

Bases de données Modèle relationnel

Module de cours BD011 Gaspard et Madeleine

par
Luc Lavoie

Version
070a
Statut
document de travail

dernière modification : 2017-08-28
dernière diffusion : 2012-10-16

Introduction	3
Objet et portée du document	3
Évolution du document	3
Problématique.....	3
Organisation du module.....	3
Travail en cours ou projeté.....	3
Présentation	4
Une petite histoire	4
Quelques commentaires	6
Exemple	7
Un premier survol.....	7
Quelques relations.....	8
Encore un peu d'analyse	9
En avant vers SQL	11
Retour à modélisation.....	17
A. Comment bien consommer un joint.....	21
B. Les colles du professeur Tournesol.....	24
C. Madeleine, Gaspard et l'absinthe	25

Historique des révisions

version	date	auteur	description
0.7.1a	2017-08-27	LL	...
0.7.0a	2015-01-31	LL	Revue à la lumière des récentes prestations de cours.
0.6.4a	2014-01-12	LL	Correction de coquilles et uniformisation de certaines notations.
0.6.3a	2013-07-24	LL	Intégration des résultats d'expérience 2012.
0.6.2a	2012-10-27	LL	Remise à plat du modèle, correction des noms d'entités
0.6.1a	2012-10-27	LL	Considérations générales sur les valeurs, les représentations, les domaines et les types
0.6.0a	2012-10-27	LL	Traitement de la requête R03
0.5.0a	2012-10-20	LL	Première élaboration des sections 3.4 et 3.5
0.4.0b	2012-10-17	LL	Corrections de coquilles.
0.4.0a	2012-10-14	LL	Deuxième version de la solution corrigée par l'étudiant de Madeleine.
0.3.0a	2012-10-01	LL+CK	Première version de la solution partielle de Madeleine ; ajout de l'iconographie.
0.2.0a	2012-10-01	LL	Précisions apportées à l'exercice
0.1.1a	2012-09-30	LL	Intégration des commentaires
0.1.0b	2012-09-30	CK	Première revue
0.1.0a	2012-09-30	LL	Bon pour revue (exemple à faire)
0.0.1a	2012-09-18	LL	Première ébauche du plan

Sommaire

Le module Gaspard et Madeleine

Le présent module de cours a été rédigé dans le cadre de l'exploration du thème Bases de données par le groupe de réflexion Ἀκαδήμεια. Le module présente un exemple simple utilisé aux fins d'illustration par les modules traitant du modèle relationnel et du langage SQL.

Le thème Bases de données comprend un ensemble de modules destinés aux étudiants des champs disciplinaires de l'informatique, du génie logiciel ou de l'informatique de gestion (niveau du B. Sc., de la licence et du *mastère*). Nous espérons toutefois qu'ils puissent être utilisés avec profit par toute personne curieuse d'appivoiser ce champ de connaissance.

Le groupe Ἀκαδήμεια

Le groupe de réflexion Ἀκαδήμεια s'intéresse à la transmission des connaissances dans le domaine de l'informatique.

© 2009-2017 Groupe Ἀκαδήμεια,
Département d'informatique, Faculté des sciences,
Université de Sherbrooke.
Sherbrooke, QUÉBEC J1K 2R1.

Introduction

Objet et portée du document

Le présent document a pour but présenter un exemple qui sera utilisé dans le cadre des modules de cours subséquents (en particulier le module BD010 traitant du modèle relationnel et le module BD100 traitant du langage SQL).

Évolution du document

La rédaction du document a commencé en aout 2012. La première version du document a été établie sur les bases suivantes :

- ◇ la relecture de certains travaux de Platon, Pappus, Descartes, Tarski, Codd et Date ;
- ◇ l'écoute de Brel et la relecture de Rimbaud ;
- ◇ des réminiscences de quelques tableaux de Vincent Van Gogh et d'Edgard Degas ;
- ◇ quelques faits divers survenus au sommet de Québec, au sommet de Toronto et lors des manifestations du printemps érable ;
- ◇ de nombreux échanges avec Christina Khnaisser.

Problématique

Les concepts mis en cause dans le modèle relationnel sont simples en apparence. Il aura néanmoins fallu presque trois millénaires pour les dégager, les circonscrire et, encore aujourd'hui, il n'est pas certain que nous les ayons bien maîtrisés. Nous cherchons ici, à travers un exemple très simple, à lever le voile sur cette passionnante recherche. Nous cherchons également à présenter un exemple qui nous servira tout au long de la présentation du modèle relationnel lui-même ainsi que lors de la présentation du langage SQL.

Organisation du module

NA

Travail en cours ou projeté

Développer la présentation des domaines et des types.

Présentation

Une petite histoire posera le contexte de notre exemple. Par la suite, nous avons ajouté quelques commentaires, comme autant de sujets de réflexion ou de pistes qui seront explorés dans les modules subséquents.

Une petite histoire

Gaspard est propriétaire-exploitant d'un magasin d'armement. Son entrepôt peut contenir jusqu'à six-mille articles différents, dont certains, en plusieurs centaines d'exemplaires. Pour tout personnel, Gaspard a recours à un magasinier et un commis-vendeur. Gaspard est désireux de bien servir sa clientèle. En particulier, il aimerait pouvoir répondre rapidement et avec certitude à des questions telles que :

- Q1. Telle arme est-elle en stock ?
- Q2. Quelle est sa masse ?
- Q3. D'où provient-elle ?
- Q4. Combien d'exemplaires y a-t-il en stock ?
- Q5. Quels fournisseurs ont livré des armes de ce type ?
- Q6. Quelles sont les armes de masse inférieure à une masse donnée ?
- Q7. Quelles sont les munitions appropriées pour une arme donnée ?
- Q8. Quelles sont les autres armes qui utilisent les mêmes munitions qu'une arme donnée ?
- Q9. Combien d'exemplaires de chacune de ces armes y a-t-il en stock ?

À la base, Gaspard connaît certains faits¹, par exemple :

- F1. Latendresse & filles a livré 24 canons le 2012-04-01.
- F2. Boucher & fils a livré 120 pistolets le 2012-04-08.
- F3. La masse d'un canon est de 454 grammes².
- F4. La masse d'un fusil est de 250 grammes.
- F5. Latendresse & filles est localisé à Montréal.
- F6. Boucher & fils est localisé à Québec.
- F7. 50 canons ont été expédiés à Sganarelle le 2012-10-24.

Il remarque que ces faits sont « stéréotypés », c'est à dit qu'on peut les regrouper en catégories selon leur structure, par exemple

- ◇ F1 et F2 : Le fournisseur F a livré N armes de type A le jour J.
- ◇ F3 et F4 : La masse d'une arme de type A est de G grammes.
- ◇ F5 et F6 : Le fournisseur F est à l'adresse X.

Gaspard, bien qu'il soit marchand d'armes, a approché la poésie dans sa jeunesse (comme Rimbaud, pense-t-il). Il se pique également d'être philosophe (comme Alexandre le Grand, paraît-il). Il comprend

¹ Gaspard dispose de plusieurs autres faits qu'il n'a pas pris le soin de recenser et dont pourtant il a l'usage. Par exemple, pour connaître le nombre d'armes en stock, il faut connaître les ventes.

² La plupart des clients désirent plutôt en connaître le poids, du moins le pensent-ils. La confusion est particulièrement aigüe dans les pays n'ayant pas adopté le SI. Revoir vos cours de physique... où le court texte de vulgarisation suivant : <http://phys.free.fr/maspoids.htm>

que ces « faits » sont des propositions et ces stéréotypes des prédicats. Il rêve dès lors de représenter le monde (de son entrepôt) par un modèle logique.

Ainsi le prédicat

Masse (A, G)

veut dire

« Une arme de type A a une masse de G grammes. »

et l'instance

Masse $\langle A : \text{'Canon'}, G : 256 \rangle$

représenterait la proposition (vraie) suivante :

« Une arme de type Canon » a une masse de 256 grammes. »

Ainsi le prédicat

Livraison (F, N, A, J)

veut dire

« Le fournisseur F a livré N armes de type A le jour J . »

et l'instance

Livraison $\langle F : \text{'Latendresse & filles'}, N : 24, A : \text{'Canon'}, J : 2012-04-01 \rangle$

représenterait la proposition (vraie) suivante :³

« Le fournisseur 'Latendresse & filles' a livré 24 armes de type 'Canon' le 1^{er} avril 2012. »

Gaspard connaît le calcul des prédicats, la logique classique, et y voit une bonne base pour l'interrogation de l'état de son entrepôt, surtout s'il ajoute la quantification !

Par exemple pour savoir si 'Latendresse & filles' a fait au moins une livraison en 2012, il suffit d'évaluer le prédicat suivant :

$$\exists f. \text{Livraison}(f, n, a, j) \wedge (2012-01-01 \leq j) \wedge (j \leq 2012-12-31)$$

Peut-il aller encore plus loin ? Il aimerait pouvoir énumérer et compter. Il fait donc appel aux ensembles et aux entiers.

Ainsi, la fonction t définie comme suit :

$$t(a) = \{f \mid \text{Livraison}(f, N, a, J)\}$$

permet de répondre à la question

« Quels sont les fournisseurs ayant livré des armes de type a ? »⁴.

De même, la fonction m définie comme suit⁵ :

$$m(k) = \#\{a \mid \text{Masse}(a, G) \text{ et } (G < k)\}$$

permet de répondre à la question

« Combien de types d'armes ont une masse inférieure à k grammes ? ».

Gaspard sait en outre qu'il existe des règles de calcul, des algorithmes, lui permettant de connaître la valeur de ces ensembles et de les dénombrer. Voilà Gaspard heureux et satisfait, il pourra répondre aux attentes de ses clients (et aux siennes).

Le peut-il vraiment ? Seulement si son monde imaginaire correspond bien au monde réel ! Est-ce la fin de son rêve ? Gaspard se rappelle alors les leçons de son bon maître Bertrand : un système de prédicats

³ On remarque au passage le changement de représentation de la date. Ceci est volontaire et se veut une illustration de la différence entre une valeur et une représentation. En effet, les deux représentations désignent la même date. Il sera donc de première importance de s'assurer que toute comparaison est effectuée sur les valeurs et non sur la représentation de celles-ci. Par exemple, si Gaspard demande la liste des livraisons du 2012-04-01, il désire forcément avoir celles du 1^{er} avril 2012. Si le plus simple est d'uniformiser les représentations, ceci n'est pas toujours possible en pratique. Il faut alors s'assurer de mécanismes appropriés. Ces mécanismes sont des fonctions dites de conversion... elles nous causeront bien des soucis.

⁴ On remarque que la fonction $t(a)$ ne répond pas adéquatement à la question « Quels sont les fournisseurs d'armes de type a ? ». Pourquoi ?

⁵ Le symbole $\#$ représente la fonction retournant la cardinalité d'un ensemble. Nous utiliserons cette notation dorénavant, même si elle est pas universelle (bien que fréquente).

permet de répondre à toutes les questions⁶ si et seulement s'il est clos. Il fait donc l'hypothèse d'un monde clos et, pour cela, définit deux fonctions de correspondance : une du monde réel vers le monde imaginaire et l'autre du monde imaginaire vers le monde réel. Idéalement, ses fonctions sont l'inverse l'une de l'autre⁷.

Sur ce, Gaspard se réveille et procède rapidement à sa toilette afin de ne pas être en retard pour l'ouverture de son magasin.

Après sa journée de travail, autour d'un verre d'absinthe et d'une bière, Gaspard raconte son idée à Madeleine qui s'exclame « mais oui, la relation est la base de tout ! » Gaspard croit comprendre que Madeleine serait enfin sensible à ses avances. Madeleine le détrompe indirectement en lui expliquant : « Une relation est un sous-ensemble d'un produit cartésien⁸. On peut la représenter comme un graphe, ou comme une table. Par exemple, les faits décrits par la relation Livraison pourraient être consignés sous la forme d'une table comme celle-ci » et elle s'empresse de griffonner le tableau suivant sur une serviette de papier :

Livraison			
F	N	A	J
Latendresse & filles	24	Canon	2012-04-01
Boucher & fils	120	Pistolet	2012-04-08
Nakamura SA	32	Catapulte	2012-04-12

La relation est donc une représentation du prédicat et le tuple, une représentation de la proposition. Madeleine n'obtient qu'un commentaire lacunaire de Gaspard : « joli ». Il montre par là son peu d'intérêt pour la beauté mathématique (et, sur un autre plan, sans doute aussi sa déception amoureuse). Madeleine ne se décourage pas et poursuit : « Nous avons donc toute une gamme d'opérations sur les relations : l'union, l'intersection, la différence (car les relations demeurent des ensembles), mais aussi des opérations spécifiques aux relations : la projection, la restriction et la jointure ». Enfin, Gaspard semble s'ouvrir aux explications de Madeleine (avec quelque difficulté, puisqu'il en est à sa troisième absinthe) : « Les opérations ensemblistes, je comprends, mais les opérations relationnelles m'intriguent, elles me semblent mal définies. Par exemple, dans la projection, comment décides-tu des colonnes à enlever ? » Madeleine est heureuse de voir l'intérêt de Gaspard, mais pousse un soupir (*in petto*). Elle explique : « C'est à cela que servent les noms de colonnes ! »

Soudain, le coup de génie ! Gaspard : « Mais oui, et il faudra alors une opération pour renommer les colonnes afin de pouvoir composer librement les relations entre elles. » Madeleine lui laisse croire qu'il vient de faire une découverte lumineuse et omet de lui dire que E. F. Codd avait déjà découvert tout ça en 1969 : Madeleine est informaticienne.

De l'autre côté de la rue, Jacques les observe.

Quelques commentaires

Aux fins de notre histoire, nous avons implicitement convenu que nos ensembles étaient finis. Comme c'est également le cas de la mémoire d'un ordinateur, celle-ci semble être réaliste, voire incontournable. En fait, c'est bien plus que ça... Kurt nous l'expliquera bientôt.

⁶ Les seules questions bien formulées (*well formed*) comme il se doit. Bertrand étant anglais, cela va sans dire, *indeed*.

⁷ Socrate, par l'entremise de Platon, vient nous titiller avec une histoire confuse de caverne avec laquelle il nous embrouille à ne plus savoir si nous sommes dans le monde réel ou dans le monde imaginaire. Nous décidons de l'ignorer. Henri se ramène aussitôt et souligne que cela ressemble furieusement à des morphismes et nous oriente vers la topologie algébrique pour obtenir une vision unifiée des mondes ouverts et fermés. Kurt rappelle lui aussi et souligne que dès lors qu'on désire énumérer et compter, certaines questions pourraient ne pas avoir de réponses. Gaspard se demande bien pourquoi et considère Henri et Kurt comme des drôles d'illuminés.

⁸ Tiens, il fallait que René s'invite et, en plus, qu'il en remette avec une histoire de diabolotin posté à l'entrée de la caverne de Platon — nous l'ignorons lui aussi. Plaise à Dieu que ce joueur de Blaise reste à l'écart !

Il fallait un peu de poésie pour concevoir un monde imaginaire construit sur la logique des philosophes et les ensembles des mathématiciens. Le prix Turing a été décerné à E. F. Codd en reconnaissance de cette contribution majeure à la science. En fait, il en découle que le calcul au sein de ce monde imaginaire est de celle qu'on écarte immédiatement tant elle est facile⁹ alors que celle de la détermination des fonctions entre le monde réel et le monde imaginaire¹⁰ est beaucoup plus difficile.

Je vous sens sceptique. Vous présentez que le calcul est parsemé d'embuches. Vous n'avez pas tort (sans avoir tout à fait raison - le vrai monde n'est jamais tout à fait clos¹¹). Pour cette raison, nous aborderons **d'abord** la question du calcul dans le cadre de notre cours. Ceci réglé, nous aborderons **ensuite** le problème de la modélisation. Bien sûr, nous n'épuiserons aucun de ces sujets. C'est pourquoi le Département d'informatique offre pas moins de cinq autres activités reliées à ce domaine de connaissance, trois au premier cycle (IFT 287, IGE 487, IFT 501) et deux au second cycle : (BIN 701, IFT 723). De plus, les connaissances que nous couvrons sont nécessaires aux activités, comme IGL 301, IGE 502 et IFT 604 notamment. La génétique, la physique des particules, la modélisation chimique, la géomatique, l'écologie l'environnement et l'imagerie numérique, entre autres disciplines, utilisent des quantités faramineuses de données très complexes que seules les bases de données peuvent permettre d'exploiter de façon fiable, efficace et efficiente. Cette situation justifie le fait que les spécialistes de nombreuses disciplines doivent s'initier, du moins minimalement, aux bases de données qui, à bien des égards, ne sont qu'une transposition des systèmes logiques que nous utilisons quotidiennement en science et en philosophie.

Exemple

Un premier survol

Voici un survol de quelques-unes des informations mises en relation par notre problème.

Les types d'armes

- ◇ Catapulte (à confettis)
- ◇ Pistolet (à eau)
- ◇ Canon (à bulles de savon)
- ◇ Casserole
- ◇ Cuillère à pot
- ◇ Fusil (à bouchons)

Les munitions

- ◇ Recharge de confettis, 5 000 unités
- ◇ Recharge de confettis, 10 000 unités
- ◇ Recharge de savon, 100 ml
- ◇ Recharge de savon, 5 x 100 ml

Les fournisseurs et les villes

- ◇ Latendresse & filles, Montréal
- ◇ Boucher & fils, Québec

⁹ Comme ironiserait peut-être Boris Vian. Qu'on se rassure toutefois, les rédacteurs de la norme SQL (le langage relationnel le plus utilisé) et les éditeurs de systèmes de gestion de bases de données relationnelles, se sont ingénies à compliquer cette tâche, de sorte que le métier d'informaticien spécialiste des bases de données n'a jamais été aussi en demande !

¹⁰ On dit aussi monde virtuel ou modèle.

¹¹ Consulter Michel Tremblay.

- ◇ Nakamura SA, Tokyo
- ◇ Sganarelle, Paris
- ◇ Arlequin, Venise
- ◇ Vladimir, Moscou
- ◇ Charlot & co, Hollywood

Quelques relations

Madeleine va donc travailler avec Gaspard afin de modéliser son entrepôt (ou plutôt le contenu de celui-ci¹²). D'entrée de jeu, Gaspard précise qu'il a plusieurs armes de même type, mais que ce sont pour lui des produits différents (se différenciant par la couleur, les matériaux utilisés ou la décoration). Il distingue chaque produit par un numéro de produit, pour plus de commodité. Par ailleurs, un même produit peut être offert par plus d'un fournisseur et un fournisseur peut proposer plusieurs produits. Il en est de même des munitions. Dans un premier temps, Madeleine propose ceci¹³ :

*Arme : Le produit N est une arme de nom X, de type T, de poids P utilisant des munitions de type M.
Munition : Le produit N est une munition de nom X et de type M.*

- Gaspard réagit tout de suite Attention, certaines armes peuvent utiliser plus d'un type de munition !
- Madeleine lui demande alors : Certaines armes ou certains types d'armes ?
- Gaspard ne saisit pas la nuance : ...
- Madeleine lui explique : Si tu as deux catapultes à confettis de même type, mais de couleurs différentes, utilisent-elles les mêmes types de munitions ? Plus généralement deux armes de mêmes types peuvent-elles se distinguer quant aux munitions qu'elles peuvent utiliser ?
- Gaspard comprend : En effet, le type de munition utilisable dépend du type d'arme.

*Arme : Le produit N est une arme de nom X, de type T et de poids P.
Armement : Le type d'arme T utilise le type de munition M.*

Voilà une description des produits qui semble acceptable.

Ceci permet déjà de répondre à plusieurs questions, mais pas à celles concernant le nombre d'exemplaires en stock. Pour ce faire, Gaspard propose d'ajouter une relation.

Stock : Il y a Q produits N en stock.

- Un numéro de produit d'une arme ne peut donc être le même qu'un numéro de produit d'une munition, sans quoi on ne s'y retrouverait plus.
- Hum ! Il faut donc une contrainte pour s'en assurer.

UnicitéProduit : un même numéro de produit peut désigner soit une arme soit une munition¹⁴.

- Ah, ce n'est pas suffisant, j'achète et je vends, les stocks, ça bouge !
- Bien sûr, c'est classique !

¹² Pourquoi cette nuance est-elle importante ?

¹³ On remarque que Madeleine a renoncé à imposer le nom correct (masse) à son client (Gaspard) qui s'entête à utiliser poids à mauvais escient. Cette situation est fréquente. Madeleine ne devrait-elle toutefois pas utiliser le vocable correct à l'interne (dans le modèle et la documentation technique) et le vocable incorrect à l'externe (dans les documents destinés au client et dans l'interface personne-machine) ? Ces questions sont abordées en IFT 215 et en IGL 301.

¹⁴ L'espace d'un instant, Gaspard se demande pourquoi Madeleine ne s'est pas exprimée plus simplement « Un numéro de produit peut désigner une arme ou une munition », mais la mémoire de ses cours de français lui revient : le « ou » est inclusif, pas exclusif (finalement, le français, c'est comme les mathématiques). Donc, si on veut être précis, non ambigu, il faut formuler l'exclusion explicitement.

*Achat : Le fournisseur F a livré Y exemplaires du produit N à la date D.
 Vente : Le client C a pris possession d'Y exemplaires du produit N à la date D.*

— J'imagine qu'en pratique tu utilises un numéro de fournisseur, un numéro de client et un numéro de produit plutôt que les descriptions de ceux-ci ?

— Comment sais-tu ça ?

— Madeleine soupire, puis ajoute : Il faut donc décrire les clients et les fournisseurs séparément et en profiter pour consigner leur nom, leur adresse et leur numéro de téléphone.

*Fournisseur : Le numéro F correspond au fournisseur de nom X, d'adresse A et de téléphone S.
 Client : Le numéro C correspond au client de nom X, d'adresse A et de téléphone S.*

— Bon, écoute ! Voici ce que ça donne, si on synthétise et si on utilise des noms d'attributs significatifs :

```
Arme(noProduit, nomArme, typeArme, poids, typeMunition)
Munition(noProduit, nomMunition, typeMunition)
Armement(typeArme, typeMunition)

Stock(noProduit, quantite)

Achat(noFournisseur, noProduit, dateLivraison, quantite)
Vente(noClient, noProduit, dateVente, quantite)
Fournisseur(noFournisseur, nomFournisseur, adresseFournisseur, telephoneFournisseur)
Client(noClient, nomClient, adresseClient, telephoneClient)

UnicitéProduit : {p | Arme(p,X,T,P)} ∩ {p | Munition(p,X,M)} = ∅
```

Leur satisfaction est de courte durée.

— Madeleine, il faut aussi que je connaisse les prix ! s'écrit Gaspard. Que je sois capable de facturer !

Il se fait tard et Madeleine travaille tôt (sans compter l'effet des absinthes qu'ils ont continué à siroter tout en travaillant).

— Écoute Gaspard, j'en ai marre, je rentre chez moi. Prise de remords, elle ajoute : Je donne le cours « Éléments de BD » à l'Uni. Je t'enverrai demain un de mes potaches et il terminera ça ! Bon, maintenant, je me casse.

Gaspard est pantois, tant devant la réaction de Madeleine, que par le brusque changement de niveau de langue. Il n'a pas vu Jacques qui est toujours là, de l'autre côté de la rue.

Encore un peu d'analyse

Antoine est inscrit au cours de Madeleine. Celle-ci l'a désigné pour terminer la modélisation qu'elle avait entreprise avec Gaspard. Antoine est plutôt mal à l'aise de prendre ainsi la suite de son enseignante. Ce malaise s'accroît lorsqu'il constate après 30 minutes de discussion avec Gaspard que certains choix de modélisation devraient être revus et corrigés. Habile, il manoeuvre de telle sorte que Gaspard pense qu'il est l'auteur de ces changements. Ainsi, Antoine pense-t-il se mettre hors de cause si jamais Madeleine s'en trouvait froissée.

En même temps, il explore plus à fond certaines questions laissées en suspens, notamment en ce qui concerne le domaine de définition (le type) de certains attributs. Antoine sent déjà venir la prochaine demande de son enseignante : transcrire le modèle en un schéma SQL. Fort heureusement, Antoine est un étudiant méthodique et il transcrit ainsi ses changements et précisions.

- C01. Dans la mesure où le type de munition est déterminé par le type d'arme (voir la relation Armement) et non l'arme elle-même, l'attribut typeMunition de la relation Arme est au mieux redondant, au pire faux. Antoine le retire.
- C02. Gaspard l'informe que numéro de produit est en réalité une étiquette composée de deux « parties », le type de produit et un numéro séquentiel au sein du type de produit. Il y a deux types de produits : les armes (étiquettes débutant par la lettre 'A') et les munitions (étiquettes débutant par la lettre 'M'). Le numéro séquentiel est compris entre '00000' et '99999'. Cette information structurelle doit-elle être conservée dans le modèle ou au contraire retirée ? Gaspard tient beaucoup à sa convention d'étiquetage. À défaut d'information plus probante, la modélisation actuelle est conservée, mais des contraintes devront être ajoutées pour refléter les règles d'étiquetage.
- C03. Le nom de produit est présent dans les relations Arme et Munition. Il convient de les factoriser et de mettre un seul attribut, dans la relation Stock.
- C04. Les types d'armes et de munitions sont des domaines fermés, c'est-à-dire qu'il en existe un nombre restreint et qu'il faut s'assurer que chaque arme (chaque munition) est décrite par un type d'arme (un type de munition) légitime. Il faut donc ajouter une relation pour définir chacun de ces types.
- C05. Gaspard confirme qu'il existe certains fournisseurs qui sont aussi de clients. Pour éviter une redondance ou des incohérences dans les adresses de ceux-ci, il est préférable de séparer la description des organisations du fait qu'elles sont Client ou Fournisseur.
- C06. Il devient alors préférable d'utiliser un seul numéro par organisation, Gaspard propose des numéros compris entre '100000' et '999999'. Il désire ne pas utiliser les numéros inférieurs à '100000' afin d'éviter toute confusion avec les anciens numéros qu'il utilisait.
- C07. Gaspard confirme que les « achats » se déroulent en deux temps, la commande et la livraison. D'ailleurs, on ne reçoit pas toujours la quantité commandée, et pas toujours en une seule livraison. Comme l'objectif du modèle est la gestion de l'entrepôt, il est convenu de se concentrer sur les seules livraisons et de renommer en conséquence les éléments du modèle. Un raisonnement analogue s'applique aux Ventes.
- C08. Manifestement, Gaspard et Madeleine ont oublié l'attribut prix. Où faut-il indiquer le prix ? Clairement les prix d'acquisition (réception) et celui de vente (expédition) sont distincts et, en fait, sont susceptibles de varier d'une transaction à l'autre. Par ailleurs, s'il faut donner un prix à un client, lequel faut-il lui fournir ? Un troisième prix semble s'imposer et doit être ajouté à la relation Stock. Par souci d'uniformité, dans tous les cas, il s'agira d'un prix unitaire.

```
Arme(noProduit, typeArme, poids)
Munition(noProduit, typeMunition)
Armement(typeArme, typeMunition)
DefTypeArme(typeArme)
DefTypeMunition(typeMunition)
Stock(noProduit, nomProduit, quantite, prixUnitaire)

Organisation(idOrganisation, nomOrganisation, adresse, ville, pays, telephone)
Fournisseur(idOrganisation)
Client(idOrganisation)
Reception(idOrganisation, noProduit, date, quantite, prixUnitaire)
Expédition(idOrganisation, noProduit, date, quantite, prixUnitaire)

UnicitéProduit : {p | Arme(p,T,P)} ∩ {p | Munition(p,M)} = ∅
```

En avant vers SQL

Élaboration du schéma

Voir le programme *tp1sol/gaspard_def.sql* écrit par Antoine dans le dossier public du cours IFT 187.

Initialisation du schéma

Voir le programme *tp1sol/gaspard_ini.sql* écrit par Antoine dans le dossier public du cours IFT 187.

Choix des requêtes et des modifications

Gaspard et Antoine conviennent de tester leur schéma de bases de données en programmant les requêtes suivantes :

- R01. Combien y a-t-il d'exemplaires de pistolet à eau en stock ?
- R02. Combien y a-t-il de variétés différentes de pistolets à eau en stock ?
- R03. Quelles sont les armes qui utilisent les mêmes munitions qu'un pistolet à eau ?
- R04. Quelles sont les armes dont le poids est de moins d'une demi-livre (système impérial) ?
- R05. Quelle est la ville dont les fournisseurs livrent le plus d'armes ?
- R06. Quel est le pays dont les clients achètent le plus de munitions ?
- R07. Qui est le plus important fournisseur ?
- R08. Qui est le plus important client ?
- R09. Quelle est la munition la moins chère pour les catapultes à confettis ?
- R10. Quel est le bilan des expéditions de la première semaine d'octobre 2012 ?

et les modifications suivantes :

- M01. Augmenter le prix de tous les produits de 10 %, sauf celui des canons dont l'augmentation est fixée à 18 %.
- M02. Retirer les cuillères à pot et les casseroles de la gamme des produits. Radier ces produits de l'inventaire.

Élaboration des requêtes et des modifications

Antoine s'attaque donc à la première requête. Suivons-le dans sa démarche.

Requête R01 - Combien y a-t-il d'exemplaires de pistolet à eau en stock ?

Remarquons d'abord que 'Pistolet à eau' est un type d'armes. Il s'agit donc de faire la somme des quantités d'une liste de produits d'un type (d'armes) particulier. Supposons que cette liste soit une relation `Liste(noProduit,quantite)`, il nous suffirait d'utiliser la fonction vectorielle SUM ainsi :

```
SELECT SUM(quantite) FROM Liste
```

pour obtenir le résultat désiré. Il ne nous reste plus qu'à déterminer comment construire cette liste. Celle-ci ne peut être construite qu'à partir de deux relations (tables) : Stock (pour les quantités) et Arme (pour déterminer celles parmi les armes qui sont du type 'Pistolet à eau'). En fait, la description de chaque arme du stock peut être complétée par la portion correspondante de la relation Arme. Cela s'appelle une jointure. C'est ici que notre feuille d'équivalence entre l'algèbre relationnelle et SQL peut nous aider :

```
SELECT noProduit, quantite FROM Stock, Arme WHERE Stock.noproduit=Arme.noProduit
```

Antoine remarque aussitôt, parce qu'il teste au fur et à mesure, qu'il y a une erreur. La première référence à noProduit est ambiguë, il faut donc en préciser la provenance. En fait, puisque c'est une jointure, il peut également choisir l'une ou l'autre des tables d'origine.

Certes, voilà une première approximation de la liste, mais elle comporte tous les types d'armes, seules les munitions ont été omises. Il faut donc faire une sélection et ne conserver que les tuples (lignes) dont le type d'armes est 'Pistolet à eau'.

```
SELECT Stock.noProduit, quantite
FROM Stock, Arme
WHERE (Stock.noProduit=Arme.noProduit) AND (typeArme='Pistolet à eau')
```

La solution simple consiste finalement à réunir les deux portions de code ainsi :

```
WITH Liste AS
(
  SELECT Stock.noProduit, quantite
  FROM Stock, Arme
  WHERE (Stock.noProduit=Arme.noProduit) AND (typeArme='Pistolet à eau')
)
SELECT SUM(quantite) FROM Liste
```

Par contre, si on désire simplifier et optimiser, la sommation peut être directement appliquée au résultat de la jointure... et il n'y a plus lieu de retenir la colonne noProduit !

```
SELECT SUM(quantite)
FROM Stock, Arme
WHERE (Stock.noProduit=Arme.noProduit) AND (typeArme='Pistolet à eau')
```

Requête Q02 - Combien y a-t-il de variétés différentes de pistolets à eau en stock ?

En fait, cette requête est plus simple que la précédente ! Il suffit de calculer la cardinalité de la relation Liste. Cette propriété, curieusement, est dénotée COUNT(*) en SQL :

```
SELECT COUNT(*)
FROM Stock, Arme
WHERE (Stock.noProduit=Arme.noProduit) AND (typeArme='Pistolet à eau')
```

Requête Q03 - Quelles sont les armes qui utilisent les mêmes munitions qu'un pistolet à eau ?

Constatons tout d'abord que « utiliser les mêmes munitions » signifie en fait « utiliser les mêmes types de munitions ». Il est facile de déterminer les types de munitions utilisables par sélection sur la table Armement.

```
SELECT typeMunition
FROM Armement
WHERE (typeArme='Pistolet à eau')
```

On pourra ensuite déterminer les autres types d'armes utilisant ces mêmes types de munitions (par jointure de la table Armement avec cette liste).

```

WITH ListeTM AS
(
    SELECT typeMunition
    FROM Armement
    WHERE (typeArme='Pistolet à eau')
)
SELECT typeArme
FROM Armement, ListeTM
WHERE (Armement.typeMunition=ListeTM.typeMunition)
    
```

Si nous conservons ce résultat dans ListeTA, il suffit ensuite de retracer les armes dont le type est dans ListeTA à l'aide des tables Stock et Arme jointes par le numéro de produit.

```

WITH
    ListeTM AS
    (
        SELECT typeMunition
        FROM Armement
        WHERE (typeArme='Pistolet à eau')
    ),
    ListeTA AS
    (
        SELECT typeArme
        FROM Armement, ListeTM
        WHERE (Armement.typeMunition=ListeTM.typeMunition)
    )
SELECT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, ListeTA
WHERE (Stock.noProduit=Arme.noProduit)
    AND (Arme.typeArme=ListeTA.typeArme);
    
```

Cela semble trop simple. D'abord, quel résultat Gaspard désire-t-il : une liste de numéros de produit, une liste de numéros et de noms de produit, une liste encore plus complète ? Faisons l'hypothèse qu'une liste de numéros et de noms de produits est suffisante. Nous aurons soin toutefois d'ajouter le mode **DISTINCT** puisque nous savons que SQL manipule des tables (collections) et non des relations (ensembles) et que la requête, telle que formulée, pourrait produire des doublons.

```

WITH
    ListeTM AS
    (
        SELECT DISTINCT typeMunition
        FROM Armement
        WHERE (typeArme='Pistolet à eau')
    ),
    ListeTA AS
    (
        SELECT DISTINCT typeArme
        FROM Armement, ListeTM
        WHERE (Armement.typeMunition=ListeTM.typeMunition)
    )
SELECT DISTINCT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, ListeTA
WHERE (Stock.noProduit=Arme.noProduit)
    AND (Arme.typeArme=ListeTA.typeArme);
    
```

Nous pouvons ensuite nous demander si notre requête est optimale. Après examen, nous pouvons proposer ceci :

```
SELECT DISTINCT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, Armement, Armement AS Liste
WHERE (Stock.noProduit=Arme.noProduit)
      AND (Arme.typeArme=Armement.typeArme)
      AND (Armement.typeMunition=Liste.typeMunition)
      AND (Liste.typeArme='Pistolet à eau')
```

La vraie question à se poser est probablement celle-ci : qu'est-ce que Gaspard a en tête quand il écrit « les armes qui utilisent les mêmes munitions qu'un pistolet à eau » ? Pour se qualifier, une arme doit-elle utiliser exactement les mêmes munitions qu'un pistolet à eau ? Est-il suffisant qu'elle utilise un sous-ensemble de ces munitions d'un pistolet à eau ? Ou encore, est-il suffisant que certaines de ses munitions soient parmi celles d'un pistolet à eau tout en permettant que d'autres munitions ne soient pas les mêmes ? C'est toute la différence qui existe entre l'égalité, l'inclusion et l'intersection. En attendant, la réponse de Gaspard, voyons si nous pouvons répondre à chacune de ces trois requêtes.

Constatons d'abord que notre requête, telle que rédigée, traite de la variante intersection. Examinons maintenant comment rédiger la variante égalité.

```
WITH
  ListeTM AS
  (
    SELECT typeMunition
    FROM Armement
    WHERE (typeArme='Pistolet à eau')
  ),
  ListeTA AS
  (
    SELECT typeArme
    FROM Armement
    WHERE (ListeTM = Liste_des_types_de_munitions_de_typeArme)
  )
SELECT DISTINCT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, ListeTA
WHERE (Stock.noProduit=Arme.noProduit)
      AND (Arme.typeArme=ListeTA.typeArme);
```

Or, nous savons déjà comment calculer la *Liste_des_types_de_munitions* d'une arme donnée, sur le modèle du calcul de ListeTM. On remarque également qu'il suffit de remplacer l'opérateur d'égalité (=) par l'opérateur d'inclusion (\subseteq) pour obtenir la variante inclusion :

```
(ListeTM  $\subseteq$  Liste_des_types_de_munitions_de_typeArme)
```

Nous nous croyons heureux de pouvoir proposer ceci :

```

WITH
  ListeTM AS
  (
    SELECT DISTINCT typeMunition
    FROM Armement
    WHERE (typeArme='Pistolet à eau')
  ),
  ListeTA AS
  (
    SELECT DISTINCT typeArme
    FROM Armement AS A1
    WHERE
      (
        ListeTM
        =
        ( -- Liste_des_types_de_munitions_de_typeArme
          SELECT DISTINCT typeMunition
          FROM Armement AS A2
          WHERE (A1.typeArme=A2.typeArme)
        )
      )
  )
SELECT DISTINCT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, ListeTA
WHERE (Stock.noProduit=Arme.noProduit)
AND (Arme.typeArme=ListeTA.typeArme);
    
```

Nous déchantons rapidement : en SQL, il n'y a pas d'opérateur d'égalité ni d'opérateur d'inclusion entre les tables ! Nous ne nous décourageons pas. Révisons plutôt nos identités ensemblistes :

Soit deux ensembles A et B,
si $(A-B = \emptyset)$ alors $A \subseteq B$
si $(A-B = \emptyset)$ et $(B-A = \emptyset)$ alors $A = B$

Pour tester $A-B = \emptyset$, il suffit d'écrire

```
0 = SELECT count(*) FROM (SELECT * FROM A MINUS SELECT * FROM B) AS X
```

et d'inverser A et B pour tester $B-A = \emptyset$.

```
0 = SELECT count(*) FROM (SELECT * FROM B MINUS SELECT * FROM A) AS X
```

Dans notre cas, A est la ListeTM et B est la *Liste_des_types_de_munitions_de_typeArme*. Puisque c'est plus simple, commençons par l'inclusion :

```

WITH
  ListeTM AS
  (
    SELECT DISTINCT typeMunition
    FROM Armement
    WHERE (typeArme='Pistolet à eau')
  ),
  ListeTA AS
  (
    SELECT DISTINCT typeArme
    FROM Armement A1
    WHERE -- (ListeTM = Liste_des_types_de_munitions_de_typeArme)
      (
        0
        =
        (
          SELECT count(*) FROM
            (
              SELECT * FROM ListeTM
            EXCEPT
              -- Liste_des_types_de_munitions_de_typeArme
              SELECT typeMunition
              FROM Armement A2
              WHERE (A1.typeArme=A2.typeArme)
            ) X
        )
      )
  )
SELECT DISTINCT Stock.noProduit, Stock.nomProduit
FROM Stock, Arme, ListeTA
WHERE (Stock.noProduit=Arme.noProduit)
AND (Arme.typeArme=ListeTA.typeArme)
    
```

Antoine programme cette requête et constate avec satisfaction qu'elle livre les bons résultats. Par contre, Madeleine exige qu'il fournisse sa solution sous Oracle. Antoine adapte donc sa requête pour s'adapter à ce dialecte. Il change l'opérateur EXCEPT en MINUS. Il teste... catastrophe ! Les résultats sont faux ! Pris de panique, il téléphone à Madeleine. Celle-ci, avec un calme olympien, lui répond : « C'est vrai, il y a des erreurs dans le calcul des énoncés WITH sous la version 10g, il suffit de passer à la version 11g »¹⁵. Le problème, c'est que Gaspard ne peut s'offrir la version 11g. Il réussit tout de même à faire des tests grâce à une copine stagiaire (gentille la copine). Malheur ! Ça ne compile même plus ! Du coup, Antoine abandonne la variante égalité et songe sérieusement à devenir lui-même enseignant... il pourra ainsi refiler le travail à quelqu'un d'autre.

Nous en retenons les leçons suivantes :

- ◇ Ne jamais optimiser une requête avant de l'avoir terminée¹⁶.
- ◇ Ce n'est pas parce qu'on le paye cher qu'un logiciel est sans erreur.
- ◇ Ce n'est pas parce qu'on a payé un logiciel qu'on peut obtenir de faire réparer l'erreur (sans frais supplémentaires).

Requête Q04 - Quelles sont les armes dont le poids est de moins d'une demi-livre (système impérial) ?

RAS

¹⁵ En fait, la situation est pire que cela, la version 10g release 1 détecte une erreur de syntaxe là où il n'y en a pas, la version 10g release 2, calcule un mauvais résultat et la version 11g ne traite pas les WITH imbriqués. Oracle est le plus important éditeur de logiciels de bases de données au monde...

¹⁶ Selon certains, les trois règles de l'optimisation sont les suivantes : [1] ne considérer l'optimisation que si le programme est sans erreur [2] ne considérer l'optimisation que si elle n'introduit aucune erreur et ne change aucun résultat [3] ne considérer l'optimisation que si le gain fait la différence entre une applicable utilisable et une application non utilisable.

Requête Q05 - Quelle est la ville dont les fournisseurs livrent le plus d'armes ?

RAS

Requête Q06 - Quel est le pays dont les clients achètent le plus de munitions ?

RAS

Requête Q07 - Qui est le plus important fournisseur en termes de prix d'achat ?

Il ne s'agit d'un client qui est à la fois fournisseur. « Vente » et « Achat » changent de signification selon l'organisation qu'on prend en référence. Gaspard parlait-il en faisant référence à lui-même ou au fournisseur ? Voilà une ambiguïté à laquelle il faut faire attention. Je vous propose donc la reformulation suivante :

Qui est le plus important fournisseur en termes de volume d'achat (en valeur totale des produits achetés chez celui-ci) ?

Requête Q08 - Qui est le plus important client en termes de nombre d'achats ?

RAS

Requête Q09 - Quelle est la munition la moins couteuse pour les catapultes à confettis ?

RAS

Requête Q10 - Combien y a-t-il eu de ventes au cours de la première semaine d'octobre 2012 ?

RAS

Retour à modélisation

De l'importance des noms (ou le renommage de la table « Stock »)

— Antoine : Je n'aime pas la modélisation Arme-Munition-Stock, cela manque de flexibilité.

— Madeleine : Où vois-tu un manque de flexibilité ?

— Antoine : Il y a trop de couplage.

— Madeleine : Il y a un seul couplage par variété de produits. De plus, aucun de ces couplages n'a d'incidence sur l'autre. Ainsi, je¹⁷ peux ajouter, modifier ou retirer un nouveau produit sans impact ni sur Stock ni sur les autres types d'armes.

— Antoine : Mais alors, tu ne peux pas ajouter une arme ou une munition qui n'existe pas en Stock.

— Madeleine : J'espère bien ! J'ajoute un produit d'abord et en même temps j'ajoute les informations spécifiques à son type dans la table appropriée. C'est pour ça que le nom de la table est si important ; l'opération dont il s'agit est « ajouter un produit », pas « ajouter un stock ». Si je dois ajouter un stock (suite à une réception), je le fais pour un produit déjà existant et ça ne touche que l'attribut quantité de la table Stock... Par ailleurs, cette même structure permettrait aussi d'ajouter un produit dont je ne connais pas encore le type, et ce, sans incohérence.

— Antoine : Alors, si la table décrit le produit, pourquoi l'avoir appelée Stock ?

¹⁷ Un observateur externe aura remarqué la « personnalisation » de la discussion, Madeleine et Antoine se plaçant au centre du processus, en lieu et place des acteurs naturels (Gaspard, son commis, les fournisseurs, les clients, etc.) ou même du « système informatisé » dont la base de données est l'élément central. Ce procédé, bien que fréquent, devrait être évité pour plusieurs raisons dont le risque de perte de perspective. Le cours IGL 301 abordera cette question.

— Madeleine : C'est une erreur de parcours. Parfois, en tentant de modéliser un concept (les « stocks ») on réalise qu'il s'agit d'une caractéristique (attribut) d'une entité plus importante (relation) et non une entité en elle-même. Dans ce cas, il ne faut pas hésiter à renommer les concepts correctement, sans quoi on risque de continuer à mal concevoir le modèle, ou, à tout le moins, à le mal percevoir et à le mal expliquer.

Antoine se dit en lui-même que Madeleine aurait bien pu faire ce changement avant de lui transmettre la modélisation plutôt que de lui faire la leçon après qu'il se soit épuisé à comprendre ce modèle bancal.

Par la fenêtre, il aperçoit un curieux bonhomme tout maigre, un bouquet de fleurs à la main. Ce que voyant, Madeleine soupire « Jacques, encore et toujours ! ».

Des contraintes sur les quantités

— Antoine : « Je voulais spécifier que l'attribut quantité dans la relation Vente soit inférieure à celui de la relation Stock. Je pense que je n'ai pas bien défini les choses, j'ai ajouté le code suivant :

```
CONSTRAINT Vente_quantite CHECK(quantité<=Stock.quantite)
```

et ça ne semble pas fonctionner.... Ah ! Je sais ! Il faut plutôt faire un SELECT dans le CHECK et ... »

— Madeleine : « Stop ! Il y a deux problèmes distincts :

(a) syntaxique : puisque tu mets en relation deux tables (autrement que par une clé étrangère¹⁸), il faudrait créer une contrainte de BD (et non une contrainte de table), donc un CREATE ASSERTION en dehors du CREATE TABLE.

(b) sémantique : cette contrainte est fautive ; la vente doit être évaluée en fonction du stock courant et ne plus être évaluée ensuite (puisque la vente doit avoir pour effet de réduire le stock d'autant).

Le problème (b) appelle (au moins) deux solutions distinctes.

Solution 1

Il faut « dématérialiser » l'attribut quantité de la table Stock et le recalculer à chaque consultation sur la base des livraisons et des ventes chronologiquement antérieures (correct, mais pas performant du tout).

Solution 2

Il faut conserver le stock courant, mais placer une contrainte évaluable au seul moment de l'insertion d'un tuple dans la relation Vente — mécanisme qui peut être mis en oeuvre de plusieurs façons : actions associées aux clés étrangères, automatismes généraux de la base de données (*trigger*), etc.

Dans les deux solutions toutefois, il faut ajouter une contrainte importante que nous avons négligée jusqu'à présent : les dates de Livraison et les dates de Vente doivent être antérieures au moment courant et postérieures à la dernière transaction enregistrée (une transaction, c'est une livraison ou une vente). Plus précisément, si nous ne respectons pas ces contraintes, nous pourrions nous retrouver à devoir refuser rétroactivement une transaction... ne jouons pas trop avec le continuum espace-temps, laissons cela aux physiciens... pour le moment ! »

Le silence s'installe. Antoine est perplexe, il a compris la notion d'assertion et le problème de performance. Il se demande toutefois si les observations de Madeleine relativement au « continuum espace-temps » ne sont pas une résultante de sa consommation d'absinthe.

¹⁸ Plusieurs experts estiment en fait que les contraintes de clé étrangère ne devraient pas faire partie de la définition des tables, mais être à l'extérieur de celles-ci – précisément pour cette raison.

— Madeleine : De toute façon, Gaspard, en vrai commerçant, confond Achat, Réception et Décaissement d'une part et Vente, Livraison et Encaissement d'autre part. Si au moins, il acceptait d'engager un comptable, ça nous faciliterait le travail !

Antoine se dit que, décidément, ce modèle est vraiment tordu.

De l'importance de distinguer la donnée de sa représentation

Il y a encore un élément incorrectement modélisé, ce sont les clés « noProduit-noArme-noMunition ». La solution consiste à dissocier la clé interne du numéro de produit utilisé par Gaspard. Ainsi l'attribut noProduit deviendrait une simple clé interne (comme idOrganisation) et les numéros seraient alors gérés indépendamment par des attributs dans leurs tables respectives. Le nouveau noProduit devient une clé interne « artificielle » et pour cette raison change de nom pour « idProduit ».

C'est le moment de rappeler la règle de Lavoie : « À peu près tout ce qu'on pense être unique peut ne plus l'être (et inversement) ». Il en découle qu'il est souvent plus prudent en pratique d'ajouter une clé (candidate) artificielle à une relation même s'il y existe des clés (candidates) naturelles (autres que l'ensemble de tous les attributs). Ceci facilite considérablement les modifications requises lorsqu'une clé cesse de l'être suite à l'évolution des conditions du monde réel.

Encore une fois !

Il est temps de revenir à la table à dessein pour :

- ◇ corriger le choix des noms des relations : Stock -> Produit, Achat -> Livraison, Livraison -> Réception¹⁹;
- ◇ rationaliser l'utilisation des clés « noProduit-noArme-noMunition » ;
- ◇ factoriser les éléments communs de Livraison et Réception dans une nouvelle relation Expedition ;
- ◇ distinguer le prix de vente, le prix d'achat et le prix de référence²⁰ ;
- ◇ indiquer le nombre de cartouches ou d'unités dans un paquet de munitions ;
- ◇ distinguer la date de départ d'une expédition de sa date d'arrivée.

Ne nous décourageons pas :

*Hâtez-vous lentement, et sans perdre courage,
Vingt fois sur le métier remettez votre ouvrage,
Polissez-le sans cesse, et le repolissez,
Ajoutez quelquefois, et souvent effacez.
(Nicolas Boileau, L'art poétique, Chant I)*

Voici donc un nouveau schéma :

¹⁹ Nous avons abondamment expliqué pourquoi Stock était en fait Produit. Les deux autres changements sont rendus nécessaires pour les raisons suivantes : [1] la nomenclature devrait refléter un point et un seul celui du propriétaire de l'entrepôt (Gaspard vend à un client et achète à un fournisseur) ; [2] les achats et les ventes ne font pas varier les quantités en stock, ce sont les réceptions (provenant du fournisseur) et les livraisons (faites au client) : un même achat peut faire l'objet de plusieurs réceptions à des moments différents, une même vente peut être acheminé en plusieurs livraisons.

²⁰ Souvent désigné au Québec sous le nom de « prix de liste » par calque de l'anglais.

```

Arme(idProduit, noArme, typeArme, poids)
    contrainte cle Arme(idProduit)
    contrainte cle Arme(noArme)
    contrainte Arme(idProduit) ref Produit(idProduit)
Munition(idProduit, noMunition, typeMunition, nb)
    contrainte cle Munition(idProduit)
    contrainte cle Munition(noMunition)
    contrainte Munition(idProduit) ref Produit(idProduit)
Armement(typeArme, typeMunition)
    contrainte cle (typeArme, typeMunition)
    contrainte Armement(typeArme) ref DefTypeArme(typeArme)
    contrainte Armement(typeMunition) ref DefTypeArme(typeMunition)
DefTypeArme(typeArme)
    contrainte cle DefTypeArme(typeArme)
DefTypeMunition(typeMunition)
    contrainte cle DefTypeMunition(typeMunition)
Produit(idProduit, nomProduit, quantiteEnStock, prixUnitaireDeRef)
    contrainte cle Produit(idProduit)
Organisation(idOrganisation, nomOrganisation, adresse, telephone)
    contrainte cle Organisation(idOrganisation)
Fournisseur(idOrganisation, noFournisseur)
    contrainte cle Fournisseur(idOrganisation)
    contrainte cle Fournisseur(noFournisseur)
    contrainte Fournisseur(idOrganisation) ref Organisation(idOrganisation)
Client(idOrganisation, noClient)
    contrainte cle Client(idOrganisation)
    contrainte cle Client(noClient)
    contrainte Client(idOrganisation) ref Organisation(idOrganisation)
Expedition(idExpedition, idProduit, dateDepart, dateArrivee, quantite)
    contrainte cle Expedition(idExpedition)
    contrainte Expedition(idProduit ref Produit(idProduit))
Livraison(idExpedition, idOrganisation, prixVente)
    contrainte cle Livraison(idExpedition)
    contrainte Livraison(idExpedition) ref Expedition(idExpedition)
    contrainte Livraison(idOrganisation) ref Client(idOrganisation)
Reception(idExpedition, idOrganisation, prixAchat)
    contrainte cle Reception(idExpedition)
    contrainte Reception(idExpedition) ref Expedition(idExpedition)
    contrainte Reception(idOrganisation) ref Fournisseur(idOrganisation)
    
```

Il est très tentant de fusionner Expedition, Livraison et Reception :

```

Expedition(idExpedition, noProduit, dateDepart, dateArrivee, quantite, prixTotal,
typeExp, idOrganisation)
    
```

où typeExp indique s'il s'agit d'une livraison ou d'une réception. Toutefois, il faut ajouter une contrainte externe pour s'assurer que l'organisation est un client, si le type est une livraison, et un fournisseur, si le type est une réception.

Nous atteignons ici la limite d'expressivité des relations simples utilisées jusqu'à présent. Nous nous rendons compte que plusieurs contraintes demeurent inexprimées, plusieurs éléments de la réalité sont mal représentés. Sans l'apport des contraintes, le modèle demeure trop incomplet pour permettre un échange constructif avec Gaspard (en particulier dans le contexte d'une validation). Nous poursuivrons notre exercice de modélisation relationnelle plus tard.

²¹ Remarquons que plusieurs auteurs utilisent les termes types et domaines exactement dans le sens inverse (Codd, Strachey et Cardelli, notamment). Nous ne prétendons pas trancher ce débat ici. Nous constatons toutefois la nécessité de préciser le vocabulaire utilisé dans notre document et avons choisi de nous conformer à l'usage de SQL espérant ainsi réduire la confusion... au sein de notre document.

²² Opérateur ou opération ? La distinction est la suivante : une opération est une fonction et l'opérateur une de ses représentations syntaxiques. Par exemple, l'opération de cardinalité sur un ensemble et l'opérateur #.

A. Comment bien consommer un joint

On peut classer les jointures en deux catégories :

- ◇ les jointures classiques, issues de l'algèbre relationnelle et dites jointures internes (*inner join*) ;
- ◇ les jointures propres au langage SQL, découlant de son modèle de traitement de l'annulabilité et dites jointures externes (*outer join*).

Jointures internes

La forme la plus générale de jointure interne est dénotée par :

```
R JOIN S ON <condition>
```

qui est par ailleurs équivalente à l'énoncé SELECT suivant :

```
SELECT * FROM R, S WHERE <condition>
```

L'opération consiste donc à joindre (réunir) tout tuple de R avec tout tuple de S pour lesquels la condition est satisfaite, et à placer le nouveau tuple ainsi formé dans le résultat de la jointure. Le résultat comportera donc entre 0 et n tuples, où n est le produit des cardinalités des deux relations.

La notation

```
R JOIN S USING (a1, ..., an)
```

appelée équijointure est une dénotation abrégée équivalente :

```
R JOIN S ON (R.a1=S.a1 AND ... AND R.an=S.an)
```

À une différence près cependant, pour chaque paire de colonnes identifiées, une seule est conservée dans le résultat. Par exemple, si les entêtes des relations R et S sont les suivantes :

```
R (id, u, v)
S (id, x, y)
```

l'en-tête de la relation dénotée par :

```
R JOIN S USING (id) AS T
```

est

```
T (id, u, v, x, y)
```

alors que celle de :

```
R JOIN S on (R.id=S.id) AS T
```

est

```
T (id, u, v, id_1, x, y)
```

où *id_1* est un identificateur distinct des autres engendrés selon la recette secrète de l'éditeur du SGBD.

On peut encore « raccourcir » l'expression d'une jointure, si on désire faire une équijointure sur exactement tous les attributs de mêmes noms de R et de S :

```
T NATURAL JOIN S
```

est équivalent à :

```
T JOIN S USING (b1, ... bn)
```

où (b1, ..., bn) est la liste des **tous** les attributs de mêmes noms appartenant à R et à S.

Cette dernière dénotation est toutefois **fortement** déconseillée, car elle peut conduire à des erreurs « silencieuses » lorsque des attributs sont ajoutés, retirés ou renommés dans les tables. Or c'est une

situation fréquente en pratique. C'est pourquoi cette dénotation est même strictement interdite dans la plupart des standards de programmation SQL.

Pour terminer, on remarque que les dénotations suivantes sont équivalentes :

- `R JOIN S ON true`
- `SELECT * FROM R, S WHERE true`
- `SELECT * FROM R, S`

et correspondent en fait au produit cartésien auquel, par souci de complétude, une notation spécifique a été prévue :

`R CROSS JOIN S`

Note 1

Qu'arrive-t-il si l'un des attributs de l'équijointure est nul ?

La réponse est donnée par le raisonnement suivant :

- ◇ La comparaison de deux termes dont l'un (au moins) est nul donne le résultat `unknown`.
- ◇ La conjonction de termes dont au moins un `unknown` est `unknown` également.
- ◇ Une clause `where` dont la condition est `unknown` n'est pas satisfaite²³.
- ◇ Les deux tuples soumis à examen ne seront donc pas joints.

Par ailleurs, les autres attributs nuls (ceux qui ne sont pas parties prenantes à la condition de jointure) sont préservés. Un raisonnement analogue s'applique à la jointure générale.

Jointures externes

La jointure externe se veut un outil pour pallier la perte des tuples décrite dans la note 1. Bien sûr, cela soulève une question difficile : Quel tuple doit-on choisir pour le joindre à un tuple dont l'un des attributs de jointure est nul ?

Les concepteurs de SQL ont proposé la réponse suivante : le tuple (dégénéré) dont tous les attributs sont nuls ! Cette réponse n'est pas plus satisfaisante que bien d'autres. Au moins a-t-elle le mérite d'être simple, en apparence. Il y a donc trois jointures externes : gauche (LEFT), droite (RIGHT), totale (FULL). La jointure gauche préserve tous les tuples du terme de gauche, la jointure droite préserve tous les tuples du terme de droite, la jointure totale préserve tous les tuples des deux termes.

Relativement au tuple dégénéré, plusieurs questions se posent, dont celles-ci :

- ◇ Que se passe-t-il quand un des attributs du tuple dégénéré est non annulable ?
- ◇ N'aurait-on pas mieux fait de prendre la valeur par défaut pour certains des attributs du tuple dégénéré ?
- ◇ Les colonnes des attributs de jointure et de l'équijointure ne sont plus identiques si on joint un tuple dégénéré, lesquelles seront retirées ?

Pour ces raisons, et plusieurs autres, l'utilisation des jointures externes est le plus souvent problématique et les programmeurs expérimentés ont tendance à en réduire l'utilisation le plus possible.

Finalement, il existe également une forme « naturelle » aux jointures externes, mais, pour les mêmes raisons que celles mentionnées lors de la présentation des jointures internes, il est préférable de ne pas les utiliser.

Note 2

Les mots-clés `INNER` et `OUTER` ont été délibérément omis de la présentation, car ils sont totalement redondants sémantiquement.

²³ Rappelons que c'est le contraire dans le cas d'une contrainte `CHECK`. SQL n'en est malheureusement pas à une incohérence près.

Un exemple simple

```
TABLE A
ID V
-- -
1 a
2 b
3 c
  d
```

```
TABLE B
ID W
-- -
1 g
2 h
2 i
5 j
  k
```

```
SELECT * FROM A, B
SELECT * FROM A CROSS JOIN B
ID V ID W
-- - - - -
1 a 1 g
1 a 2 h
1 a 2 i
1 a 5 j
1 a  k
2 b 1 g
2 b 2 h
2 b 2 i
2 b 5 j
2 b  k
3 c 1 g
3 c 2 h
3 c 2 i
3 c 5 j
3 c  k
  d 1 g
  d 2 h
  d 2 i
  d 5 j
  d  k
```

```
SELECT * FROM A JOIN B ON (A.ID=B.ID)
ID V ID W
-- - - - -
1 a 1 g
2 b 2 h
2 b 2 i
```

```
SELECT * FROM A JOIN B USING (ID)
ID V W
-- - -
1 a g
2 b h
2 b i
```

```
SELECT * FROM A LEFT JOIN B USING (ID)
ID V W
-- - -
1 a g
2 b h
2 b i
  d
3 c
```

```
SELECT * FROM A RIGHT JOIN B USING (ID)
ID V W
-- - -
1 a g
2 b i
2 b h
  k
5 j
```

```
SELECT * FROM A FULL JOIN B USING (ID)
ID V W
-- - -
1 a g
2 b h
2 b i
  d
3 c
  k
5 j
```

```
SELECT * FROM A LEFT JOIN A "X"
      ON (A.ID=X.ID AND A.v=X.v)
ID V ID V
-- - - - -
1 a 1 a
2 b 2 b
3 c 3 c
  d
```

```
SELECT * FROM A LEFT JOIN A "X" USING (ID, v)
ID V
-- -
1 a
2 b
3 c
  d
```

B. Les colles du professeur Tournesol

Qui sont Bertrand, Henri, René, Kurt, Blaise et Alan ?

• gnirunT nAlA te lacsaF
esialB , ledëoc truk , setracsed èneh , èranciop ìrneH , lessurH dnartreb

On remarque que la fonction $t(a)$ ne répond pas adéquatement à la question « Quels sont les fournisseurs d'armes de type a ? ». Pourquoi ?

• etxètnoc ec snad tnerèlucitrap
sup te « russinruof » noissèrpxe l rap drapsag dnèthè'ud
noitsèuq al à ènèma suon ìcec .sup non èèrèdisnoc sap tiarès en selcitra
sertua d ruop tneruèqinu lepna tiarf tiarua drapsag lèuqua sïam , drapsag
à elcitra l rìnruf tnavuop noitasinagro ènu sulp ed .srnessinruof
sed ètsil al snad sap tiarèrugiif en èrvil èrocne sap tiarua n iud
sïam , sïof èrèimèrp al ruop èdnamoc tiarua drapsag iud zèhc noitasinagro
ènu cnod .drapsag à selcitra sèmèm sec èrvil àjèd tnya noitasinagro
sel tötulp tse'c , icl .drapsag à selcitra sed un nu (rìnruf) èrdnev
tnavup snoitasinagro sel tneruègised srnessinruof sel lènsu sèns un

À quel écrit de Boris Vian fait-on référence ?

• « » seunqimota sebomob sed avaj al « èlultitni nosnàhc al

Section 3.2

Madeleine va donc travailler avec Gaspard afin de modéliser son entrepôt (ou plutôt le contenu de celui-ci). Pourquoi cette nuance est-elle importante ?

• noitcudortni ènu tse 103 LGI tnod lèicigol
èinèg ud ènïamod , secnègixè sed èirèinègni ne èèdroba tse noitsèuq
èttèc .èlìciffid tnevus tse resillèdom sap en iouq rïovas .)secnatsid
sed in ueil ud èigolopot al sap tïannoc en no (èdnamoc nu d èttèllicuc
al ruop lamitpo srucrap ed noitcnof ènu rètuoja d èlbiissop sap arès en
li unev tnom el , edoc nu'ud tse n ec is , èrtnoc rap .neib .èuqitnamès
ed rehcatra y à sap a n èlèdom èrtoc enq te « edoc » nu tnerèlpmis
tse tnerècalpme nu'ud rìnevnoc tiarruop n0 ? tnerètib na ècnerèfèr sap
lì-tïaf en « tnerècalpme » nu r0 ? elcitra èuqahc røkocots uo tnerècalpme l
èruncil d tnerèitrep l i-tïarès , dnarè zèssa tse tòpèrthè l is .èuqioù
• èmèm-iul tnerètib ud tøjus na snoitseuq ed resop sap èrisèd en drapsag
èuq ècrap èlèdom na ècàrg èrìcèd el ed tnerèitrep sap tse n li .drapsag
ed snoitinun sel te sèma sel tnerèitnoc iud tnerètib el tse tòpèrthè l

...

Annexe A

Initialement, seule la jointure naturelle a été définie. Montrer qu'elle est suffisante en définissant les autres jointures internes en fonction de celle-ci.

C. Madeleine, Gaspard et l'absinthe

Ce soir j'attends Madeleine
 J'ai apporté du lilas
 J'en apporte toutes les semaines
 Madeleine elle aime bien ça
 Ce soir j'attends Madeleine
 On prendra le tram trente-trois
 Pour manger des frites chez Eugène
 Madeleine elle aime tant ça
 Madeleine c'est mon Noël
 C'est mon Amérique à moi
 Même qu'elle est trop bien pour moi
 Comme dit son cousin Joël
 Mais ce soir j'attends Madeleine
 On ira au cinéma
 Je lui dirai des « je t'aime »
 Madeleine elle aime tant ça

Elle est tellement jolie
 Elle est tellement tout ça
 Elle est toute ma vie
 Madeleine que j'attends là

Ce soir j'attends Madeleine
 Mais il pleut sur mes lilas
 Il pleut comme toutes les semaines
 Et Madeleine n'arrive pas
 Ce soir j'attends Madeleine
 C'est trop tard pour le tram trente-trois
 Trop tard pour les frites d'Eugène
 Madeleine n'arrive pas
 Madeleine c'est mon horizon
 C'est mon Amérique à moi
 Même qu'elle est trop bien pour moi
 Comme dit son cousin Gaston
 Mais ce soir j'attends Madeleine
 Il me reste le cinéma
 Je pourrai lui dire des « je t'aime »
 Madeleine elle aime tant ça

Elle est tellement jolie
 Elle est tellement tout ça
 Elle est toute ma vie
 Madeleine qui n'arrive pas

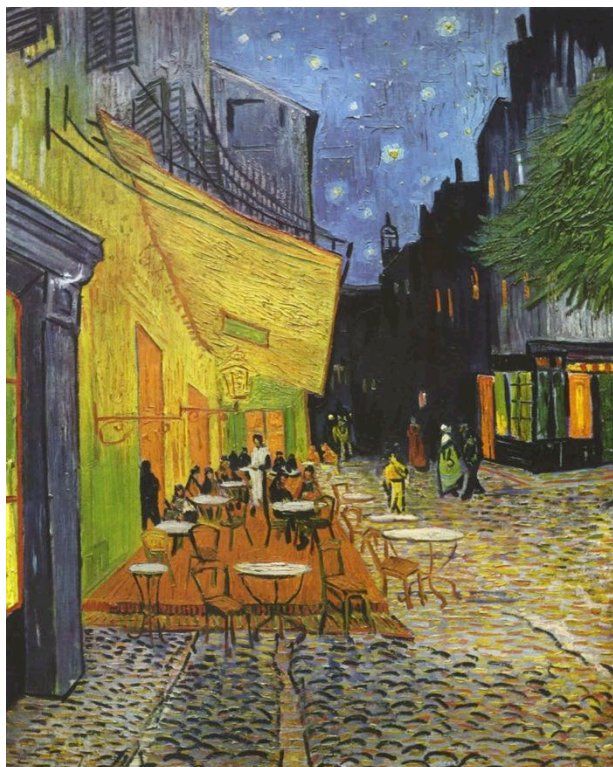
Ce soir j'attendais Madeleine
 Mais j'ai jeté mes lilas
 Je les ai jetés comme toutes les semaines
 Madeleine ne viendra pas
 Ce soir j'attendais Madeleine
 C'est fichu pour le cinéma
 Je reste avec mes « je t'aime »
 Madeleine ne viendra pas
 Madeleine c'est mon espoir
 C'est mon Amérique à moi
 Mais sûr qu'elle est trop bien pour moi
 Comme dit son cousin Gaspard
 Ce soir j'attendais Madeleine
 Tiens le dernier tram s'en va
 On doit fermer chez Eugène
 Madeleine ne viendra pas

Elle est elle est pourtant tellement jolie
 Elle est pourtant tellement tout ça
 Elle est pourtant toute ma vie
 Madeleine qui ne viendra pas

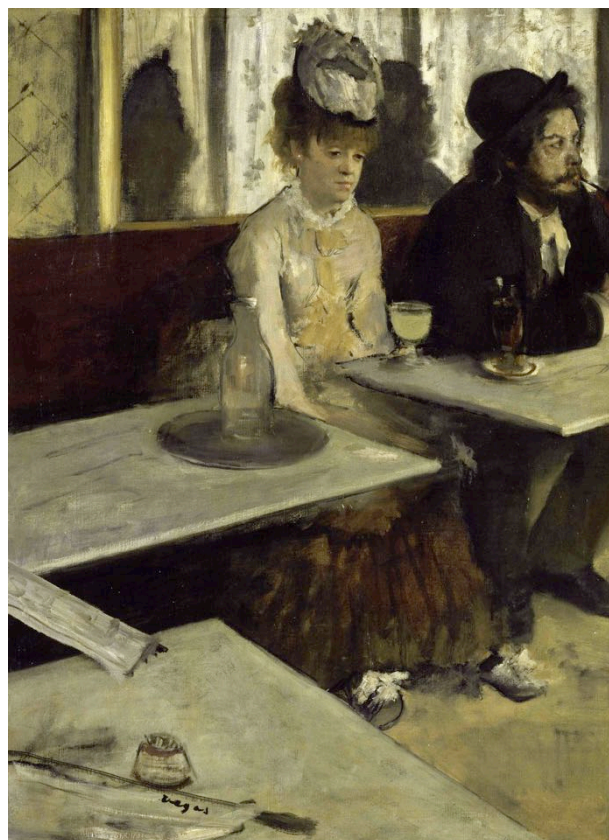
Mais demain j'attendrai Madeleine
 Je rapporterai du lilas
 J'en rapporterai toute la semaine
 Madeleine elle aimera ça
 Demain j'attendrai Madeleine
 On prendra le tram trente-trois
 Pour manger des frites chez Eugène
 Madeleine elle aimera ça
 Madeleine c'est mon espoir
 C'est mon Amérique à moi
 Tant pis si elle est trop bien pour moi
 Comme dit son cousin Gaspard
 Demain j'attendrai Madeleine
 On ira au cinéma
 Je lui dirai des « je t'aime »
 Et Madeleine elle aimera ça.

Jacques Brel
 1961

tiré de *Jacques Brel – oeuvre intégrale*
 Éditions Robert Lafont, Paris, 1982 ; pp 202-204



« Terrasse d'un café le soir »
Vincent Van Gogh (1853-1890)²⁴



« Buveur d'absinthe »
Edgar Degas (1834-1917)²⁵

²⁴ Source consultée le 2012-10-05 :
http://fr.wikipedia.org/wiki/Fichier:Vincent_Willem_van_Gogh_015.jpg

²⁵ Source consultée le 2012-10-05 :
[http://fr.wikipedia.org/wiki/Fichier:EDGAR_DEGAS_-_En_un_café_o_La_Absenta_\(Museo_de_Orsay,_Par%C3%ADs,_1873._Óleo_sobre_lienzo,_92_x_68.5_cm\).jpg](http://fr.wikipedia.org/wiki/Fichier:EDGAR_DEGAS_-_En_un_café_o_La_Absenta_(Museo_de_Orsay,_Par%C3%ADs,_1873._Óleo_sobre_lienzo,_92_x_68.5_cm).jpg)