresse de livraison se met dans la table des clients ou dans la table des commandes?) ;
- nous avons du mal à prévoir les tables de jonction intermédiaires (par exemple, la table des interprétations qui est indispensable entre les tables des films et la table des acteurs).

Il est donc nécessaire de recourir à une étape préliminaire de conception.

Les techniques présentées ici font partie de la méthodologie Merise (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) élaborée en France en 1978 [Tardieu *et al.*], qui permet notamment de concevoir un système d'information d'une façon standardisée et méthodique.

Le but de ce support de cours est d'introduire le schéma entités-associations (section 1), le schéma relationnel (sections 2 et 3) et d'expliquer la traduction entre les deux (sections 2.3 et 4). La construction du schéma entités-associations peut se faire en étudiant les dépendances fonctionnelles (section 1.3) et en tenant compte d'un certain nombre d'extensions conceptuelles incontournables (section 5).

Ne sont malheureusement pas abordés ici : les contraintes, les traitements, le langage relationnel et la gestion de projet. Pour toutes ces notions importantes, car liées à la conception de systèmes d'information, le lecteur est dirigé vers [Akoka et Comyn-Wattiau, Matheron, Nanci *et al.*]. La modélisation objet ne fait pas non plus partie des outils exposés dans ce document.

## 1 Modèle conceptuel de données (MCD)

Avant de réfléchir au schéma relationnel d'une application, il est bon de modéliser la problématique à traiter d'un point de vue conceptuel et indépendamment du logiciel utilisé.

### 1.1 Schéma entités-associations

La modélisation conceptuelle que nous proposons dans ce document pour un univers dont on veut stocker les données, conduit à l'élaboration d'un type de schéma très répandu, le schéma entités-associations.

#### 1.1.1 Entités et associations

Une entité est une population d'individus homogènes. Par exemple, les produits ou les articles vendus par une entreprise peuvent être regroupés dans une même entité **articles** (figure 1), car d'un article à l'autre, les informations ne changent pas de nature (à chaque fois, il s'agit de la désignation, du prix unitaire, etc.).

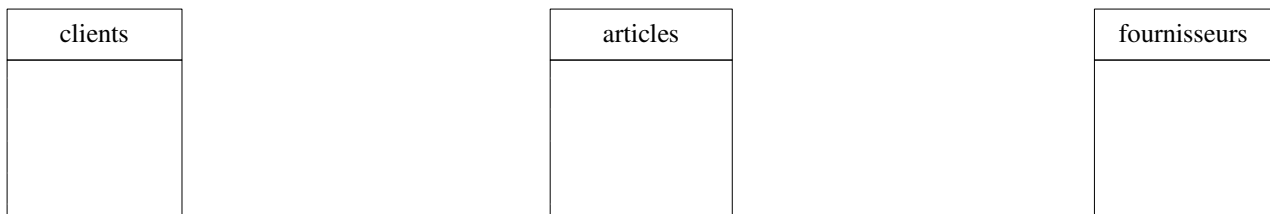


FIG. 1 – Entités

Par contre, les articles et les clients ne peuvent pas être regroupés : leurs informations ne sont pas homogènes (un article ne possède pas d'adresse et un client ne possède pas de prix unitaire). Il faut donc leur réserver deux entités distinctes : l'entité **articles** et l'entité **clients**.

Une association est une liaison qui a une signification précise entre plusieurs entités. Dans notre exemple, l'association **commander** est une liaison évidente entre les entités **articles** et **clients**, tandis que l'association **livrer** établit le lien sémantique entre les entités **articles** et **fournisseurs**.

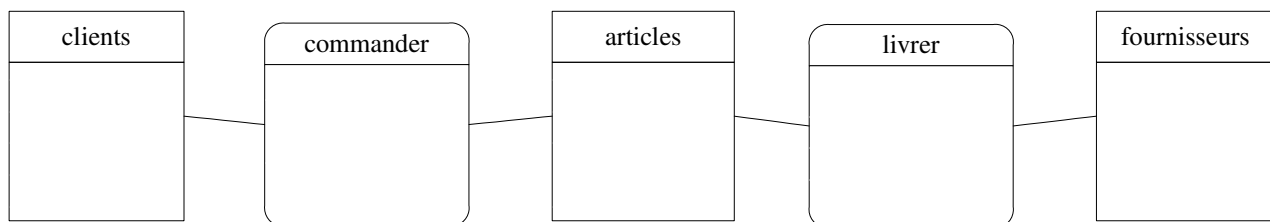


FIG. 2 – Associations

Remarquons que dans ce schéma, les entités **clients** et **fournisseurs** ne sont pas liées directement, mais indirectement, via l'entité **articles**, ce qui est assez naturel.

### 1.1.2 Attributs et identifiants

Un attribut est une propriété d'une entité ou d'une association.

Toujours dans notre exemple (figure 3), le **prix unitaire** est un attribut de l'entité **articles**, le **nom de famille** est un attribut de l'entité **clients**, la **quantité commandée** est un attribut de l'association **commander** et la **date de livraison** est un attribut de l'association **livrer**.

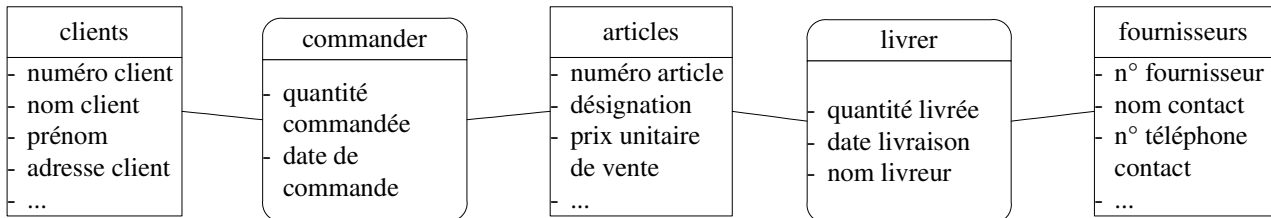


FIG. 3 – *Attributs*

Une entité et ses attributs ne doivent traiter que d'un seul sujet afin d'assurer une certaine cohérence au modèle. Dans notre exemple, il est donc préférable de ne pas mettre les informations relatives aux fournisseurs dans l'entité des articles mais plutôt dans une entité **fournisseurs** séparées (et liée à l'entité **articles** via l'association **livrer**).

Ensuite, chaque individu d'une entité doit être identifiable de manière unique. C'est pourquoi toutes les entités doivent posséder un attribut sans doublon (c'est-à-dire ne prenant pas deux fois la même valeur). Il s'agit de l'identifiant que l'on souligne sur le schéma, par convention. Le numéro de client constitue un identifiant classique pour l'entité **clients** (figure 4).

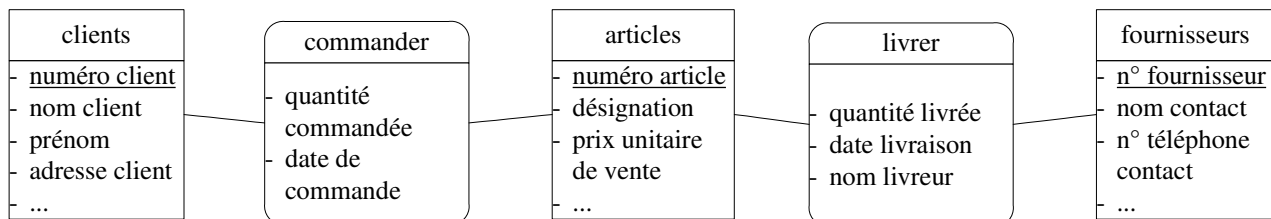


FIG. 4 – *Identifiants*

Remarques :

- une entité possède au moins un attribut (son identifiant) ;
- au contraire, une association peut être dépourvue d'attribut.

### 1.1.3 Cardinalités

La cardinalité d'un lien entre une entité et une association précise le minimum et le maximum de fois qu'un individu de l'entité peut être concerné par l'association.

Exemple : un client a au moins commandé un article et peut commander  $n$  articles ( $n$  étant indéterminé), tandis qu'un article peut avoir été commandé entre 0 et  $n$  fois (même si ce  $n$  n'est pas le même  $n$  que précédemment). On obtient alors le schéma entités-associations complet<sup>1</sup> (figure 5).

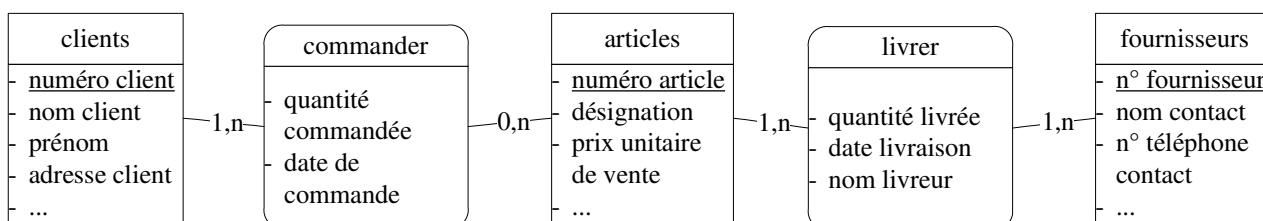


FIG. 5 – Cardinalités

Une cardinalité minimale de 1 doit se justifier par le fait que les individus de l'entité en question ont besoin de l'association pour exister (un client n'existe pas avant d'avoir commandé quoique ce soit, donc la cardinalité minimale de l'entité **clients** dans l'association **commander** est 1). Dans tous les autres cas, la cardinalité minimale vaut 0 (c'est le cas pour une liste pré-établie d'articles par exemple).

Ceci dit, la discussion autour d'une cardinalité minimale 0 ou 1 n'est vraiment intéressante que lorsque la cardinalité maximale est 1. Nous verrons en effet lors de la traduction vers un schéma relationnel (section 2.3), que lorsque la cardinalité maximale est  $n$ , nous ne pouvons pas faire la différence entre une cardinalité minimale de 0 et une cardinalité minimale de 1.

Notons que sur notre exemple, un article peut être commandé par plusieurs clients. Cela provient du fait que tous les crayons rouges ne sont pas numérotés individuellement, mais portent un numéro d'article collectif. En toute rigueur, notre entité **articles** aurait dû s'appeler **types d'article**. Ainsi, un crayon rouge peut être commandé par plusieurs clients, ce n'est simplement pas le même crayon à chaque fois. Il s'agit d'un choix de modélisation, le lecteur peut très légitimement faire le choix inverse qui consiste à numéroté individuellement chaque crayon rouge.

La seule difficulté pour établir correctement les cardinalités est de se poser les questions dans le bon sens. Autour de l'association **commander**, par exemple :

- côté **clients**, la question est « un client peut commander combien d'articles ? » et la réponse est « entre 1 et plusieurs » ;
- côté **articles**, la question est « un article peut être commandé par combien de client ? » et cette fois-ci, la réponse est « entre 0 et plusieurs ».

1. Le lecteur avisé aura noté que le schéma de la figure 5 comporte des erreurs de conception. Ces erreurs seront corrigées dans la section 1.2 dédiée à la normalisation des schémas entités-associations.

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.

