

II. Construire un modèle de données

II.1. Phases de construction

Un *modèle de données* est une description formelle et structurée des données et de leurs liens de dépendance dans un système d'information. Pour aboutir à une base de données relationnelle décrivant le monde réel, il faut procéder en trois étapes :

- *analyse des données*. A partir d'une réalité perçue, il s'agit de spécifier les données nécessaires pour en construire un système d'information représentatif. Les utilisateurs participent activement à cette phase en exprimant leurs besoins dans leur langage. Le résultat de ces consultations est une documentation aussi complète que possible, comportant une description explicite des objectifs recherchés, ainsi qu'une spécification textuelle des données et de leurs interdépendances.
- élaboration d'un *modèle entité-association*. Partant de l'analyse des données réalisée, on identifie les objets - concrets ou abstraits - à propos desquels on veut conserver des informations, ainsi que les liens entre ceux-ci. Dans ce modèle, un objet est appelé *entité* et un lien, *association* ; les ensembles d'entités et d'associations sont, respectivement, représentés graphiquement par des rectangles et par des losanges. Le modèle entité-association fournit des outils et un cadre rigoureux pour l'analyse des données et de leurs liaisons.
- conversion du modèle entité-association en un *modèle relationnel*. Il s'agit de transformer en tables toutes les entités et les associations du modèle précédent. Chaque liaison est convertie en une table contenant les clés des ensembles d'entités y participant (appelées *clés étrangères*), en plus d'éventuels attributs de la liaison.

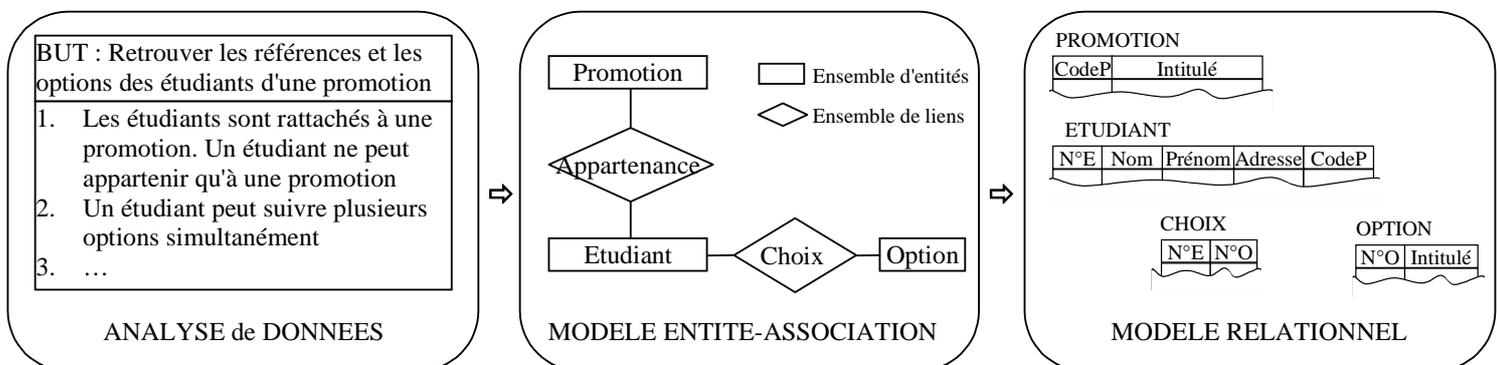


Figure 2 : Phases de construction d'une base de données relationnelle

Une fois ainsi obtenu un *schéma de base de données relationnelle*, il faut détecter et étudier les dépendances au sein des tables pour en éliminer les informations redondantes et les anomalies qui en résultent. Les concepts de *dépendance fonctionnelle*, de *forme normale* et d'*intégrité référentielle*, permettent de réaliser ces objectifs.

II.2. Le modèle entité-association

II.2.1 Définitions

- Une *entité* est un objet spécifique, concret ou abstrait, de la réalité perçue. Ce peut être une personne, un objet inerte, un concept abstrait, un événement, ...

- Un **attribut** est une caractéristique ou une qualité d'une entité ou d'une association. Il peut être atomique (ex. nom, prénom) ou composé (ex. adresse=n°+rue+code_postal+ville) et peut prendre une ou plusieurs valeur(s) (on parle d'attribut *mono-* ou *multivalué*). Le **domaine** d'un attribut est l'ensemble des valeurs que peut prendre celui-ci; il est utile pour vérifier la validité d'une donnée.
- Un **type d'entité** est la classe de toutes les entités de la réalité perçue qui sont de même nature et qui jouent le même rôle. Un type d'entité est défini par un *nom* et un ensemble d'*attributs*, qui sont les caractéristiques communes à toutes les entités de même type; ces dernières forment un *ensemble d'entités* (par exemple, un ensemble d'étudiants, caractérisés par leurs nom et prénom). Par simplification de la terminologie, on appellera *entité* un type d'entité, et *occurrence d'une entité* un individu particulier faisant partie d'une entité.
- Le **schéma** ou **intention** d'une entité en est la description ; l'ensemble des occurrences d'une entité qui existent dans la base à un instant donné s'appelle l'**extension** de l'entité. Le schéma d'une entité ne change pas fréquemment car il en décrit la structure ; son extension, en revanche, change à chaque insertion ou suppression d'une occurrence d'entité.
- L'attribut **clé** ou **identifiant** d'une entité est un groupe minimal d'attributs permettant de distinguer sans ambiguïté les occurrences d'entités dans l'ensemble considéré.
- Une association est une correspondance entre 2 ou plusieurs occurrences d'entités à propos de laquelle on veut conserver des informations. On dit que les occurrences d'entités *participent* ou jouent un **rôle** dans l'association. Un **type d'association** est défini par un *nom* et une liste d'entités avec leur rôle respectif (notation : $A(ro_1:E_1, ro_2:E_2, \dots, ro_n:E_n)$). Pour simplifier, on appelle *association* un type d'association et *occurrence d'association* toute correspondance qui existe entre 2 ou plusieurs occurrences d'entités. L'ensemble des occurrences d'une association qui existe dans la base à un instant donné s'appelle l'**extension** de l'association.

Exemple d'association : APPARTENANCE(appartient: ETUDIANT, inclut: PROMOTION) décrit le fait qu'un étudiant appartient à une promotion et, symétriquement, qu'une promotion inclut plusieurs étudiants (cf. figure 3)

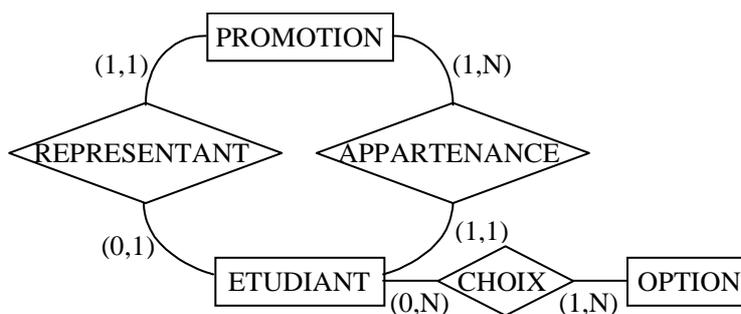


Figure 3-1 : Exemple de modèle entité-association

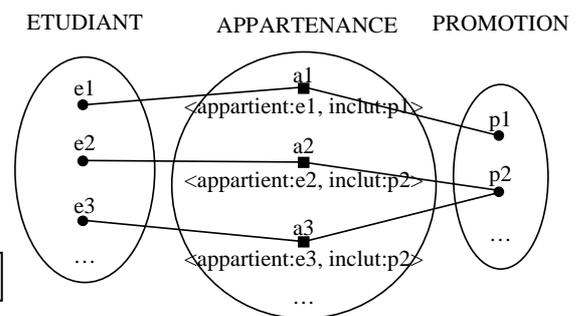


Figure 3-2 : Exemples d'occurrences de l'association APPARTENANCE

- Une association peut aussi posséder des attributs. Un attribut de l'association APPARTENANCE pourrait être, par ex., un entier indiquant le(s) semestre(s) universitaire(s) suivi(s) par l'étudiant.
- Le **degré** (ou la **dimension**) d'une association est le nombre d'entités y participant. Le cas le plus fréquent est celui de l'association binaire

II.2.2. Cardinalité d'une association

La **cardinalité** d'une association spécifie le nombre minimum et le nombre maximum de participations de chaque occurrence d'entité à chaque rôle de l'association. Autrement dit, la cardinalité de $A(ro_1:E_1,$

$ro_2:E_2, \dots, ro_n:E_n$) est définie par un ensemble de couples (min_i, max_i) où min_i (resp. max_i) indique le nombre minimum (resp. maximum) de fois que toute occurrence de E_i doit assumer le rôle ro_i . On distingue 4 cas principaux :

Type	Cardinalité	Exemple (cf. figure 3)
Simple	(1,1) (ou <i>de type 1</i>)	Chaque étudiant est inscrit dans une et une seule promotion
	(0,1) (ou <i>conditionnelle</i>)	Chaque étudiant n'est pas forcément représentant de sa promo
Complexe	(1,N) (ou <i>multiple</i>)	Chaque promotion comporte plusieurs étudiants
	(0,N) (ou <i>multiple conditionnelle</i>)	Chaque étudiant choisit un nombre quelconque d'options

La section suivante décrit des règles permettant d'obtenir un modèle relationnel à partir du modèle entité-association. Ce dernier occupe une place importante dans la modélisation de données assistée par ordinateur, mais le processus de traduction en un modèle relationnel n'est que partiellement automatisé, même dans les ateliers de génie logiciel CASE (Computer-Aided Software Engineering).

II.3. Passage au modèle relationnel

II.3.1. Notion de clé étrangère

Une **clé étrangère** d'une relation R est un sous-ensemble C des attributs de R tel que :

- il existe une relation R' (pas nécessairement distincte de R) possédant une clé candidate C' ;
- pour chaque valeur différente de C dans R, il existe une valeur (unique) de C' dans R' identique à C.

Autrement dit, une clé étrangère dans une table est un attribut ou une concaténation d'attributs qui forme une clé d'identification d'une autre table (ou éventuellement de la même table). La clé primaire d'une table comportant des clés étrangères peut être soit une concaténation de celles-ci, soit une autre clé candidate - par exemple une clé créée artificiellement.

II.3.2. Règles de passage

L'opération consiste à représenter sous forme de tables les entités et les associations obtenues précédemment. Pour cela, on applique les règles suivantes :

- Chaque entité est traduite en une table distincte, dont la clé primaire peut être soit celle de l'entité, soit une autre clé candidate. Les autres attributs de l'entité sont reportés comme attributs de la nouvelle table.
- La conversion d'une association dépend de sa cardinalité :
 - R1. Une association de dimension 2 de type simple-complexe (par exemple, (1,1)-(1,N)) ne nécessite pas la création d'une nouvelle table, mais est traduite en définissant une clé étrangère dans la table qui se situe du côté "simple" de l'association. Cette clé doit faire référence à la clé d'identification de la seconde table, et son nom est judicieusement choisi en conséquence.
 - R2. Une association de dimension 2 de type simple-simple (par exemple, (1,1)-(0,1)) se traite de la même façon, en choisissant en principe d'introduire la clé étrangère dans la table située du côté (1,1) de l'association.
 - R3. Chaque association de dimension 2 de type complexe-complexe (par exemple, (0,N)-(1,N)) est représentée par une table distincte, contenant les identifiants des deux entités associées comme clés étrangères. Ces attributs constituent souvent, à eux deux, la clé primaire de la nouvelle table. Si l'association comporte d'autres attributs, ceux-ci sont également ajoutés à la table.
 - R4. Une association de dimension supérieure à 2 se réécrit selon la règle R3.

II.3.3. Exemples

Règle	Modèle entité-association	Traduction en modèle relationnel
R1		
R2		
R3		
R4		

Figure 4 : Ex. d'applications des règles de passage modèle entité-association ⇔ modèle relationnel

II.4. Nécessité de la normalisation dans la conception d'une BDr

II.4.1. Redondance d'un attribut

Supposons que le modèle relationnel obtenu pour représenter les données relatives à l'inscription des étudiants dans une promotion se soit réduit à une seule table (suite à une mauvaise conception du modèle entité-association) :

ETUDIANT					
N°E	Nom	Prénom	Adresse	Promotion	IntituléPromo

Une telle conception entraîne des problèmes de **redondance** des informations. On s'aperçoit en effet que l'intitulé de la promotion est répété pour chaque étudiant, et qu'il est identique pour tous les étudiants inscrits dans la même promotion. Cet attribut est donc redondant, et il est préférable de stocker ces informations, relatives à une promotion, dans une table séparée.

II.4.2. Anomalies de mutation

En outre, le schéma précédent n'est pas satisfaisant car il engendre certaines anomalies lors des opérations de mise à jour, d'insertion et de suppression des données (anomalies dites *de mutation*).

- Si par exemple on modifie l'intitulé d'une promotion dans un tuple (ex. de "Sciences de la Vie" en "Sciences du Vivant"), ce même intitulé devra être changé pour tous les autres étudiants concernés, sans quoi cette promotion possédera plusieurs intitulés différents (*anomalie de mise à jour*).
- Si une nouvelle promotion se crée, elle ne pourra pas être ajoutée dans la table sans y inscrire immédiatement des étudiants, car la clé primaire (N°E) ne peut être laissée vide. Cette *anomalie d'insertion* peut être gênante, par exemple, parce que le stockage d'informations sur la nouvelle formation (maquette, intervenants, ...) doit pouvoir intervenir avant toute inscription d'étudiant.
- Si l'on supprime tous les étudiants de la table (par exemple parce qu'ils ont tous eu leur diplôme !), on perd du même coup toutes les informations relatives aux promotions (*anomalie de suppression*).

Une base de données relationnelle doit stocker un ensemble de schémas relationnels sans redondance inutile et sans comportement anormal des relations lors des opérations de mise à jour. C'est pourquoi la théorie de la normalisation des relations a été associée au modèle relationnel.

II.5. Dépendances fonctionnelles et normalisation des relations

II.5.1. Dépendance fonctionnelle : définitions

Introduite par Codd en 1970, la notion de dépendance fonctionnelle (DF) permet de caractériser des relations qui peuvent être décomposée sans perte d'information. Sa formalisation mathématique découle de l'observation des rôles de cardinalité simple (au plus égale à 1). Intuitivement, une DF traduit le fait que les valeurs de certains attributs sont nécessairement déterminées (fixées) lorsque le sont celles d'autres attributs.

On dit qu'un attribut ou un ensemble d'attributs Y est *fonctionnellement dépendant* d'un autre (ensemble d') attribut(s) X si, à chaque valeur prise par X correspond une valeur unique de Y . Autrement dit, des valeurs identiques de X impliquent des valeurs identiques de Y . Cette dépendance est notée $X \rightarrow Y$.

Une propriété remarquable des clés d'identification est que, justement, tout attribut non-clé dépend fonctionnellement de la clé, puisque celle-ci identifie chaque tuple de manière unique.

Une des propriétés de la DF est la transitivité : $X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$. On dit qu'un attribut Z est *transitivement dépendant* d'un attribut X s'il existe un attribut Y tel que $X \rightarrow Y$ et $Y \rightarrow Z$, et aussi que X ne dépend pas fonctionnellement de Y .

II.5.2. Normalisation des relations

Afin d'éliminer les divers types de dépendances, et donc éviter les anomalies de mutation, on applique un processus de normalisation sur les différentes relations de la base. Les étapes successives de ce processus imposent des critères de plus en plus restrictifs aux tables normalisées :

- Une relation est en *première forme normale* (notée 1NF) si les domaines de tous ses attributs sont des valeurs atomiques (et non des ensembles, types énumérés ou listes). Tout attribut d'une relation 1NF doit donc être monovalué.

La 1NF permet d'éliminer les domaines composés.

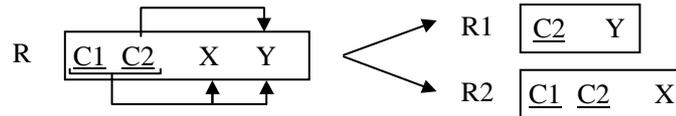
Pour normaliser une table à la 1NF, il suffit de créer un tuple distinct pour chaque valeur de l'attribut multivalué. Cela introduit des redondances, mais la deuxième forme normale permet d'y remédier.

- Une relation est en deuxième forme normale (2NF) si elle est 1NF et si, de plus, tout attribut non-clé dépend fonctionnellement de **toute** la clé (et pas seulement d'une partie de celle-ci). Pour qu'une table soit 2NF, il faut donc, si sa clé est *composée* (sous-entendu, de plusieurs attributs), que tout autre attribut soit fonctionnellement dépendant de la clé entière (on dit parfois qu'il y a *dépendance fonctionnelle totale*).

La 2NF garantit qu'aucun attribut n'est déterminé seulement par une partie de la clé

Pour normaliser à la 2NF une table possédant une clé composée, il faut décomposer celle-ci en :

- une table formée des attributs dépendants d'une partie de la clé, et de cette partie même.
- une seconde table formée de la clé composée et des attributs restants :

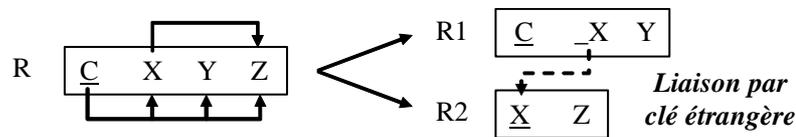


- Une relation est en troisième forme normale (3NF) si elle est 2NF et si, de plus, tout attribut non-clé ne dépend pas transitivement de la clé (ou, autrement dit, si tout attribut non-clé ne dépend pas fonctionnellement d'un attribut non-clé).

La 3NF permet d'éliminer les redondances dues aux dépendances transitives.

Pour normaliser à la 3NF une table possédant une dépendance transitive, il faut décomposer celle-ci en :

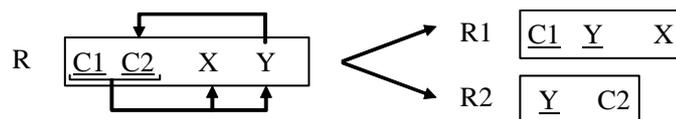
- une table formée de l'attribut redondant et de l'attribut dont il dépend (nommé ici X); ce dernier devient la clé de la nouvelle table
- une seconde table formée de la clé, de l'attribut X comme clé étrangère, et des autres attributs :



Les formes normales 2NF et 3NF assurent l'élimination des redondances parmi les attributs non-clé, mais pas les redondances potentielles au sein d'attributs formant une clé composite. C'est pourquoi Boyce et Codd ont proposé une extension de la 3NF.

- Une relation est en forme normale de Boyce-Codd (BCNF) si, et seulement si, les seules dépendances fonctionnelles élémentaires¹ sont celles dans lesquelles une clé détermine un attribut non-clé.

Plus simple que la 3NF, un peu plus restrictive, cette forme est utile lorsqu'une table possède plusieurs clés candidates. Plus précisément, lorsque les clés se chevauchent dans une table, celle-ci risque de transgresser la forme normale de Boyce-Codd, même si elle est en 3NF. Il faut alors la décomposer d'après les clés candidates :



Toute relation admet au moins une décomposition en BCNF qui est sans perte ; cependant, une telle décomposition ne préserve généralement pas les dépendances fonctionnelles.

¹ Une *dépendance fonctionnelle élémentaire* est de la forme $X \rightarrow A$, où A est un attribut unique n'appartenant pas à X et où il n'existe pas X' inclus dans X tel que $X' \rightarrow A$.

Remarque :

Il existe deux autres formes normales, encore plus restrictives, mais qui se rencontrent bien moins fréquemment. La quatrième (4NF), par exemple, est basée sur les dépendances multivaluées¹.

II.5.3. Exemple

Dans la pratique, les critères des formes normales sont satisfaits si le modèle entité-association est bien construit et si les règles de passage au modèle relationnel ont été correctement appliquées. La vérification de toutes les formes normales successives n'a plus alors besoin d'être systématique.

Supposons au contraire que le modèle entité-association ait fourni la relation suivante (figure 5-1). A l'évidence, celle-ci n'est pas en 1NF ; il faut donc la transformer conformément à la figure 5-2. La clé de la nouvelle table, composée des attributs *N°E* et *Option*, détermine bien de manière unique les attributs non-clé (c'est-à-dire qu'on a les dépendances fonctionnelles $(N^{\circ}E, Option) \rightarrow Nom$, $(N^{\circ}E, Option) \rightarrow Prénom$, et $(N^{\circ}E, Option) \rightarrow Adresse$). Cependant, la transformation a manifestement créé des redondances et, intuitivement, on sait que l'adresse de l'étudiant n'a aucun rapport avec l'option choisie. Cela se traduit par le fait que les attributs non-clé sont fonctionnellement dépendants d'une partie de la clé ($N^{\circ}E \rightarrow Nom$, $N^{\circ}E \rightarrow Prénom$, et $N^{\circ}E \rightarrow Adresse$). La deuxième forme normale exige donc de créer deux tables séparées (figure 5-3).

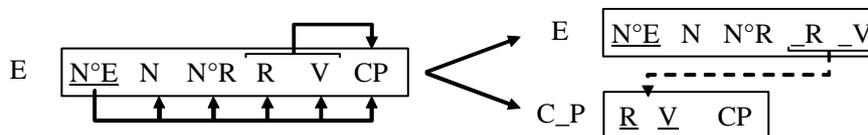
<p>ETUDIANT</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Nom</th> <th>Adresse</th> <th>Options</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>Dulac</td> <td>2, allée d</td> <td>{O2,O3,O4}</td> </tr> <tr> <td>0002</td> <td>Abbou</td> <td>35, rue</td> <td>{O1,O3}</td> </tr> <tr> <td>0003</td> <td>C</td> <td>...</td> <td>...</td> </tr> </tbody> </table>	N°E	Nom	Adresse	Options	0001	Dulac	2, allée d	{O2,O3,O4}	0002	Abbou	35, rue	{O1,O3}	0003	C	<p>ETUDIANT</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Option</th> <th>Nom</th> <th>Adresse</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>O2</td> <td>Dulac</td> <td>2, allée d</td> </tr> <tr> <td>0001</td> <td>O3</td> <td>Dulac</td> <td>2, allée d</td> </tr> <tr> <td>0001</td> <td>O4</td> <td>Dulac</td> <td>2, allée d</td> </tr> <tr> <td>0002</td> <td>O1</td> <td>Abbou</td> <td>35, rue</td> </tr> <tr> <td>0002</td> <td>O3</td> <td>Abbou</td> <td>35, rue</td> </tr> <tr> <td>0002</td> <td>O3</td> <td>Abbou</td> <td>35, rue</td> </tr> </tbody> </table>	N°E	Option	Nom	Adresse	0001	O2	Dulac	2, allée d	0001	O3	Dulac	2, allée d	0001	O4	Dulac	2, allée d	0002	O1	Abbou	35, rue	0002	O3	Abbou	35, rue	0002	O3	Abbou	35, rue	<p>ETUDIANT CHOIX</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Nom</th> <th>Adresse</th> <th>N°E</th> <th>Option</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>Dulac</td> <td>2, allée d</td> <td>0001</td> <td>O2</td> </tr> <tr> <td>0002</td> <td>Abbou</td> <td>35, rue</td> <td>0001</td> <td>O3</td> </tr> <tr> <td>0002</td> <td>Abbou</td> <td>35, rue</td> <td>0001</td> <td>O4</td> </tr> <tr> <td>0002</td> <td>Abbou</td> <td>35, rue</td> <td>0002</td> <td>O1</td> </tr> <tr> <td>0002</td> <td>Abbou</td> <td>35, rue</td> <td>0002</td> <td>O3</td> </tr> </tbody> </table>	N°E	Nom	Adresse	N°E	Option	0001	Dulac	2, allée d	0001	O2	0002	Abbou	35, rue	0001	O3	0002	Abbou	35, rue	0001	O4	0002	Abbou	35, rue	0002	O1	0002	Abbou	35, rue	0002	O3
N°E	Nom	Adresse	Options																																																																									
0001	Dulac	2, allée d	{O2,O3,O4}																																																																									
0002	Abbou	35, rue	{O1,O3}																																																																									
0003	C																																																																									
N°E	Option	Nom	Adresse																																																																									
0001	O2	Dulac	2, allée d																																																																									
0001	O3	Dulac	2, allée d																																																																									
0001	O4	Dulac	2, allée d																																																																									
0002	O1	Abbou	35, rue																																																																									
0002	O3	Abbou	35, rue																																																																									
0002	O3	Abbou	35, rue																																																																									
N°E	Nom	Adresse	N°E	Option																																																																								
0001	Dulac	2, allée d	0001	O2																																																																								
0002	Abbou	35, rue	0001	O3																																																																								
0002	Abbou	35, rue	0001	O4																																																																								
0002	Abbou	35, rue	0002	O1																																																																								
0002	Abbou	35, rue	0002	O3																																																																								

Fig. 5-1: Forme non normalisée

Figure 5-2: Forme 1NF

Figure 5-3 : Forme 2NF

Supposons qu'en plus des attributs précédents, on ait stocké dans la table ETUDIANT le code postal et la ville, et qu'on ait séparé le numéro de voie et la rue. La 2NF correspondante est représentée figure 6-1 (la table CHOIX n'a pas été représentée). Dans cette table, l'attribut *CodePostal* dépend fonctionnellement de l'attribut composé $(Rue, Ville)^2$ et, par là même, dépend transitivement de la clé. Pour passer en 3NF, il faut donc créer une table CODE_POSTAL distincte, et faire apparaître dans ETUDIANT l'attribut codant la rue et la ville sous forme de clé étrangère (figure 6-2) :

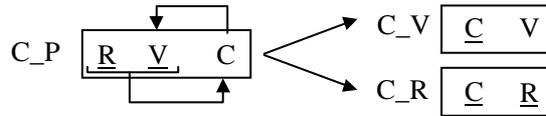


Dans la nouvelle table, on s'aperçoit toutefois qu'il y a encore de la redondance. En effet, si la clé composée de la rue et de la ville détermine bien le code postal, la ville est aussi fonctionnellement

¹ On dit qu'il existe une *dépendance multivaluée* d'un attribut X vers un attribut Z (notée $X \twoheadrightarrow Z$) si toute combinaison d'une valeur donnée de X avec celle d'un autre attribut Y détermine un ensemble identique de Z.

² Il faut entendre "Ville" ici au sens de "Bureau distributeur". S'il peut en effet y avoir plusieurs codes postaux pour un même bureau distributeur (ex. 59000 et 59800 pour Lille), un bureau distributeur peut regrouper plusieurs petites "villes" rurales. On a supposé de surcroît que $(Rue, Ville) \rightarrow CodePostal$, c'est-à-dire que la connaissance d'un nom de rue et de la ville détermine sans ambiguïté le code postal. En conséquence, l'attribut composé $(Rue, Ville)$ est choisi comme clé, ce qui implique qu'il n'y a qu'une seule rue portant un nom donné dans une ville donnée, même pour des secteurs de codes postaux différents.

dépendante du code postal (c'est-à-dire (Rue,Ville)→CodePostal et CodePostal→Ville). Ceci est lié au fait que l'attribut (Rue, CodePostal) constitue également une clé candidate (*clé secondaire*). Pour obtenir une forme normale de Boyce-Codd, il faut donc décomposer cette relation en deux (figure 6-3) :



<p>ETUDIANT</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Nom</th> <th>Nrue</th> <th>Rue</th> <th>Ville</th> <th>CP</th> </tr> </thead> <tbody> <tr><td>0001</td><td>Dulac</td><td>2</td><td>allée du chêne</td><td>Lille</td><td>59000</td></tr> <tr><td>0002</td><td>Abbou</td><td>35</td><td>rue V. Hugo</td><td>Lille</td><td>59800</td></tr> <tr><td>0003</td><td>Caron</td><td>3</td><td>place Allende</td><td>Calais</td><td>62000</td></tr> <tr><td>0004</td><td>Fabre</td><td>428</td><td>rue V. Hugo</td><td>Calais</td><td>62000</td></tr> </tbody> </table>	N°E	Nom	Nrue	Rue	Ville	CP	0001	Dulac	2	allée du chêne	Lille	59000	0002	Abbou	35	rue V. Hugo	Lille	59800	0003	Caron	3	place Allende	Calais	62000	0004	Fabre	428	rue V. Hugo	Calais	62000	<p>ETUDIANT</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Nom</th> <th>Nrue</th> <th>Rue</th> <th>Ville</th> </tr> </thead> <tbody> <tr><td>0001</td><td>Dulac</td><td>2</td><td>allée du chêne</td><td>Lille</td></tr> <tr><td>0002</td><td>Abbou</td><td>35</td><td>rue V. Hugo</td><td>Lille</td></tr> <tr><td>0003</td><td>Caron</td><td>3</td><td>place Allende</td><td>Calais</td></tr> <tr><td>0004</td><td>Fabre</td><td>428</td><td>rue V. Hugo</td><td>Calais</td></tr> </tbody> </table> <p>CODE_POSTAL</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Rue</th> <th>Ville</th> <th>CP</th> </tr> </thead> <tbody> <tr><td>allée du chêne</td><td>Lille</td><td>59000</td></tr> <tr><td>rue V. Hugo</td><td>Lille</td><td>59800</td></tr> <tr><td>place Allende</td><td>Calais</td><td>62000</td></tr> <tr><td>rue V. Hugo</td><td>Calais</td><td>62000</td></tr> </tbody> </table>	N°E	Nom	Nrue	Rue	Ville	0001	Dulac	2	allée du chêne	Lille	0002	Abbou	35	rue V. Hugo	Lille	0003	Caron	3	place Allende	Calais	0004	Fabre	428	rue V. Hugo	Calais	Rue	Ville	CP	allée du chêne	Lille	59000	rue V. Hugo	Lille	59800	place Allende	Calais	62000	rue V. Hugo	Calais	62000	<p>ETUDIANT</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>N°E</th> <th>Nom</th> <th>Nrue</th> <th>Rue</th> <th>Ville</th> </tr> </thead> <tbody> <tr><td>0001</td><td>Dulac</td><td>2</td><td>allée du chêne</td><td>Lille</td></tr> <tr><td>0002</td><td>Abbou</td><td>35</td><td>rue V. Hugo</td><td>Lille</td></tr> <tr><td>0003</td><td>Caron</td><td>3</td><td>place Allende</td><td>Calais</td></tr> <tr><td>0004</td><td>Fabre</td><td>428</td><td>rue V. Hugo</td><td>Calais</td></tr> </tbody> </table> <p>CODE_VILLE CODE_RUE</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CP</th> <th>Ville</th> <th>CP</th> <th>Rue</th> </tr> </thead> <tbody> <tr><td>59000</td><td>Lille</td><td>59000</td><td>allée du chêne</td></tr> <tr><td>59800</td><td>Lille</td><td>59800</td><td>rue V. Hugo</td></tr> <tr><td>62000</td><td>Calais</td><td>62000</td><td>place Allende</td></tr> <tr><td>62000</td><td>Calais</td><td>62000</td><td>rue V. Hugo</td></tr> </tbody> </table>	N°E	Nom	Nrue	Rue	Ville	0001	Dulac	2	allée du chêne	Lille	0002	Abbou	35	rue V. Hugo	Lille	0003	Caron	3	place Allende	Calais	0004	Fabre	428	rue V. Hugo	Calais	CP	Ville	CP	Rue	59000	Lille	59000	allée du chêne	59800	Lille	59800	rue V. Hugo	62000	Calais	62000	place Allende	62000	Calais	62000	rue V. Hugo
N°E	Nom	Nrue	Rue	Ville	CP																																																																																																																
0001	Dulac	2	allée du chêne	Lille	59000																																																																																																																
0002	Abbou	35	rue V. Hugo	Lille	59800																																																																																																																
0003	Caron	3	place Allende	Calais	62000																																																																																																																
0004	Fabre	428	rue V. Hugo	Calais	62000																																																																																																																
N°E	Nom	Nrue	Rue	Ville																																																																																																																	
0001	Dulac	2	allée du chêne	Lille																																																																																																																	
0002	Abbou	35	rue V. Hugo	Lille																																																																																																																	
0003	Caron	3	place Allende	Calais																																																																																																																	
0004	Fabre	428	rue V. Hugo	Calais																																																																																																																	
Rue	Ville	CP																																																																																																																			
allée du chêne	Lille	59000																																																																																																																			
rue V. Hugo	Lille	59800																																																																																																																			
place Allende	Calais	62000																																																																																																																			
rue V. Hugo	Calais	62000																																																																																																																			
N°E	Nom	Nrue	Rue	Ville																																																																																																																	
0001	Dulac	2	allée du chêne	Lille																																																																																																																	
0002	Abbou	35	rue V. Hugo	Lille																																																																																																																	
0003	Caron	3	place Allende	Calais																																																																																																																	
0004	Fabre	428	rue V. Hugo	Calais																																																																																																																	
CP	Ville	CP	Rue																																																																																																																		
59000	Lille	59000	allée du chêne																																																																																																																		
59800	Lille	59800	rue V. Hugo																																																																																																																		
62000	Calais	62000	place Allende																																																																																																																		
62000	Calais	62000	rue V. Hugo																																																																																																																		

Figure 6-1: Forme 2NF

Figure 6-2: Forme 3NF

Figure 6-3 : Forme BCNF

II.6. Intégrité structurelle

II.6.1. Définition et typologie

Une **contrainte d'intégrité** (CI) est une propriété ou une règle que doivent satisfaire les données de la base pour être considérées comme correctes (sans ambiguïtés ni incohérences). Une base de données est dite **intègre** ou **cohérente** si ses contraintes d'intégrité sont satisfaites. En pratique, ces contraintes ont pour effet de limiter les occurrences possibles des structures d'informations. La conception d'une base de données relationnelle se compose du schéma relationnel même et d'un ensemble de CIs.

De façon générale, on distingue :

- les contraintes d'intégrité *statiques* sont des propriétés qui doivent être vérifiées à tout moment :
 - les contraintes *individuelles* imposent un type de donnée (ex. entier long), un ensemble de valeurs (par exemple, AnnéeEmbauche={1980..2002}, AnnéeEmbauche={1980, 1984, 1995}, AnnéeEmbauche>AnnéeNaissance) ou une valeur obligatoire (ex. "l'attribut *pilote* doit être obligatoirement renseigné")
 - les contraintes *intra-relation* portent sur les valeurs des attributs : unicité de valeur, cardinalité, dépendances entre les valeurs d'attributs différents (ex. "les espèces e3 et e6 ne peuvent cohabiter")
 - les contraintes *inter-relations* sont des dépendances référentielles ou existentielles (ex. "tout vol doit avoir comme commandant de bord une personne référencée dans la table des pilotes")
- les contraintes d'intégrité *dynamiques* sont des propriétés que doit respecter tout changement d'état de la base de données; elles en définissent les séquences possibles de changement d'état. (ex. "le salaire d'un employé ne peut qu'augmenter")

Trois catégories de contraintes d'intégrité structurelles sont particulièrement importantes dans le modèle relationnel : les contraintes d'**unicité** (chaque table possède une clé d'identification qui en identifie les tuples de manière unique), les contraintes de **domaine** (les valeurs possibles d'un attribut sont restreintes à un ensemble de valeurs prédéfinies) et les contraintes d'**intégrité référentielle** (chaque valeur d'une clé étrangère doit exister comme valeur de la clé d'identification dans la table référencée).

II.6.2. Intégrité référentielle

Comme il a déjà été dit, le mécanisme d'intégrité référentielle impose que chaque valeur d'une clé étrangère existe comme valeur de la clé d'identification dans la table référencée¹. Tout SGBDR digne de ce nom supporte ce concept fondamental ; prenons un exemple pour l'illustrer. Supposons que la normalisation en 3NF de la table ETUDIANT, dans laquelle figure la promotion d'inscription, ait fourni la table de la figure 7-2. On constate qu'aucune valeur de la clé étrangère `_Promo` ne viole l'intégrité référentielle. Mais l'insertion d'un tuple tel que "0005, Mader, DESSDC" serait rejetée si la clé primaire de la table PROMOTION ne contient pas la valeur "DESSDC".

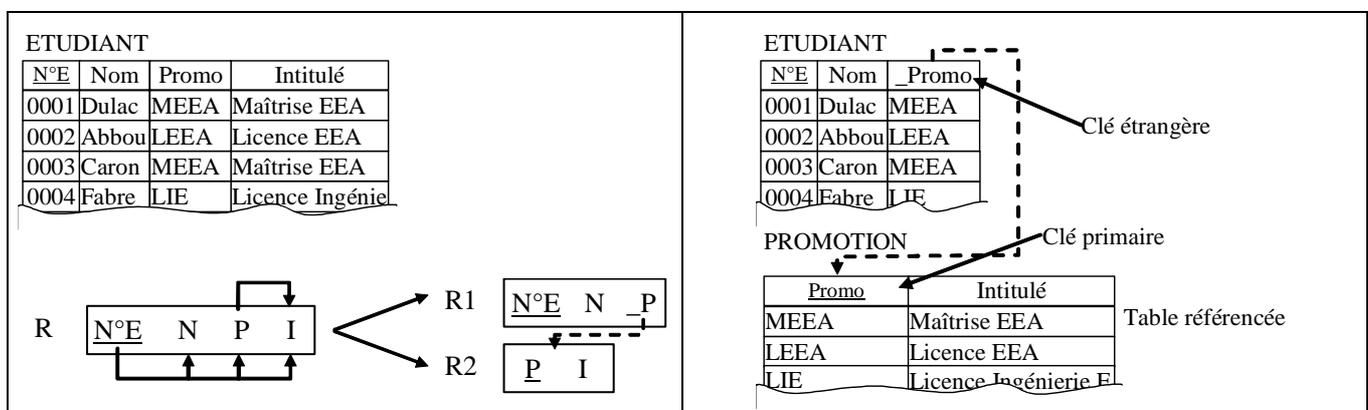


Figure 7-1: Table ETUDIANT en 2NF

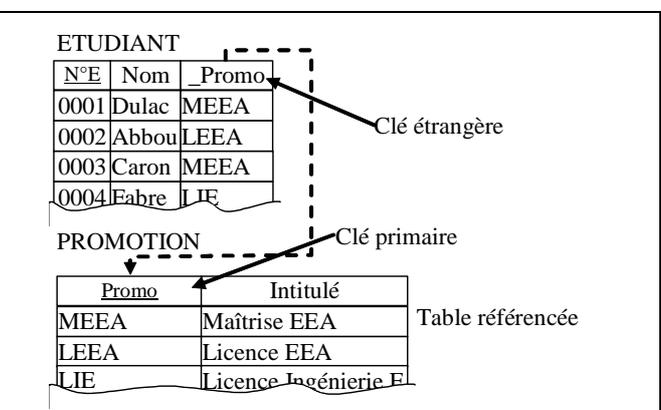


Figure 7-2: Forme 3NF et Intégrité référentielle

Le respect de l'intégrité référentielle implique certaines actions. On a vu que le SGBDR vérifiait l'existence d'une valeur de clé étrangère lors de l'insertion d'un nouveau tuple; mais que se passe-t-il si l'on supprime ou si l'on modifie une valeur de la clé primaire dont dépendent d'autres valeurs dans une clé étrangère ? Plusieurs stratégies sont possibles :

- la *suppression* (ou la mise à jour) *interdite* : l'opération est tout simplement refusée (ex. la suppression de la ligne "MEEA, Maîtrise EEA" est impossible car des tuples d'ETUDIANT y font référence)
- la *suppression* (ou la mise à jour) *en cascade* : dans ce mode, lorsqu'un tuple de la table référencée est supprimé (ou, respectivement, modifié dans ses attributs clés), tous les tuples dépendants sont aussi supprimés (resp. modifiés). Par exemple, la suppression du tuple "MEEA, Maîtrise EEA" entraîne celle des étudiants portant les numéros 0001 et 0003 ; le changement de clé "MEEA" en "MEA" affecte de la même façon les valeurs de la clé étrangère pour ces étudiants.
- la *suppression avec initialisation* : toutes les clés étrangères égales à la clé primaire supprimée sont mises à la valeur nulle (ex. la suppression du tuple "MEEA, Maîtrise EEA" entraîne la mise à NULL de l'attribut `_Promo` des étudiants portant les numéros 0001 et 0003).

¹ Dans les SGBDR, la liaison entre les tables assurée par le mécanisme de clé étrangère se fait *par valeur*, (comme dans les systèmes de gestion de fichiers); dans les systèmes orientés objets au contraire, cette liaison est réalisée *par adresse*.