Chapitre 3
Le modèle relationnel

1. Généralités

Il a été introduit par Codd en 1970.
Par rapport aux autres modèles il ajoute la rigueur des concepts s'appuyant sur

- la logique des prédicats
- les schémas de données, sous forme de tables, faciles à utiliser
- un langage de haut niveau non procédural (on n'a pas besoin de préciser comment obtenir l'information)
- une bonne indépendance entre le modèle physique et le modèle logique : (le modèle ne comporte aucune description physique de l'organisation des données)

On appelle Domaine, noté D_i, un ensemble, énuméré ((bleu, blanc, rouge), marques de voitures, ...) ou non (N, nombres réels, intervalles numériques, etc..)
Une relation est une partie (un sous ensemble) d'un produit cartésien de domaines $D_1 \times D_2 \times ... \times D_n$
Il définit un ensemble fini d'éléments (on dit aussi tuples ou nuplets) appartenant à la relation.

- Une relation est représentée par une table : une ligne représente un tuple,
- une colonne représente les composants appartenant à un Domaine donné déterminé par un attribut. À chaque attribut correspond un domaine : Att(Domaine)

Attributs et domaines constituent un schéma de relation. R, noté R(A1,A2, ... An).

Une relation r est une instance de R c'est-à-dire une table contenant des tuples (provenant du monde réel en général).

Exemple : relation livre(numéro,auteur,titre,editeur).

Livre	numéro	Auteur	Titre	Editeur
tuple1	2004	" François le Berre"	" les bases de données bretonnes"	"les éditions du far"
tuple2	1515	" François 1er"	" la bataille de Marignan"	" historia"
tuple3	12 693	" gilles desrocques"	" les bases du paramoteur"	" cavole"
tuple4	309	" gilles desrocques"	" introduction a l'ULM"	" cavole"

Remarque : l'ordre des colonnes, comme celui des lignes n'a aucune importance.

Un schéma de base de données relationnelle B est un ensemble de schémas de relations Ri.
Une base de donnée relationnelle b est une instance de B (c'est donc un ensemble de relations r1, r2, ... rn).

Exemple : schéma de la base de données relationnelle bibliothèque :
Personne(numInscription, nom, prénom, adresse)
Livre(numero, auteur, titre, éditeur)
Auteur(nom, pr énom, adresse)
Editeur(nom, adresse)
empruntéPar(numInscription, numéro, date)
écritPar(nom, numéro)
éditéPar(nom, numéro).

Remarque : en cas d'ambiguïtés, on procède à un renommage : aPourPere(personne, personne) sera transformé en aPourPere(ascendant, descendant)

Il est clair qu'il vaut mieux bien choisir les noms des éléments afin d'éviter les confusions.

La notion de clé, définie dans le modèle entités association est reprise ici et s'applique également aux associations.
Lorsqu'il y a plusieurs clés possibles, on choisit l'une d'entre elles, appelé clé primaire. Les autres clés s'appellent clés candidates (envisageables).

2. Passage du MCD entité-association au modèle relationnel

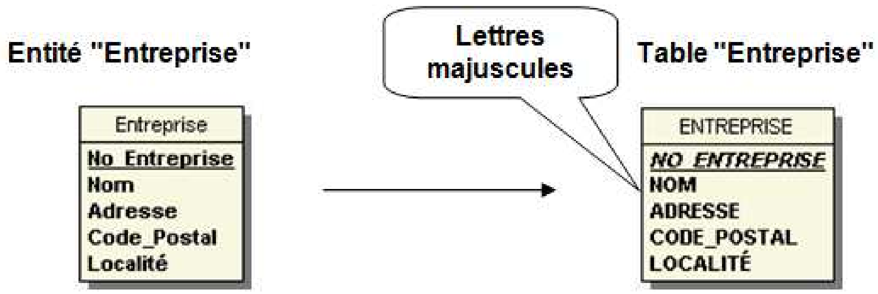
Un schéma entité- association peut être traduit mécaniquement en un schéma relationnel. (Certains logiciels, WinDesign, BD_Designer ... le font automatiquement). Nous allons voir ici comment pratiquer "à la main" pour passer du Modèle Conceptuel des Données entité-association au modèle relationnel.

Nous allons définir les règles de transformation pour le passage du MCD au MLD, en respectant les différents cas qui se posent.

i. Transformation des entités

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

Exemple:



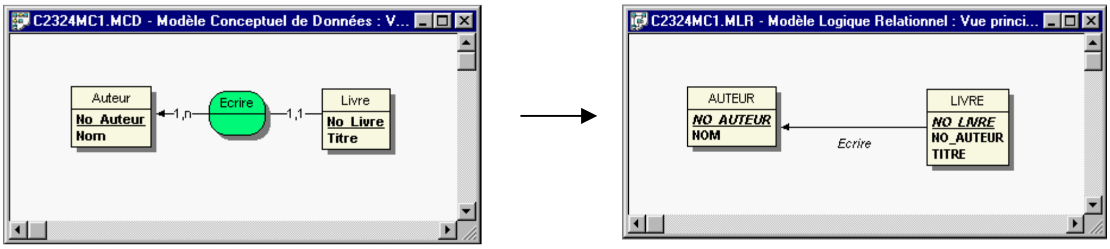
ii. Transformation des relations binaires du type[1] (x,n) – (x,1)

Afin de représenter la relation, on duplique la clé primaire de la table basée sur l'entité à cardinalité (x,n) dans la table basée sur l'entité à cardinalité (x,1). Cet attribut est appelé clé

1 x peut prendre les valeurs 0 ou 1

étrangère. Les deux tables sont liées par une flèche nommée selon la relation, qui pointe de la table à clé étrangère vers la table qui contient la clé primaire correspondante.

Exemple:



L'attribut No_Auteur qui est clé primaire de la table Auteur, devient clé étrangère dans la table Livre.

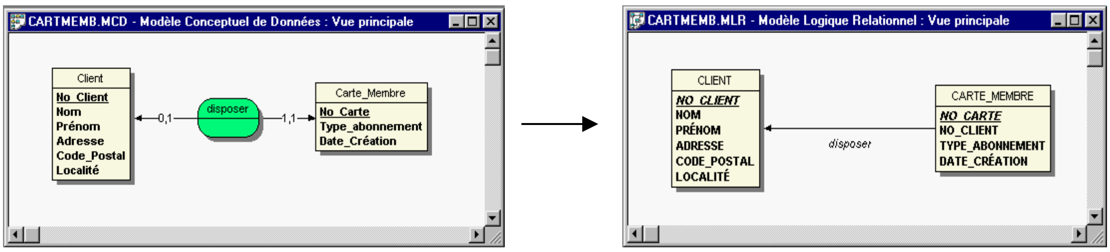
iii. Transformation des relations binaires du type (x,1) – (x,1)

Nous devons distinguer plusieurs cas. Sachant qu'une relation binaire du type (1,1)-(1,1) ne doit pas exister il nous reste les 2 cas suivants:

Relation binaire (0,1)-(1,1)

On duplique la clé de la table basée sur l'entité à cardinalité (0,1) dans la table basée sur l'entité à cardinalité (1,1).

Exemple:

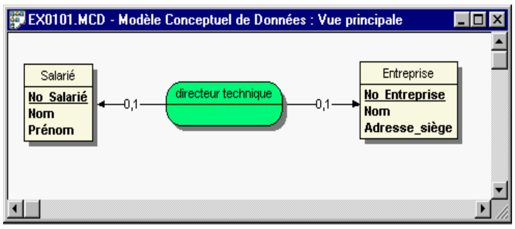


Le No_Client, qui est clé primaire de la table Client, devient clé étrangère dans la table Carte_Membre.

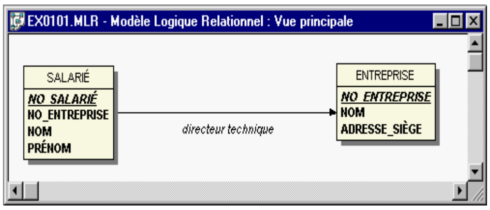
Relation binaire (0,1)-(0,1)

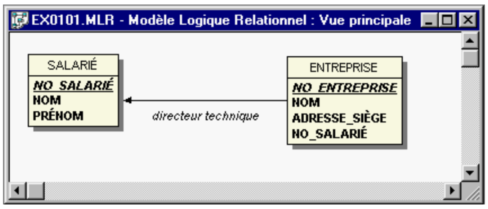
On duplique la clé d'une des tables dans l'autre. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent également attributs de la table dans laquelle a été ajoutée la clé étrangère.

Exemple:



ou



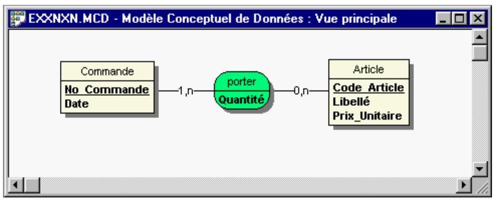


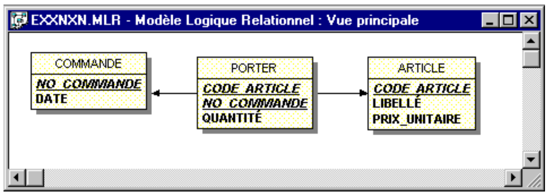
Soit on migre la clé primaire de la table Entreprise dans la table Salarié, soit on fait l'inverse.

iv. Transformation des relations binaires du type (x,n) – (x,n)

On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires des 2 tables. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.

Exemple:



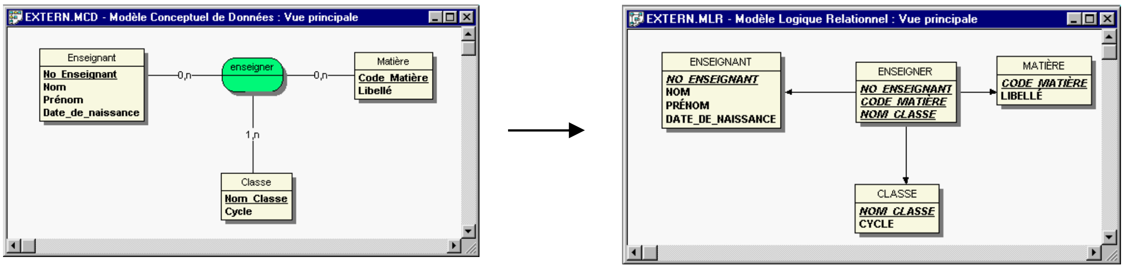


On crée une table Porter, qui contient comme clé primaire une clé composée de No-Commande et Code_Article. Elle contient également la propriété Quantité issue de la relation Porter.

v. Transformation des relations ternaires

On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires de toutes les tables reliées. Cette règle s'applique de façon indépendante des différentes cardinalités. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.

Exemple:

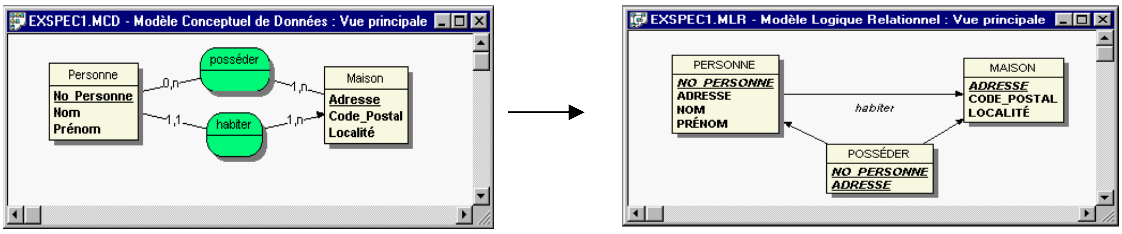


La table Enseigner contient une clé composée de No_Enseignant, Code_Matière et Nom_Classe.

vi. Transformation de plusieurs relations entre 2 entités

⚠ Les règles générales s'appliquent

Exemple:

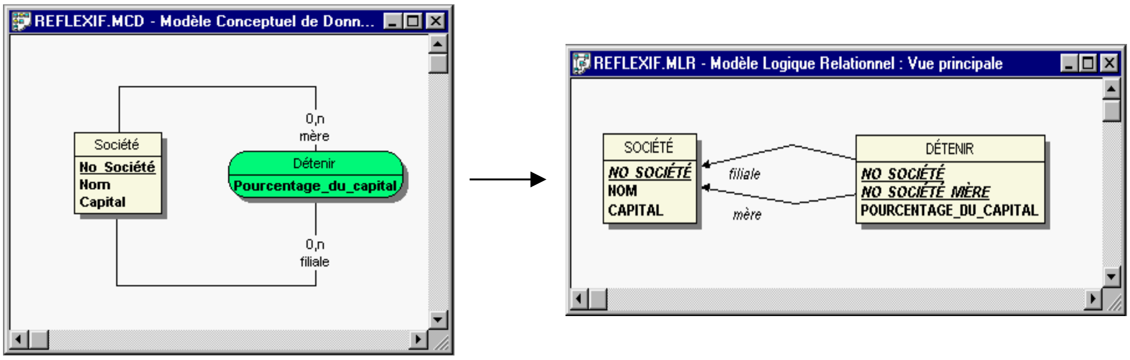


La relation habiter du type (x,n)-(x,1), est traduite par la migration de l'attribut Adresse dans la table Personne. La relation posséder du type (x,n)-(x,n) est traduite par la création d'une table supplémentaire du même nom. Cette table contient comme clé primaire composée, les clés des deux tables reliées Personne et Maison. On a donc simplement appliqué 2 fois de façon indépendante les règles de transfert MCD / MLD.

vii. Transformation des relations réflexives

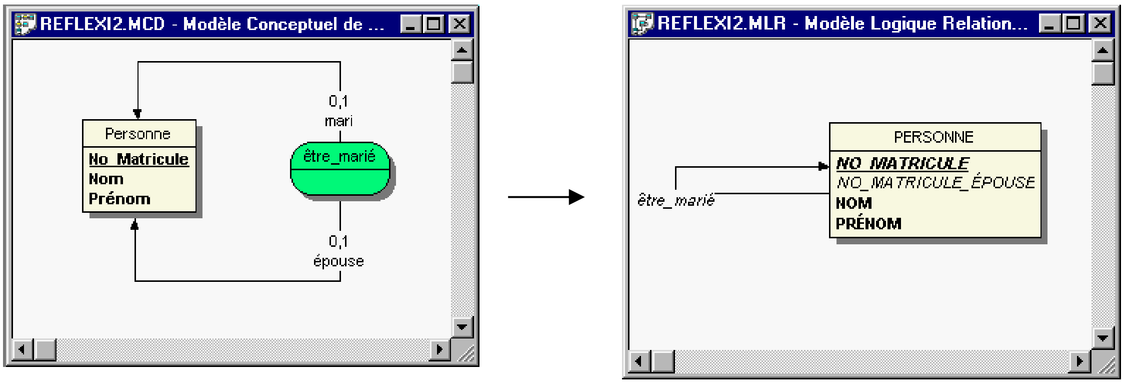
⚠ Nous appliquons les règles générales avec la seule différence que la relation est 2 fois reliée à la même entité

Exemple 1:



Comme il s'agit d'une relation (x,n)-(x,n), une table supplémentaire est créée. Cette table contient comme clé primaire composée, la clé des "deux" entités reliées. Comme la même entité est liée 2 fois à la relation, on ne peut pas utiliser 2 fois le même nom pour la clé. Dans ce cas il convient d'utiliser des rôles dans le MCD, et d'intégrer le rôle dans le nom d'une des clés migrées dans le MLD.

Exemple 2:

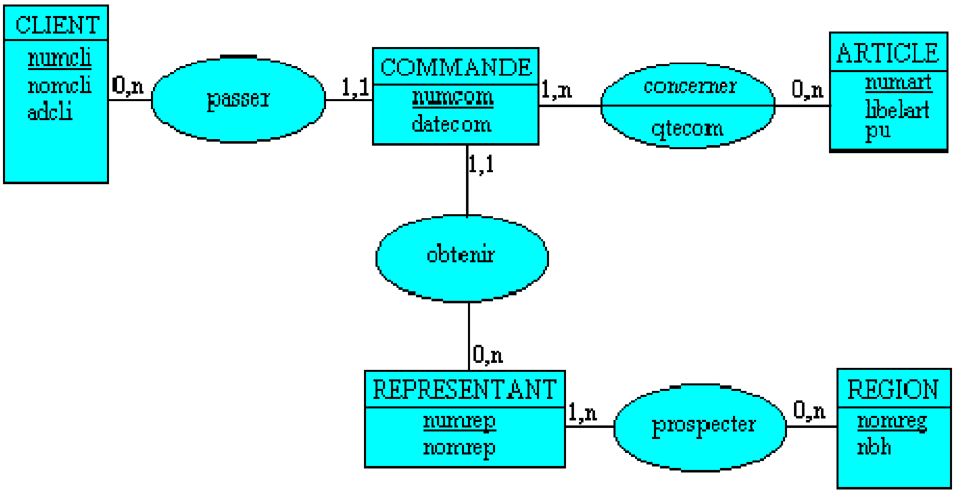


Comme il s'agit d'une relation (0,1)-(0,1), nous avons en général le choix en ce qui concerne quelle entité contiendra la clé étrangère. Comme cette relation est liée deux fois à la même entité, il est évident que nous devons dupliquer la clé primaire, tout en veillant que le même nom de clé ne sera pas utilisé pour la clé primaire et la clé étrangère. Dans notre exemple, tous les hommes mariés, ont comme valeur de la clé étrangère la matricule de leur épouse actuelle. Pour les hommes non mariés et les femmes, la clé étrangère est sans valeur. On pourrait bien sûr utiliser la modélisation inverse avec une clé étrangère NO_MATRICULE_MARI, qui indique pour chaque femme mariée, la matricule de son mari.

viii. Exemple complet

Nous allons exposer ces techniques en s'appuyant sur l'exemple suivant

Exemple :



Appliquons successivement trois règles :

Règle 1 : Les entités deviennent des relations (ou tables) ; l'identifiant de l'entité devient la clé de la relation ; les propriétés de l'entité deviennent des attributs.

On obtient donc immédiatement les relations :

CLIENT (numcli, nomcli, adcli)
COMMANDE (numcom, datcom)
ARTICLE (numart, libelart, pu)
REPRESENTANT (numrep, nomrep)
REGION (nomreg, nbh)

Règle 2 : Quand on trouve une cardinalité 1,1, on ajoute l'identifiant cible à la relation source (il devient une clé étrangère)

L'entité COMMANDE (donc relation) possède deux associations avec des cardinalités 1,1. On modifie alors le schéma relationnel ci-dessus de la façon suivante en appliquant la règle 2 :

COMMANDE (numcom, datcom, #numrep, #numcli)

Règle 3 : Une association de la forme x,n -------y,n devient une relation à part entière dont la clé est la concaténation des deux identifiants des entités concernées et dont les attributs sont les propriétés (si elles existent) de l'association.

On obtient donc les relations supplémentaires suivantes :

PROSPECTION (#numrep, #nomreg)
LIGNE_COMMANDE (#numcom, #numart, qtecom)

En définitive on aboutit au modèle logique des données (dans le formalisme relationnel) :

CLIENT (numcli, nomcli, adcli)
COMMANDE (numcom, datcom, #numrep, #numcli)
ARTICLE (numart, libelart, pu)

REPRESENTANT (numrep, nomrep)
REGION (nomreg, nbh)
PROSPECTION (#numrep, #nomreg)
LIGNE_COMMANDE (#numcom, #numart, qtecom)

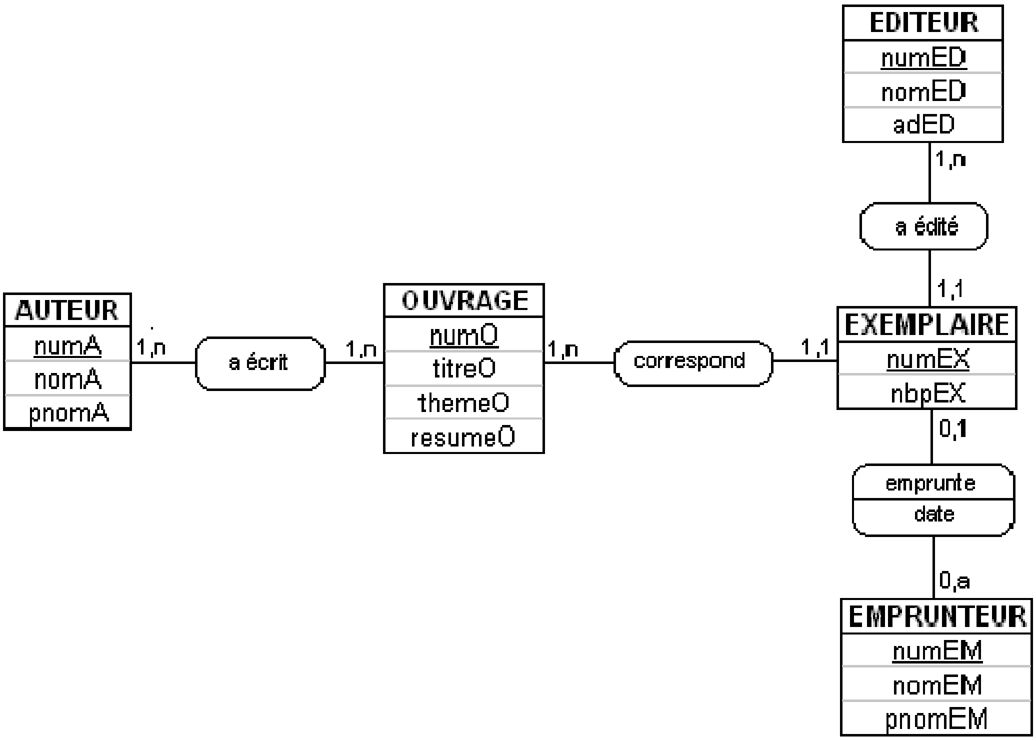
NB : le signe # permet de repérer les clés étrangères et ne fait pas partie du nom de l'attribut.

Série n° 3 : Modèle relationnel

Exercice 1 :

Reprendre les schémas Entité Association faits dans les trois premiers exercices de la série précédente et indiquer des schémas relationnels pouvant convenir…

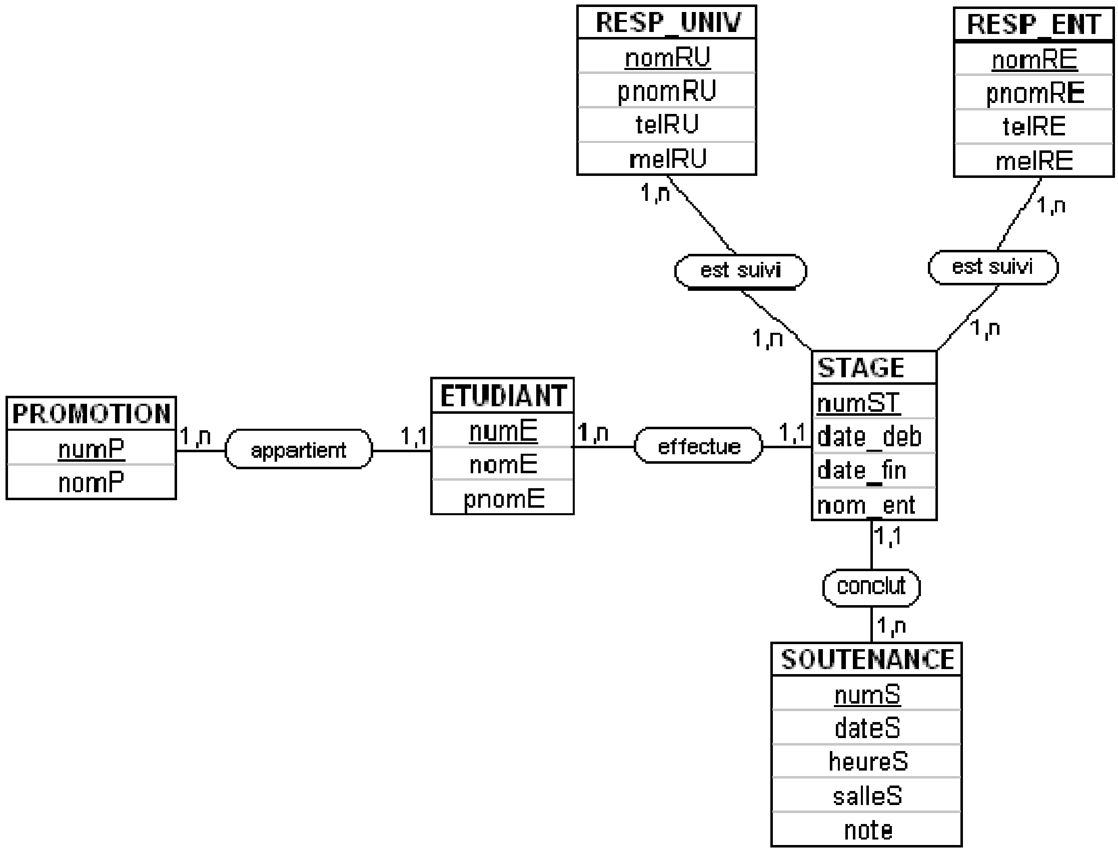
Nous avons obtenu le modèle conceptuel de données (entité-association) suivant :



Appliquons les trois règles pratiques.
Règle 1 : Toutes les entités deviennent des relations :
AUTEUR(numA, nomA, pnomA)
OUVRAGE(numO, titreO, themeO, resumeO)
EXEMPLAIRE(numEX, nbpEX)
EDITEUR(numED, nomED, adED)
EMPRUNTEUR(numEM, nomEM, pnomEM)
Règle 2 : Traitement des cardinalités 1,1 ou 0,1 : introduction de clés étrangères
EXEMPLAIRE(numEX, nbpEX, #numO, #numED, #numEM)
Règle 3 : Traitement des associations x,n--y,n : des associations deviennent des relations
ECRITURE(#numA, #numO)
En définitive, on obtient le schéma relationnel suivant
AUTEUR(numA, nomA, pnomA)
OUVRAGE(numO, titreO, themeO, resumeO)
EXEMPLAIRE(numEX, nbpEX, #numO, #numED, #numEM)
EDITEUR(numED, nomED, adED)
EMPRUNTEUR(numEM, nomEM, pnomEM)
ECRITURE(#numA, #numO)

Solution de l'exercice 2

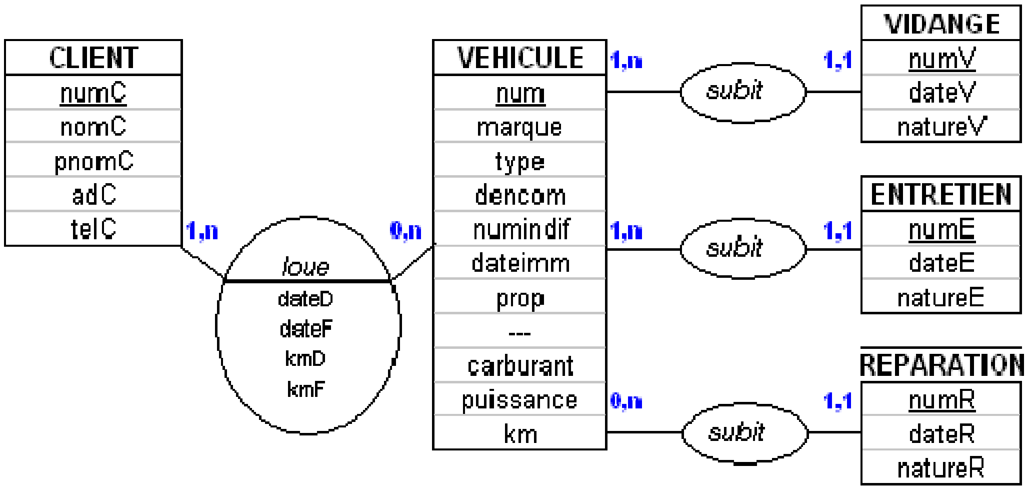
Le modèle entité-association obtenu était :



Appliquons les règles pratiques.
Règle 1 : les entités deviennent des relations
PROMOTION(<u>numP</u>, nomP)
ETUDIANT(<u>numE</u>, nomE, pnomE)
STAGE(<u>numST</u>, date_deb, date_fin, nom_ent)
SOUTENANCE(<u>numS</u>, dateS, heureS, salleS, note)
RESP_UNIV(<u>nomRU</u>, pnomRU, telRU, melRU)
RESP_ENT(<u>nomRE</u>, pnomRE, telRE, melRE)
Règle 2 : Traitement des cardibalités 1,1
ETUDIANT(<u>numE</u>, nomE, pnomE, #numP)
STAGE(<u>numST</u>, date_deb, date_fin, nom_ent, #numS)
Règle 3 : Traitement des associations x,n--y,n)
SUIVI_U(<u>#numST, #nomRU</u>)
SUIVI_E(<u>#numST,#nomRE</u>)
En définitive, on obtient le schéma relationnel suivant :
PROMOTION(<u>numP</u>, nomP)
ETUDIANT(<u>numE</u>, nomE, pnomE, #numP)
STAGE(<u>numST</u>, date_deb, date_fin, nom_ent, #numS)
SOUTENANCE(<u>numS</u>, dateS, heureS, salleS, note)
RESP_UNIV(<u>nomRU</u>, pnomRU, telRU, melRU)
RESP_ENT(<u>nomRE</u>, pnomRE, telRE, melRE)
SUIVI_U(<u>#numST, #nomRU</u>)
SUIVI_E(<u>#numST,#nomRE</u>)

Solution de l'exercice 3

Nous partons du modèle entité-association et nous appliquons les trois règles "standards" :



Règle 1 : les entités deviennent des relations :
CLIENT(numC, nomC, pnomC, adC, telC)
VEHICULE (num, marque, type, ----)
VIDANGE (numV, dateV, natureV)
ENTRETIEN (numE, dateE, natureE)
REPARATION (numR, dateR, natureR)
Règle 2 : traitement des cardinalités 1,1
VIDANGE (numV, dateV, natureV, #num)
ENTRETIEN (numE, dateE, natureE, #num)
REPARATION (numR, dateR, natureR, #num)
Règle 3 : traitement des associations x,n--y,n
LOCATION (#num, #numC, dateD, dateF, kmD, kmF)
On a donc en définitive le schéma suivant
CLIENT(numC, nomC, pnomC, adC, telC)
VEHICULE (num, marque, type, ----)
VIDANGE (numV, dateV, natureV, #num)
ENTRETIEN (numE, dateE, natureE, #num)
REPARATION (numR, dateR, natureR, #num)
LOCATION (#num, #numC, dateD, dateF, kmD, kmF)

3. L'algèbre relationnelle

i. Introduction

Nous introduisons la famille d'opérations habituellement associées au modèle relationnel. Il existe deux types de notations employées pour exprimer ces opérations :

- une notation algébrique dite « algèbre relationnelle » où les requêtes sont exprimées en appliquant des opérateurs aux relations ; c'est un langage procédural.
- une notation logique dite « calcul relationnel » où les requêtes sont exprimées sous forme de formules logiques que doivent satisfaire les tuples recherchés (langage prédicatif).

Cette seconde classe de langages (les langages prédicatifs) se divise en deux :

- ceux pour lesquels les objets primitifs sont des n-uplets encore appelés tuples : c'est le Calcul Relationnel à Variables Tuples (CRVT),
- ceux pour lesquels les objets primitifs sont les attributs des tuples appartenant à différents domaines : c'est le Calcul relationnel à Variables Domaines (CRVD).

ii. Généralités

Rappel : une relation d'arité k est un ensemble de k-uplets éléments d'un **produit cartésien** E1 x E2 x ... x Ek.

On donne en général un nom à chaque élément de tuple : c'est un attribut.
Moyennant cette dénomination des attributs, l'ordre des attributs n'a pas d'importance.
Nous supposerons toutefois qu'un certain ordre a été choisi.

Notation : on dénotera en minuscules les noms de relations (exemple : r), par le même nom en majuscules le schéma relationnel auquel elle appartient (exemple : R) et par Ur l'ensemble des attributs de la relation r, UR l'ensemble des attributs du schéma R, U tout simplement s'il n'y a pas d'ambiguïté.

Remarque : on suppose toujours qu'une relation est finie. Le complément d'une relation est en général infini ; il n'y a donc aucun moyen de le lister.

L'algèbre relationnelle est constituée d'opérateurs qui prennent en argument des relations :

- relations constantes (par exemple décrites en extension),
- relations variables : noms de relations dont le contenu peut varier mais dont l'arité est fixée.

Pour plus de commodité, on sépare les opérateurs en deux catégories :

- **opérateurs de base** : union, différence, produit, projection, sélection,
- **opérateurs additionnels** qui s'expriment à partir des opérateurs de base : intersection, quotient, c-jointure, jointure naturelle, semi-jointure.

4. Les cinq opérateurs de base

i. Union : ∪

Soit r et s deux relations **compatibles**, c'est-à-dire deux parties du même produit E_1 x ... x E_k, alors r∪ s est l'ensemble des tuples qui sont dans r et dans s.
Notons que si deux relations appartiennent au même schéma, alors elles sont évidemment compatibles.

exemple 1 *:* soit r et s deux relations de schéma Etudiant (Nom, Numéro, Ville) définies en extension par :

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

s =

Nom	Numéro	Ville
Martin	98213	Paris
Duval	95565	Lille
Dupont	99123	Lyon

alors r ∪ s =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon
Duval	95565	Lille

Remarque : si r et s contiennent respectivement m et n tuples, l'union r ∪ s contient au plus m+n tuples.

ii. Différence : -

On suppose comme pour l'union que les deux relations r et s sont compatibles. Alors r - s est l'ensemble des tuples qui sont dans r mais pas dans s.

exemple 2 : avec les relations r et s de l'exemple 1 on obtient

r - s =

Nom	Numéro	Ville
Durand	99345	Orléans
Duval	95564	Rennes
Martin	98214	Paris

Remarque : si r et s contiennent respectivement m et n tuples, la différence r - s contient entre m-n et m tuples.

iii. Produit : x

Si r est une relation d'arité p et s une relation d'arité q, alors r x s est l'ensemble des (p+q)-uplets formés en mettant bout à bout un tuple de r et un tuple de s.

exemple 3 : soit r et s les relations ainsi définies

r =

A	B
a1	b1
a2	b2
a3	b3

s =

C	D
c1	d1
c2	d2

alors r x s =

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d1
a2	b2	c2	d2
a3	b3	c1	d1
a3	b3	c2	d2

Remarques :
Si r et s contiennent respectivement m et n tuples, r x s contient exactement m x n tuples.

Lorsque r et s ont des noms d'attributs en commun, il faut les renommer pour éviter les confusions. Si r a les attributs A et B, s a les attributs B et C, on pourra désigner les attributs de r x s par A, B1, B2, C.
C'est le cas dans le produit r x r de r par elle-même ; si r a les attributs A1 et B, r x r a les attributs A1, A2, B1, B2.

iv. Projection : Π

Soit r une relation d'arité k ayant pour attributs U = {A_1, A_2, ... A_k} . Soit i_1, i_2, ... i_p des entiers distincts compris entre 1 et k. La projection de r sur les attributs de numéros i_1, i_2, ... i_p notée $\Pi_{i1, i2, \ldots ip}$ est obtenue en ne gardant des tuples de r que la partie correspondant aux attributs de numéros i_1, i_2, ... i_p.

exemple 4 : reprenons la première relation de l'exemple 1

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

alors $\Pi_{1,3}(r)$ =

Nom	Ville
Durand	Orléans
Martin	Paris
Duval	Rennes
Dupont	Lyon

Remarque : si la relation de départ a n tuples, la relation provenant de la projection a au plus n tuples, car plusieurs tuples peuvent se projeter sur le même tuple. C'est ici le cas de <Martin, 98213, Paris> et <Martin, 98214, Paris> qui produisent tous les deux <Martin, Paris>.
Les attributs de la projection peuvent être désignés par leurs noms plutôt que par leurs numéros.
Dans l'exemple 4 ci-dessus, la projection $\Pi_{1,3}$ (r) se note également $\Pi_{nom,\ ville}$(r).

v. Sélection : σ

Soient

- r une relation et f une formule logique construite à partir de constantes des attributs de r désignés par leurs noms ou leurs numéros,
- les opérateurs de comparaisons =, ≠, <, ≤, >, ≥
- les connecteurs logiques ∨, ∧, ¬

la relation σ_f (r) désigne l'ensemble des tuples u de r tels que f(u) donne la valeur "vrai".

Remarque : la relation σ_f (r) a en général moins de tuples que r.

exemple 5 : reprenons la première relation de l'exemple 1

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

alors, $\sigma_{nom= \text{« Martin »}}$(r) =

Nom	Numéro	Ville
Martin	98213	Paris
Martin	98214	Paris

$\sigma_{nom= \text{« Martin »} \vee\ ville= \text{« Orléans »}}$(r) =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Martin	98214	Paris

Lorsque l'on souhaite désigner un attribut par son numéro, on précède son nom par le caractère $ pour éviter la confusion avec une constante. Ainsi $\sigma_{nom= \text{« Martin »}}$(r) se récrit $\sigma_{\$1= \text{« Martin »}}$(r).

5. Opérateurs additionnels

i. Intersection : ∩

Etant donné deux relations compatibles r et s, l'intersection r ∩ s est constituée des tuples qui sont à la fois dans r et dans s.

exemple 6 : reprenons les relations r et s de l'exemple 1.

r =

Nom	Numéro	Ville
Durand	99345	Orléans
Martin	98213	Paris
Duval	95564	Rennes
Martin	98214	Paris
Dupont	99123	Lyon

s =

Nom	Numéro	Ville
Martin	98213	Paris
Duval	95565	Lille
Dupont	99123	Lyon

alors $r \cap s$ =

Nom	Numéro	Ville
Martin	98213	Paris
Dupont	99123	Lyon

L'intersection s'exprime avec les opérateurs de base : $r \cap s = r - (r - s) = s - (s - r)$.

ii. Quotient : ÷

Soit r une relation d'arité n et s une relation dont les attributs U_s sont une partie stricte de U_r ; s a donc une arité p inférieure à n. Alors r ÷ s est l'ensemble des (n-p)-uplets appelés t portant sur l'ensemble d'attributs $U_T = U_R - U_S$ tels que, pour tout tuple u de s, le tuple t.u sur U_R obtenu en concaténant t et u est dans r (en faisant abstraction de l'ordre des attributs).

exemple 7 : considérons

r =

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c2	d2
a3	b3	c1	d1
a3	b3	c2	d2
a1	b1	c3	d3

et s =

C	D
c1	d1
c2	d2

alors r ÷ s =

A	B
a1	b1
a3	b3

car le produit (r ÷ s) × s donne les tuples

a1 b1 c1 d1
a1 b1 c2 d2
a3 b3 c1 d1
a3 b3 c2 d2

qui sont tous dans r.

a2 b2 n'est pas dans le quotient car le produit par s donne les deux tuples

a2 b2 c1 d1
a2 b2 c2 d2

le premier d'entre eux ne fait pas partie de r.

Remarque :
Le quotient s'exprime avec les opérateurs de base. Pour l'exemple 7 ci-dessus,
$r \div s = \Pi_{A,B}(r) - \Pi_{A,B}\left((\Pi_{A,B}(r) \times s) - r\right)$.

iii. c-jointure : ►◄$_c$

r ►◄$_c$s est l'ensemble des tuples de r × s qui satisfont la condition c. C'est donc $\int_c (r \times s)$

iv. jointure naturelle : ►◄

On suppose que r et s ont des attributs nommés, et qu'un certain nombre d'attributs sont partagés par les deux relations, alors la jointure naturelle r►◄s est obtenue en calculant le produit cartésien r × s sélectionnant les tuples dans lesquels les attributs de mêmes noms ont mêmes valeurs, supprimant les valeurs doubles.

Pour résumer r ►◄ s = $\Pi_{Ur \cup Us}(\sigma_{=}(r \times s))$ où $\sigma_{=}(r \times s)$ sélectionne dans le produit les tuples qui ont mêmes valeurs sur les attributs en commun.

exemple 8 :

soit r =

A	B	C
a1	b1	c1
a2	b1	c1
a3	b1	c2
a4	b2	c3

et s =

B	C	D
b1	c1	d1
b1	c1	d2
b2	c3	d3

alors $r \bowtie s$ =

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b1	c1	d1
a2	b1	c1	d2
a4	b2	c3	d3

$r \bowtie s$ s'exprime évidemment avec les opérateurs de base. Pour l'exemple ci-dessus cela donne

$r \bowtie s = \Pi_{1,2,3,6}\,(\,\sigma_{\$2=\$4 \,\wedge\, \$3=\$5}\,(\,r \times s\,))$

Remarques :
Si r et s n'ont aucun attribut en commun, c'est-à-dire si $U_r \cap U_s = \varnothing$, alors $r \bowtie s = r \times s$
Si r et s ont tous leurs attributs en commun, c'est-à-dire si $U_r = U_s$, alors $r \bowtie s = r \cap s$.

v. Semi-jointure $\ltimes$

$r \ltimes s$ est la projection sur les attributs de r de la jointure naturelle $r \bowtie s$.

C'est donc $\Pi_{Ur}\,(\,r \bowtie s\,)$.

6. Propriétés des opérateurs.

L'union est associative et commutative.
Le produit × est associatif, et commutatif si l'on fait abstraction de l'ordre des attributs.
La jointure est commutative (si l'on fait abstraction de l'ordre des attributs), et **associative** :

$(\,r \bowtie s\,) \bowtie t = r \bowtie (\,s \bowtie t\,)$

parce que c'est l'ensemble des tuples u sur l'union des ensembles d'attributs $U_r \cup U_s \cup U_t$ tels que

(1) $\Pi_{Ur}(u) \in r \wedge \Pi_{Us}(u) \in s \wedge \Pi_{Ut}(u) \in t$

Il est en effet évident que si u satisfait (1) alors

$\Pi_{Ur \cup Us}(u) \in r \bowtie s$, $\Pi_{Ut}(u) \in t$, et donc $u \in (r \bowtie s) \bowtie t$.

Inversement, si $u \in (r \bowtie s) \bowtie t$ alors $\Pi_{Ur \cup Us}(u) \in r \bowtie s$ donc $\Pi_{Ur}(\Pi_{Ur \cup Us}(u))$ $= \Pi_{Ur}(u) \in r$,

$\Pi_{Us}(\Pi_{Ur \cup Us}(u)) = \Pi_{Us}(u) \in s$, et d'autre part $\Pi_{Ut}(u) \in t$.

Ceci se généralise en la proposition suivante :

Proposition :

$r_1 \bowtie r_2 \bowtie \ldots \bowtie r_n$ est l'ensemble des tuples u sur $U_{r1} \cup U_{r2} \ldots \cup U_{rn}$ tels que $\Pi_{Uri}(t) \in r_i \; \forall i$.

7. Algèbre relationnelle comme langage de requêtes

Considérons les trois schémas Etudiants, Inscriptions et Enseignements suivants :

Etudiants (nom-et, num-et, adr)

exemple de table étudiants :

Nom-et	Num-et	Adr
Martin	347	Orléans
Durand	1024	St Jean le Blanc
Duval	425	Orléans
Dupuis	248	St Cyr en val
Martin	504	Olivet

Inscriptions (num-et, module)

exemple de table inscriptions :

Num-et	module
347	BD2
1024	BD3
248	RO
1024	BD3
504	SYS
347	RO

Enseignements (module, jour, heure, salle, nom-ens) exemple de table enseignements :

Module	Jour	Heure	Salle	Nom-ens
SYS	Lun	9	E05	Rousseau
BD2	Mar	10	E12	Proust
BD3	Lun	14	E05	Durand
BD2	Jeu	16	E08	Durand
RO	Lun	14	E08	Rousseau

Ces trois tables ont la signification suivante : l'étudiant de numéro 347 par exemple se nomme Martin et habite à Orléans ; il est inscrit aux modules BD2 et RO ; au titre de BD2 il suit les enseignements du mardi 10h en salle E12 assuré par Proust et du jeudi 16h en salle E08 assuré par Durand.

Voici comment on peut exprimer des requêtes sur cette base en utilisant les opérateurs de l'algèbre relationnelle.

R1- Numéros d'inscriptions des étudiants se nommant Martin :

$\Pi_{\text{num-et}}\ (\sigma_{\text{nom-et = « Martin »}}\ (\text{étudiants}))$

R2- Modules d'enseignements qui ont lieu le lundi en salle E05 :

$\Pi_{\text{module}}\ (\sigma_{\text{jour = lun} \wedge \text{salle = E05}}\ (\text{enseignements}))$

R3- Modules suivis par les étudiants se nommant Martin :

1ère méthode : faire d'abord une jointure, puis extraire :

$\Pi_{\text{module}}\ (\sigma_{\text{nom-et = « Martin »}}\ (\text{étudiants} \blacktriangleright\blacktriangleleft \text{inscriptions}))$

2ème méthode : réduire d'abord étudiants à ceux qui s'appellent Martin :

$\Pi_{\text{module}}\ ((\sigma_{\text{nom-et = « Martin »}}\ (\text{étudiants})) \blacktriangleright\blacktriangleleft \text{inscriptions})$

R4- Couples étudiant-enseignant tels que l'étudiant connaît l'enseignant : il suit au moins un cours assuré par celui-ci :

$\Pi_{\text{nom-et, nom-ens}}\ (\Pi_{\text{nom-et, num-et}}\ (\text{étudiants}) \blacktriangleright\blacktriangleleft \text{inscriptions} \blacktriangleright\blacktriangleleft \Pi_{\text{module, nom-ens}}(\text{enseignements}))$

R5- Etudiants qui suivent tous les enseignements de l'enseignant Durand (on suppose que lorsqu'un étudiant est inscrit à un module il suit tous les enseignements de ce module) : on fait la division de l'ensemble des couples num-et - module par l'ensemble des modules dans lesquels enseigne Durand

$\Pi_{\text{num-et, module}}\ (\text{inscriptions})\ ¸\ \Pi_{\text{module}}\ (\sigma_{\text{nom-ens = « Durand »}}\ (\text{enseignements}))$

R6- Couples étudiants-enseignants tels que l'étudiant est fidèle à l'enseignant (il suit tous les cours assurés par celui-ci). Idée : générer tous les couples étudiant-enseignant et en extraire les mauvais : ceux obtenus en combinant un étudiant, un module dans lequel il n'est pas inscrit, et un enseignant qui enseigne dans ce module ; les couples étudiant - module dans lequel il n'est pas inscrit sont obtenus par différence entre tous les couples possibles et ceux correspondant aux inscriptions.

$\Pi_{\text{num-et}}(\text{inscriptions}) \times \Pi_{\text{nom-ens}}(\text{enseignements}) - \Pi_{\text{num-et, nom-ens}}\left((\Pi_{\text{num-et}}(\text{inscriptions}) \times \Pi_{\text{module}}(\text{inscriptions}) - \text{inscriptions})) \blacktriangleright\!\blacktriangleleft \Pi_{\text{module, nom-ens}}(\text{enseignements}) \right)$

exemple :

si la table inscriptions et la restriction de enseignements à module+enseignant comprennent :

e1	m1		m1	p1
e2	m1		m1	p2
e2	m2		m2	p1
e3	m3		m3	p3

il faut trouver :

e2		p1
e3		p3