



**HAL**  
open science

# Apprentissage de la conception des bases de données : une méthodologie de la méthode

André Delacharlerie

► **To cite this version:**

André Delacharlerie. Apprentissage de la conception des bases de données : une méthodologie de la méthode. Colloque francophone sur la didactique de l'informatique, Sep 1988, Paris, France. pp.263-282. edutice-00373774

**HAL Id: edutice-00373774**

**<https://edutice.hal.science/edutice-00373774>**

Submitted on 7 Apr 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**APPRENTISSAGE DE LA CONCEPTION  
DES BASES DE DONNEES  
Une méthodologie de la méthode**

**André Delacharlerie**

**Facultés Universitaires Notre-Dame de la Paix  
61, rue de Bruxelles  
B-5000 NAMUR**



**APPRENTISSAGE DE LA CONCEPTION  
DES BASES DE DONNEES  
Une méthodologie de la méthode**

**André Delacharlerie**

**Facultés Universitaires Notre-Dame de la Paix  
61, rue de Bruxelles  
B-5000 NAMUR**

Les progiciels en général et les systèmes de gestion de fichiers et de bases de données occupent depuis quelques années une place prépondérante dans le monde de la micro-informatique. L'enseignement ne pouvait y rester insensible et l'on assiste aujourd'hui à la multiplication de formations spécifiques. Comme beaucoup d'autres organismes, le CeFIS (Centre de Formation à l'Informatique pour le Secondaire) a mis, depuis 4 ans, des cours traitement de texte et gestion de fichiers et de bases de données à son programme.

Mettre au point une méthodologie adaptée à l'enseignement des progiciels n'est pas chose facile. Je vais, dans ces quelques pages, présenter, non seulement les objectifs qui ont été poursuivis au CeFIS mais surtout les méthodes employées et les difficultés rencontrées. Je dois néanmoins insister sur le fait qu'il s'agit de l'état actuel de ma réflexion et de ma pratique. Vous verrez qu'elle a évolué au fil des expériences menées et j'espère que cette méthodologie continuera à s'affiner et à s'enrichir.

Il me faut aussi souligner, dès à présent, que ma vision de l'apprentissage des progiciels doit énormément à l'enseignement et aux conseils de Charles DUCHATEAU, directeur du CeFIS et professeur aux Facultés Universitaires Notre-Dame de la Paix à Namur, ainsi qu'aux échanges avec mes collègues du Département Education et Technologie des Facultés.

Mes contacts avec Jean-Luc HAINAUT, professeur à l'Institut d'Informatique des Facultés et co-auteur d'un "Atelier de conception de bases de données" [1] [6] m'ont également permis d'élargir et d'affiner ma perception de la problématique des bases de données et de leur conception.

Pour bien cerner l'esprit dans lequel se situe mon approche, il me semble intéressant de préciser le contexte historique dans lequel sont nés les cours "progiciels" du CeFIS.

## 1. LE CONTEXTE

Le CeFIS est né en 1981 de l'initiative de Charles DUCHATEAU qui, constatant l'apparition, timide à l'époque, de l'informatique dans les écoles, désirait apporter une aide aux enseignants du secondaire en partageant avec eux l'expérience de son apprentissage de l'informatique. Les activités du CeFIS se concentrèrent immédiatement sur l'apprentissage de la programmation et ce pour des raisons assez diverses. D'une part, les enseignants, souvent des mathématiciens, allaient eux-mêmes devoir organiser un cours de programmation et d'autre part, les micro-ordinateurs de l'époque disposaient, en général, d'un interpréteur BASIC intégré mais manquaient de mémoire vive suffisante pour "faire tourner" des progiciels tels que ceux que nous connaissons aujourd'hui. Enfin, l'approche algorithmique était et reste un moyen privilégié pour mettre en évidence les concepts propres à l'informatique.

Bientôt cependant, l'évolution, tant du matériel que des logiciels et la demande de notre public, nous a conduit à proposer des cours de traitement de texte tout d'abord et de gestion de fichiers ensuite.

Dès le début et à l'image de l'esprit qui prévalait pour l'apprentissage de la programmation, ces cours ont été centrés sur l'apprentissage des concepts fondamentaux et transférables et non pas sur l'acquisition d'habiletés spécifiques à tel ou tel logiciel. Bien sûr, un tel apprentissage doit nécessairement s'appuyer sur la manipulation concrète d'un logiciel type mais cet aspect, bien qu'indispensable est loin d'être suffisant, surtout si l'on se rappelle que le public visé n'est pas (seulement) formé d'utilisateurs mais surtout d'enseignants qui devront peut-être, à leur tour, organiser de tels cours mais bien souvent avec des matériels et des logiciels différents.

Notre objectif est donc de faire acquérir, en un laps de temps relativement court, les concepts fondamentaux qui permettent d'entrer dans le monde de l'informatique et des progiciels en particulier. Notre enseignement vise aussi à donner des méthodes de travail éprouvées mais malgré tout assez simples pour que l'utilisateur soit capable d'aborder des problèmes de complexité moyenne.

En ce qui concerne plus particulièrement l'activité "Gestion de fichiers et bases de données", notre approche a été bridée au début par le fait que le seul logiciel que nous pouvions présenter en illustration du cours était monofichier et fort limité dans ses possibilités<sup>(1)</sup> Ce n'est en fait qu'avec l'emploi de micro-ordinateurs compatibles PC et du logiciel dBASE III que le cadre s'est révélé assez large pour présenter valablement une méthode de conception de base de données et pour pouvoir la mettre en pratique sur des exemples non triviaux.

Dans le souci de rendre compte des résultats des recherches les plus récentes tant en gardant un langage accessible à notre public constitué de non spécialistes, nous n'hésiterons pas le temps de "démarrer l'apprentissage", à remplacer les termes trop sophistiqués par des vocables que nous jugeons plus expressifs et donc plus accessibles à l'apprenant. Nous n'hésiterons pas à parler ainsi de "marche à suivre" plutôt que d'algorithme, de "portrait d'une réalité" que de schéma conceptuel, etc.

Enfin, je clôturerai et je résumerai cette introduction en rappelant que notre démarche, à Charles DUCHATEAU et à moi-même, qui ne sommes ni l'un ni l'autre de "vrais informaticiens"<sup>(2)</sup>, est fondamentalement basée sur notre désir d'épargner aux enseignants qui nous consultent les tâtonnements qui nous ont été nécessaires pour approcher les concepts importants de la programmation d'une part et des progiciels de gestion de données, d'autre part.

## **2. ORGANISATION DU COURS "BASES DE DONNÉES"**

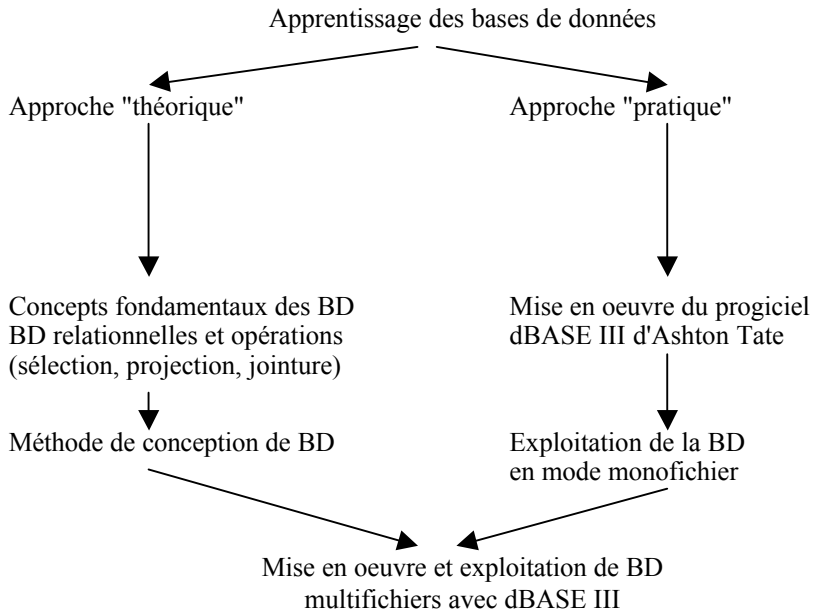
Je vais dorénavant me centrer sur la façon dont j'organise le cours consacré à la conception et à l'exploitation des bases de données. Bien qu'ayant déjà été présenté quatre fois à des publics aux caractéristiques et aux aspirations différentes (un groupe de jeunes diplômés de l'enseignement supérieur, un groupe d'inspecteurs de l'enseignement de l'Etat et deux groupes d'enseignants) je ne prétends pas détenir une méthode complète et définitive pour l'apprentissage de bases de données. J'espère seulement avoir posé des jalons solides sur un itinéraire qu'il me faudra encore certainement corriger et améliorer au contact d'expériences et de réactions nouvelles.

---

(1) Il s'agissait du logiciel PFS fonctionnant sur des micro-ordinateurs Apple II+.

(2) J'ai personnellement commencé par enseigner la mathématique dans le secondaire avant de m'initier progressivement à l'informatique.

Pratiquement, je propose deux axes d'apprentissages complémentaires qui sont conduits en parallèle<sup>(3)</sup> :



Ces deux axes permettent de mettre en évidence les deux types de tâches que l'utilisateur potentiel de BD va devoir exécuter, à savoir la conception de la base de données d'une part et son exploitation quotidienne d'autre part. Afin de ne pas disperser mon propos, je me contenterai de proposer ma façon d'envisager l'enseignement d'une méthode de conception de base de données. Au moment d'aborder cette phase du cours, les apprenants ont déjà reçu une première familiarisation avec les concepts et le vocabulaire spécifique au monde des BD.

### 3. CONCEPTION DE LA BASE DE DONNÉES

Dans une première approche, on peut dire que concevoir une base de données c'est notamment mettre en évidence les informations qu'il conviendra de stocker et imaginer pour elles une structure, une organisation qui permette par la suite une gestion facile et rapide de ces informations. Il est à ce moment assez facile de montrer par quelques exemples que le fait de représenter une même information à plusieurs endroits constitue un danger potentiel d'incohérence pour la base de données. Si chaque fiche "Elève" d'une

---

(3) Je découpe chaque cours de 4 heures (une après-midi) en deux parties à peu près égales.

école reprend les nom et adresse des parents, on conçoit bien qu'avec un peu d'inattention un changement d'adresse risque fort de ne pas être répercuté sur toutes les fiches des enfants d'une même famille. L'information que l'on peut alors extraire de la base de données n'est pas cohérente : on dira que la BD a perdu son intégrité.

Il ne faut pas perdre de vue que la base de données et le progiciel qui en permet la gestion ne sont que des outils destinés à être employés par des êtres humains. Il s'ensuit que toute l'attention du concepteur devra être centrée sur la nécessité d'établir un maximum de "garde-fous" pour limiter le plus possible le risque d'introduire des "erreurs" dans la base de données. Au premier rang de ces mesures visant à assurer l'intégrité de l'information, vient la suppression de la redondance : une information spécifique ne peut être présente qu'une seule fois dans la base.

La présentation du modèle relationnel (ou je préfère parler de tables, de colonnes et de lignes que de relations, d'attributs et de n-upplet) montre une méthode d'organisation simple qui permet de résoudre élégamment le problème de la redondance. L'auditoire est généralement très satisfait de la solution proposée. Si l'exemple est simple (une table ELEVE et une table PARENT pour résoudre le problème évoqué plus haut) cela paraît même évident. En intégrant à ce schéma d'autres "éléments" (tels que cours, classes, locaux, profs, ...) on arrive vite à se poser des questions au sujet de la répartition des informations dans les différentes tables. Il est dès lors intéressant de s'armer d'une méthode de travail assez systématique pour conduire le concepteur depuis l'observation du "réel" (l'école, par exemple) jusqu'au schéma de la base adaptée à l'outil logiciel qui aura été retenu. Ce sera dBASE III dans notre cas mais R:BASE, Paradox, Ommis, voire même Oracle pourraient très bien convenir.

L'expérience montrera que ce choix n'est certes pas gratuit et conditionne sérieusement la facilité ou la rapidité d'accès aux informations. Il n'en reste pas moins secondaire comparé à la nécessité de mettre en oeuvre une organisation de BD qui permette l'accès à un maximum d'informations tout en veillant à en assurer l'intégrité.

La méthode de conception que je propose n'est autre qu'une synthèse simplifiée des méthodes les plus employées aujourd'hui dans l'informatique de gestion<sup>(4)</sup>. Pour le public qui nous occupe, le fait même de proposer une méthode est à la fois sécurisant et novateur. Ces deux qualités conduisent à éloigner un peu plus le traditionnel "feeling" du professionnel que ne peut espérer le débutant. Par ailleurs, les simplifications apportées n'ont pas seulement pour but d'adapter le discours au public mais aussi de le débarrasser

---

(4) Méthode MERISE [7], méthode d'analyse et de conception proposée par le groupe GALACSI [4] [5] ainsi que les méthodes enseignées à l'Institut d'informatique des Facultés Universitaires N-D de la Paix [1] [6]



des aspects qui ne pourront de toute façon pas être mis en oeuvre avec les progiciels destinés aux micro-ordinateurs<sup>(5)</sup>.

### 3.1. QUE VOIS-JE ?

Voilà bien la première question à se poser : toute base de données est supposée contenir des informations "attachées" à une partie plus ou moins grande du monde réel. Tantôt, il s'agira d'une école, d'une bibliothèque, d'un cabinet médical ... Nous allons nous intéresser, à titre d'exemple, à construire une base de données adaptée aux besoins d'une institution qui dispense des formations à des enseignants telle que le CeFIS.

Pour être sûr de réaliser un outil qui "colle" à la réalité (retenir les informations utiles, toutes celles qui sont utiles mais aussi rien que celles qui sont utiles) il convient de réaliser d'abord un portrait le plus fidèle et le plus précis possible de cette "réalité". Il faudra veiller aussi à être très concis en ne reprenant que les informations qui présentent (ou présenteront dans un avenir raisonnable<sup>(6)</sup>) un intérêt réel pour l'utilisateur. Répondre à la question "Que vois-je ?", c'est donc établir ce que les professionnels appelleront un schéma conceptuel.

J'ai ici retenu le modèle entité - association comme outil de représentation du schéma conceptuel. Outre le fait qu'il est celui dont l'usage est le plus répandu, il offre l'avantage d'être facilement compréhensible par chacun du fait de son faible niveau de formalisme.

Les étapes préconisées pour l'établissement du schéma conceptuel peuvent être décomposées comme suit :

**a) Relever les informations qui sont manipulées et les règles qui sont appliquées lors de la gestion de "l'organisme" que l'on se propose d'informatiser.**

Dans le cadre de la démarche décrite ici, l'approche intuitive, basée d'une part sur la discussion avec les utilisateurs réels et d'autre part, sur l'examen des documents existants (fichiers manuels, formulaires, ...) est suffisamment efficace si elle est réalisée avec soin. On pourra ainsi dresser une sorte de dictionnaires des informations.

Dans le cas de l'institution de formation, on y trouve notamment :

- Intitulé du cours
- Nom de l'étudiant <sup>(7)</sup>

---

(5) Les méthodes "classiques" visent généralement l'informatisation d'entreprises entières et conduisent à des outils (base de données et applications) nécessitant des moyens puissants (mini-ordinateurs ou mainframe).

(6) La durée de la vie des systèmes informatiques n'est pas bien longue !

(7) Rappelons que dans le cadre des formations du CeFIS, ces "étudiants" sont des enseignants.

- Adresse de l'étudiant
- Nom de l'institution (où travaille l'étudiant)
- Adresse de l'institution
- Prix du cours
- Date de début du cours
- Date de naissance de l'étudiant
- ...

On perçoit bien ici le rôle de la langue : il s'agit de désigner par un vocable non équivoque toutes les informations manipulées. Cela n'est pas une tâche simple car il faut de plus repérer et éliminer les informations qui peuvent être déduites ou calculées à partir d'autres informations déjà répertoriées. Chacune de ces informations constitueront bientôt les attributs de notre schéma entité - association.

La démarche employée pour effectuer cette recherche peut prendre tour à tour une orientation descendante (on met en évidence des "objets" tels que étudiant, institution, ... et on en relève les caractéristiques pertinentes) ou ascendante (on relève dans des documents ou par des interviews, des informations que l'on rattache ensuite aux entités). L'essentiel est d'obtenir un dictionnaire d'informations le plus complet possible.

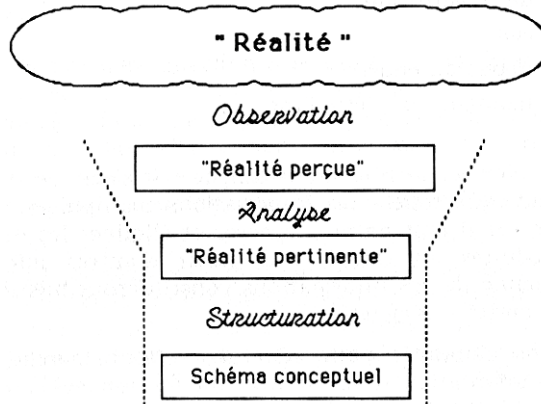
Cette description doit nécessairement être accompagnée d'une série d'expressions décrivant les règles de fonctionnement de "l'organisme". En voici quelques exemples :

- \* Un étudiant peut s'inscrire à plusieurs cours
- \* On doit pouvoir enregistrer les paiements des cours
- \* On ne retient qu'une seule institution par étudiant
- \* Les étudiants ne peuvent être âgés de plus de 30 ans
- \* ...

Cette phase du travail est particulièrement importante car elle conditionnera toute la suite des opérations. Il convient donc de la mener avec soin en veillant à n'écarter d'office aucune proposition.

Au point de vue méthodologie, j'ai souvent trouvé avantageux de proposer cette phase de travail dans le cadre d'un "devoir à domicile" puis d'organiser ensuite une synthèse commune où toutes les idées sont reprises et confrontées.

La figure suivante illustre cette première phase du "Que vois-je ?" qui va se poursuivre par la structuration des informations pour former le schéma conceptuel.



**b) Elaborer l'ossature du schéma en regroupant les attributs dans des types d'entités et en installant des types d'associations.**

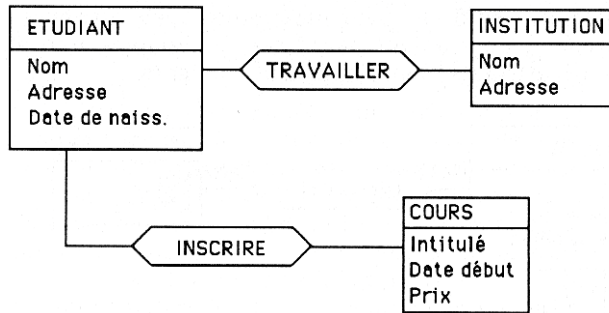
On peut très vite et très naturellement, en examinant le dictionnaire des attributs, regrouper ceux d'entre eux qui vont "manifestement" ensemble. On regroupe ainsi d'une part nom de l'étudiant et adresse de l'étudiant, d'autre part intitulé du cours, prix du cours, date de début du cours, et ainsi de suite. Chaque groupement reçoit un nom de type d'entité et l'on aboutit à un schéma du type suivant :

ETUDIANT
Nom Adresse Date de naiss.

INSTITUTION
Nom Adresse

COURS
Intitulé Date début Prix

La lecture des règles de fonctionnement permet de compléter le schéma en y ajoutant les types d'associations (que j'appellerai plus souvent liaisons) qui relient les types d'entités :



Ces types d'associations sont généralement repérés par un verbe traduisant, par exemple, qu'un étudiant est associé à une institution parce qu'il y travaille. On notera que les liaisons peuvent aussi comporter des attributs. C'est le cas ici pour la liaison INSCRIRE qui reçoit un attribut "Payé" pour permettre l'enregistrement du paiement des inscriptions.

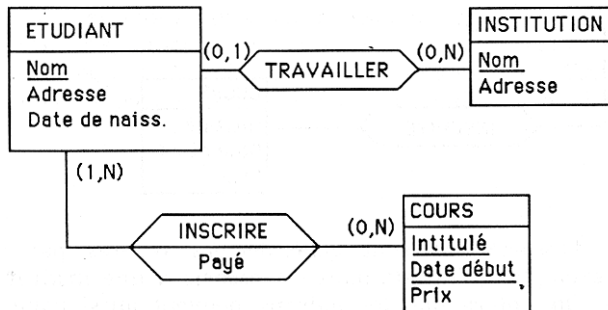
Cette solution ne doit évidemment pas être donnée directement par l'enseignant mais doit provenir d'une réflexion où l'on aura évoqué la possibilité d'attacher l'attribut payé à l'étudiant ou au cours. Les deux solutions ne pouvant qu'être rejetées dès que l'on considère qu'un étudiant peut avoir payé pour un cours et pas pour un autre et parallèlement un cours peut être payé par un étudiant et pas par un autre. Cette réflexion conduit aussi à apporter un raffinement supplémentaire au schéma qui fera l'objet de l'étape suivante.

### c) Préciser les connectivités des liaisons et déterminer les identifiants.

Il s'agit de quantifier le nombre d'associations effectives que l'on acceptera au départ des entités de chaque type. Par exemple, "A combien d'institutions peut être lié un étudiant particulier ?", la réponse que l'on fournira est rarement une valeur unique mais plutôt un intervalle du style "A zéro ou a une institution". Aussi la représentation adoptée sera un couple de valeurs exprimant le minimum et le maximum d'associations autorisées. Dans la pratique, si le maximum est supérieur à un, on se contentera de la valeur symbolique N signifiant "plusieurs" car, hormis des cas très particuliers, la suite de la démarche de conception ne sera pas différente si cette valeur est 3, 5, 10 ou 50.

Du point de vue méthodologique, on notera que ces connectivités se déduisent en général des règles de fonctionnement élaborées au début. Il est d'ailleurs probable que l'on pourra mettre en évidence des règles implicites qui n'avaient pas été répertoriées jusqu'alors.

Le choix des identifiants de chaque type d'entité est aussi une étape importante. Je ne m'étendrai pas sur ce sujet pour ne pas déplacer l'intérêt de cette communication vers des aspects trop techniques. A ce stade-ci, le bon sens est largement suffisant pour déterminer quels attributs permettent d'identifier clairement un étudiant parmi ses condisciples, une institution,... Dans le schéma, les attributs identifiants sont soulignés.



Les règles de fonctionnement qui n'auraient, jusqu'ici, pu être traduites sur le schéma représentent des contraintes d'intégrité qui seront reprises en annexe du schéma et qui devront, dans la mesure du possible, être traduites lors de la mise en oeuvre physique de la base de données<sup>(8)</sup>. C'est le cas de notre règle "les étudiants ne peuvent être âgés de plus de 30 ans". On pourrait montrer qu'il est possible d'écrire, en dBASE par exemple, une procédure qui refuse l'inscription de personnes ayant dépassé l'âge limite.

### Remarques

- Tout comme la phase de spécification (Quoi faire ?) est capitale en programmation, la définition du schéma conceptuel est aussi essentielle pour la conception d'une base de données. Cette étape ne pourra cependant jamais être automatisée (entièrement) et c'est heureux car contrairement à une photographie qui intégrerait toutes les composantes de la "réalité" que l'on veut représenter, la phase " Que vois-je ?" permet de ne pas tout voir mais de mettre plus ou moins en évidence telle ou telle caractéristique en fonction de l'usage que l'on en fera par la suite. Le schéma conceptuel n'est donc pas unique mais lié à la destination que l'on se propose d'en faire. Il en résulte que pour cerner avec un maximum de précision cette "réalité" avant d'en faire le "portrait" il y a lieu de montrer la place prépondérante que prendra la discussion entre concepteur(s) et demandeurs/futurs utilisateurs de la base de données en accordant à cette phase une tranche de temps suffisamment importante.

- Le rôle de l'animateur est ici fort important : il devra faire sentir et faire mettre au jour toutes les "règles de fonctionnement" qui passent habituellement pour tellement

<sup>(8)</sup> Via des programmes spécifiques de validation des données, par exemple.

évidentes qu'elles ne sont pas relevées naturellement. C'est pour cela que l'on choisira de préférence au moins deux sujets de BD différents : un premier appartenant à un monde qui n'est pas directement la spécialité des apprenants mais facilement compréhensible pour tous<sup>(9)</sup>. Cela permet à tout le monde de proposer des informations et des règles sans qu'une personne détienne LA solution. On voit aussi très bien qu'une même "réalité" peut prendre différents visages suivant "l'usage" que l'on désire en faire. Le second sujet de DB pourra être choisi dans la spécialité des apprenants. Un premier projet peut alors être réalisé par chacun (devoir) avant la mise en commun où l'animateur jouera le rôle du candidat qui se fait expliquer et justifier les informations et les règles retenues.

### 3.2. COMMENT ORGANISER ?

Voici la seconde étape de la démarche de conception. Il s'agit d'adapter le "portrait de la réalité" pour qu'il puisse être géré par un Système de Gestion de Bases de Données (SGBD) construit sur le modèle relationnel. Il faut bien noter qu'il s'agit bien ici d'organiser les informations suivant un modèle pour obtenir le schéma logique de la base de données. Les questions d'adaptations spécifiques à un logiciel donné ne sont pas abordées ici; elles feront, entre autre, le sujet de la troisième étape (Comment exprimer ?).

Cette étape est particulièrement "mécanique" car il suffit d'appliquer une batterie de règles de dérivation<sup>(10)</sup> pour passer très facilement du schéma conceptuel (entité-association) au schéma logique (relationnel)<sup>(11)</sup>. En voici, à titre d'exemple, l'application au schéma que nous avons construit précédemment.

- a) **Tout type d'entité (TE) donne naissance à une table dont les colonnes sont formées par les attributs du TE.**

ETUDIANT			
<u>Nom</u>	Adresse	Date de naiss.	

INSTITUTION		
<u>Nom</u>	Adresse	

COURS			
<u>Intitulé</u>	<u>Date début</u>	Prix	

(9) Je choisis souvent l'exemple d'une société d'assurance automobile gérant des contrats "responsabilité civile" et "omnium" liés à des véhicules ayant des propriétaires et pouvant avoir des accidents.

(10) Ces règles sont présentées dans [5] page 64.

(11) Il va de soi qu'au cas où le schéma logique doit être établi suivant un autre modèle (par exemple le modèle réseau) il faut utiliser d'autres règles de dérivation.

Pour bien montrer que l'on risque de devoir ajouter de nouvelles colonnes aux différentes tables, j'ai pris l'habitude de les représenter "trop larges". De plus, il convient de plus d'indiquer clairement la clé primaire de chaque table en soulignant les noms des colonnes qui la génère. A ce stade, on se contentera de reprendre simplement l'identifiant qui avait été désigné dans le schéma conceptuel.

- b) Tout type d'association ayant (au moins) une connectivité (0,1) ou (1,1) sera représenté en ajoutant, à la table "située du côté (0,1) ou (1,1)" une colonne reprenant la clé primaire de la table à associer.**

Dans notre exemple, c'est le cas de la liaison TRAVAILLER qui se traduit dans la table ETUDIANT par l'ajout d'une colonne Nom-Instit. qui reprendra le nom de l'institution où travaille l'étudiant (et ce nom seulement).

ETUDIANT				
<u>Nom</u>	Adresse	Date de naiss.	<i>Nom-instit.</i>	

Bien que la littérature ait consacré le terme de clé étrangère pour désigner une telle colonne, je préfère parler de colonne "référence". Pour la mettre en évidence, j'écris son nom en italique (ou en couleur) : elle possède en effet un statut bien différent des autres colonnes de la table car elle est le point d'ancrage de toute une série d'autres informations présentes ... dans une autre table.

- c) Les autres types d'associations (n'ayant donc que des connectivités (0,N) ou (1,N) ) donnent naissance à autant de tables nouvelles dont les clés sont formées par la réunion des clés primaires des tables associées. Le cas échéant, les attributs propres aux types d'associations forment des colonnes supplémentaires aux tables correspondantes.**

Notre liaison INSCRIRE donne donc naissance à une table INSCRIPTION comme suit :

INSCRIPTION			
<i>Nom</i>	<i>Intitulé</i>	<i>Date début</i>	Payé

C'est tout, il ne reste qu'à rassembler toutes les tables obtenues pour avoir le schéma logique de la base de données relationnelle qui corresponde exactement au "portrait" de la réalité" établi antérieurement.

ETUDIANT			
<u>Nom</u>	Adresse	Date de naiss.	<i>Nom-instit.</i>

INSTITUTION	
<u>Nom</u>	Adresse

COURS		
<u>Intitulé</u>	<u>Date début</u>	Prix

INSCRIPTION			
<u>Nom</u>	<u>Intitulé</u>	<u>Date début</u>	Payé

L'aspect très systématique de cette transformation entre schéma conceptuel et schéma logique souligne encore une fois l'importance de l'étape "Que vois-je ?" Toutes les qualités mais aussi tous les défauts (s'il y en a), de la première analyse vont se retrouver dans le schéma logique. Seule une normalisation des tables permettrait de corriger certaines erreurs de conception ; mais c'est là un sujet que nous n'aborderons pas dans ces pages.

Il est, par contre, une optimisation qui vient assez naturellement à l'esprit lorsque l'on examine le schéma logique si celui-ci comporte des "références" à des tables ayant une clé composée (c'est-à-dire formée de plus d'une colonne). En effet, de telles références risquent d'être lourdes à gérer. Dans l'exemple qui nous sert de fil conducteur, on trouve cette situation avec la clé de la table COURS qui est citée en référence dans la table INSCRIPTION. Cette référence risque d'être "encombrante" car, outre sa longueur (un intitulé d'au moins une dizaine de caractères plus une date), on devra la reprendre pour chaque inscription. Aussi, il est raisonnable d'envisager de remplacer cette référence par un numéro code qui deviendra, par la même occasion, l'identifiant du cours. Ce code, composé de quelques caractères seulement<sup>(12)</sup> sera bien plus "léger" à employer comme référence lors de l'enregistrement des inscriptions.

On sera néanmoins attentif au fait qu'une telle modification du schéma doit être accompagnée de l'énoncé d'une contrainte d'intégrité supplémentaire qui dira qu'il n'existe

---

(12) Un code formé de deux lettres seulement autorise déjà plus de 600 combinaisons.



qu'un seul intitulé et une seule date de début pour un code donné et vice versa. Le schéma modifié devient alors (pour les deux tables concernées).

COURS			
<u>Code</u>	Intitulé	Date début	Prix

INSCRIPTION		
<u>Nom</u>	<u>Code</u>	Payé

### Remarques

- Tout comme le modèle entité-association, le modèle relationnel aura été découvert dans une phase antérieure du cours lors de la découverte des concepts fondamentaux. Les apprenants sont ainsi déjà familiers de la représentation tabulaire et savent qu'il suffira, le moment venu, de recombinaison les tables par jointure pour retrouver les informations sous des formes plus habituelles.

- C'est à dessein que je propose ici le remplacement de certaines clés primaires "encombrantes" par des numéros-code. Il est très simple d'en démontrer l'utilité mais c'est aussi l'occasion de prouver qu'il n'y a pas lieu d'attribuer systématiquement des numéros à toutes les entités sans savoir si ce numéro permettra de simplifier la gestion de la base de données.

### 3.3. COMMENT EXPRIMER ?

Le travail de conception devient de plus en plus technique ; il s'agit à présent d'exprimer le schéma logique dans une formulation telle qu'elle soit acceptée par un logiciel SGBD spécifique (dBASE III d'Ashton Tate en l'occurrence). J'insiste sur le mot exprimer : il ne s'agit plus de modifier la structure de la base mais seulement de la traduire dans un langage déterminé. Cette étape est donc aussi essentiellement "mécanique". Pour guider l'apprenant dans son travail, il est souhaitable de distinguer ici aussi quelques phases clés.

#### a) Déclarer la structure des fichiers.

Il s'agit tout simplement de créer un fichier dBASE pour chaque table en traduisant chaque colonne par un champ du fichier. Les caractéristiques techniques des champs (type et dimension) seront déduites des "règles de fonctionnement" établies lors de l'élaboration du schéma conceptuel.

**b) Elaborer des masques de saisie en installant des mécanismes de contrôle des contraintes d'intégrité.**

Ici, dBASE ne nous offre que peu d'outils pour traduire les C.I. ; seules les clauses RANGE et PICTURE permettent un contrôle de quelques contraintes de valeurs (par ex: la date de naissance doit être inférieure à la date de jour) ou de forme (par ex: le code postal doit être composé de chiffres uniquement). Pour un contrôle vraiment efficace des C.I., il faut généralement recourir à la programmation. Notons cependant que l'énoncé même des CI constituent une excellente base (c'est le "Quoi faire ?") pour la rédaction de ces programmes de vérification.

**c) Déclaration des index "indispensables".**

Le choix des index est dicté en dBASE par deux types de considérations : d'abord la classique quête de rapidité lors des recherches les plus courantes mais aussi la nécessité de disposer d'un index lors des "mise en relation"<sup>(13)</sup> de deux fichiers.

**d) Mise en place de "vues" multifichiers pour permettre la consultation, sans (trop de) risques, à une large classe d'utilisateurs.**

Les possesseurs de dBASE III Plus sont ici plus choyés que les autres car ils disposent d'un peu plus d'outils pour créer des vues réutilisables par d'autres utilisateurs. Pour ceux qui ne disposent que de dBASE III, quelques (très) courts programmes ouvrant fichiers, index et masques et installant un filtre rendront déjà de fiers services.

**Remarques**

- Cette dernière étape de la démarche de conception est profondément influencée par les caractéristiques du logiciel qui servira à gérer la base de données. Elle devra donc être adaptée en conséquence si l'on désire employer un autre SGBD que dBASE III. Les phases de travail seront dans l'ensemble similaire. Les plus grandes différences se situant au niveau des possibilités d'implémentation des contraintes d'intégrité ainsi que de la création des vues.

---

(13) Je n'expliquerai pas ici l'inévitable confusion introduite par le mécanisme de "set relation" dans la théorie relationnelle ni sur les indispensables mises en garde quant à l'utilisation de ce mécanisme à la place de la "vraie" jointure : ce n'est pas l'objet de cet article. Il conviendra cependant de faire clairement le point sur ces sujets avec les apprenants.

- Je ne développe que fort peu la notion de vue (liée à celle de schéma externe) car elle ne trouve sa vraie justification que dans un contexte multiutilisateur. Or celui-ci n'est, aujourd'hui et avec les logiciels actuels, que fort partiellement disponible. C'est néanmoins une des directions dans laquelle la démarche présentée ici devra être étoffée d'ici peu.

- Du point de vue méthodologique, cette dernière étape doit nécessairement s'accompagner d'une mise en oeuvre réelle, sur machine de la base de données. Il est alors très intéressant de prévoir un ensemble de données à introduire de façon à vérifier, tant les mécanismes de saisies et de validation, que ceux d'extraction des données.

#### 4. CONCLUSIONS

Il reste certainement beaucoup à dire sur l'élaboration d'une méthodologie adaptée à l'enseignement du riche domaine des bases de données. Dans le seul aspect de la conception, il reste encore bien des zones d'ombres à éclairer, bien des méthodes à essayer, bien des images à trouver. J'espère malgré tout, vous avoir montré qu'ici aussi, à l'instar de la programmation, il est possible d'aborder le monde des bases de données avec une méthode structurée. Il ne faut pas cependant demander à la démarche proposée plus qu'elle ne peut apporter : nous avons, d'entrée de jeu, limité notre propos aux bases de données de taille modestes destinées à être exploitées sur un ordinateur personnel. On peut néanmoins, si l'on envisage une utilisation du SGBD en mode programmé, doubler la démarche de conception de la BD par une conception des traitements qui adoptera le même rythme en trois temps<sup>(14)</sup> : "Quoi faire ?" (spécifications), "Comment faire ?" et "Comment faire faire ?" (rédaction des algorithmes) et enfin "Comment dire ?" (traduction dans un langage adapté).

Parmi les motivations qui m'animent, figure une réaction vis-à-vis de certaines formations à dBASE III qui me rappellent furieusement les stages "Apprenez BASIC en 3 jours" d'il y a quelques années. Comme les méthodes descendante et structurée qui sont venues éclairer la démarche de programmation, je voudrais faire sentir que dans le monde des logiciels aussi, il y a lieu d'élever le débat. Connaître la syntaxe d'un langage ou les tours de mains d'un logiciel ne suffit pas. Les vrais problèmes sont ailleurs, dans la découverte et l'installation de structures de données adaptées notamment.

J'aurais pu montrer aussi qu'au moment même de l'exploitation de la BD, une démarche structurée<sup>(15)</sup> de construction des commandes d'interrogation est la meilleure garantie du succès mais c'est là une autre histoire que je réserverai pour une autre occasion.

---

(14) Voire à ce propos les nombreux écrits de Ch. DUCHATEAU et notamment [3].

(15) Construite, elle aussi, sur un canevas en trois temps : Que veut-on ? - Comment l'obtenir ? - Comment dire ?

Il est encore un peu tôt, à cette heure, pour évaluer les résultats "profonds" d'une telle présentation du monde des bases de données. Les réactions des apprenants sont en tout cas très positives et très encourageantes. J'espère, quant à moi, avoir pu partager mon enthousiasme pour la mise au point d'une méthodologie adaptée à l'apprentissage des progiciels. Si c'est la cas, il nous reste encore certainement beaucoup à découvrir ensemble !

**BIBLIOGRAPHIE**

- [1] F. BODART , Y. PIGNEUR. *Conception assistée des applications informatiques*. T 1 : Etude d'opportunité et analyse conceptuelle. Masson, Paris, 1983.
- [2] C. CHRISMENT. *Mise en oeuvre des bases de données*. Principes méthodologiques. Eyrolles, Paris, 1987.
- [3] C. DUCHATEAU. *Premiers pas en programmation*. Pour une didactique de l'algorithmique. A paraître (titre non-définitif).
- [4] GALACSI. *Comprendre les systèmes d'information*. Dunod, Paris, 1985.
- [5] GALACSI. *Les systèmes d'information. Analyse et conception*. Dunod, Paris, 1986.
- [6] J-L. HAINAUT. *Conception assistée des applications informatiques*. T 2. Conception de la base de données. Masson, Paris, 1986.
- [7] A. COLLONGUES, J. HUGUES, B. LAROCHE, MERISE. *Méthode de conception*. Dunod, Paris, 1986.
- [8] C. SALES, O. MAGNAN. *Les méthodes de conception*. Informatique Hebdo, n° 4, janvier 1988 (p 21-24).