

Date de dernière modification : 30/6/2015

Annexe 19

Conception logique d'une base de données

Dans cette annexe, on complète les techniques de traduction en traitant celle des attributs objets. On y trouve aussi une collection d'exercices accompagnés, pour la plupart, d'une suggestion de solution.

A19.1 REPRÉSENTATION D'UN ATTRIBUT OBJET

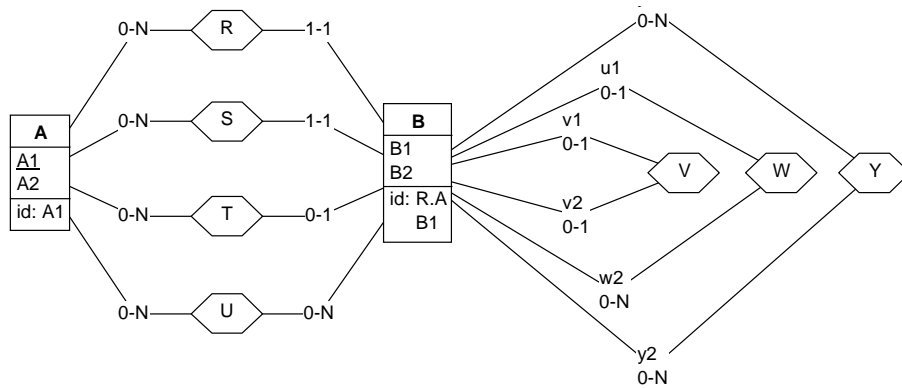
<complément de la section 19.3>

Ces attributs ont été surtout utilisés pour référencer un type d'entités générique jouant, par exemple, le rôle de dimension (section A16.1, Annexe 16). On les représentera par des clés étrangères vers la table représentant le type d'entité générique.

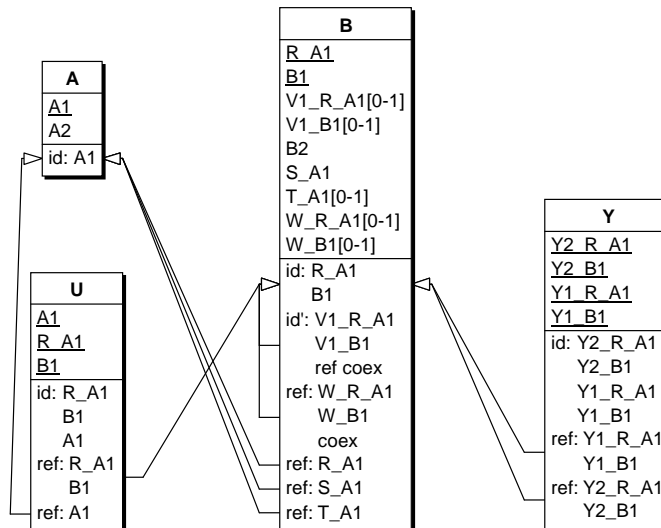
A19.2 EXERCICES DU CHAPITRE 19

A19.1 Reprendre les exercices du chapitre 13 et les résoudre en appliquant le plan de transformation développé dans ce chapitre. Pour ce qui est de l'exercice 13.2, on ajoutera la propriété suivante au schéma conceptuel : *le cheval senior d'une écurie appartient à celle-ci.*

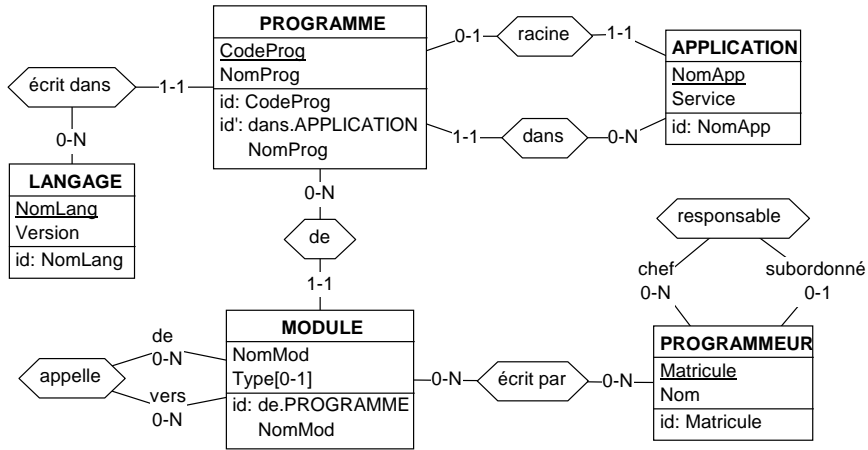
A19.2 Traduire le schéma conceptuel ci-dessous en structure de tables.



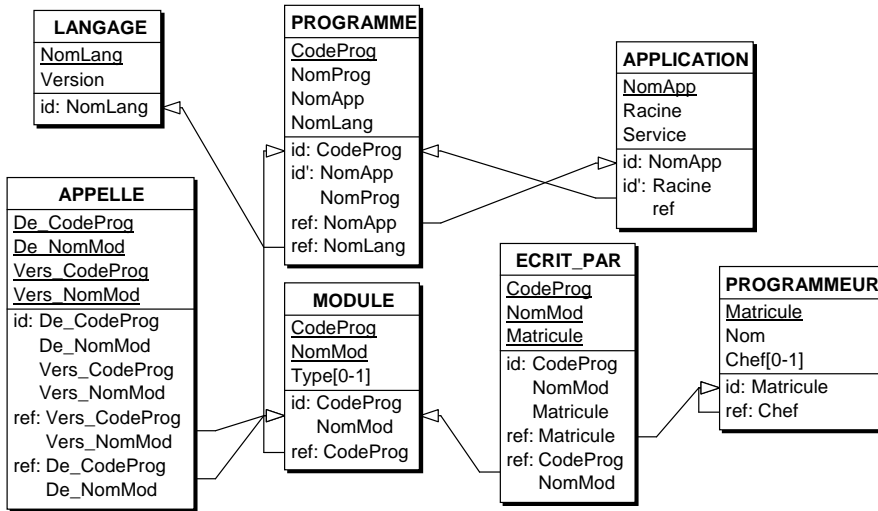
Solution



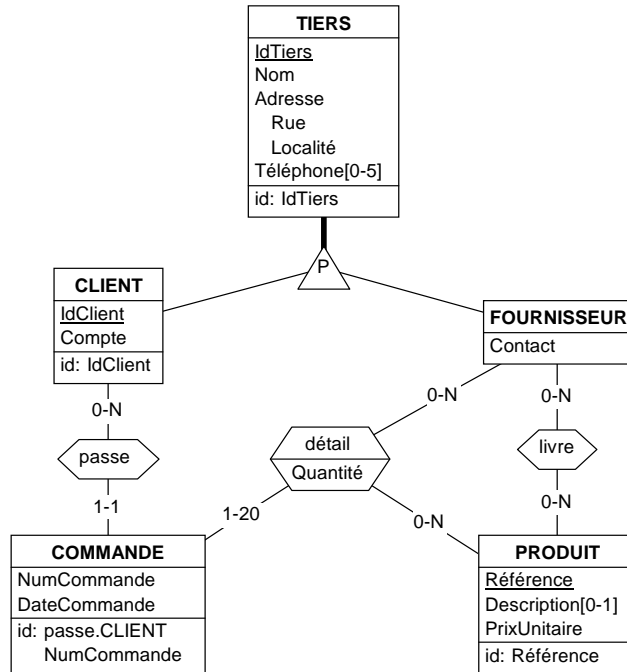
A19.3 Traduire le schéma conceptuel ci-dessous en structure de tables.



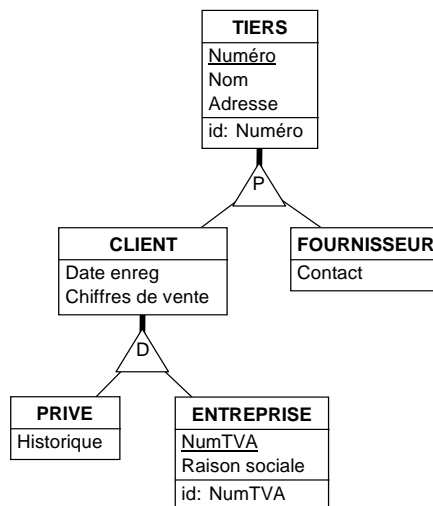
Solution



A19.4 Traduire le schéma conceptuel ci-dessous en structure de tables.

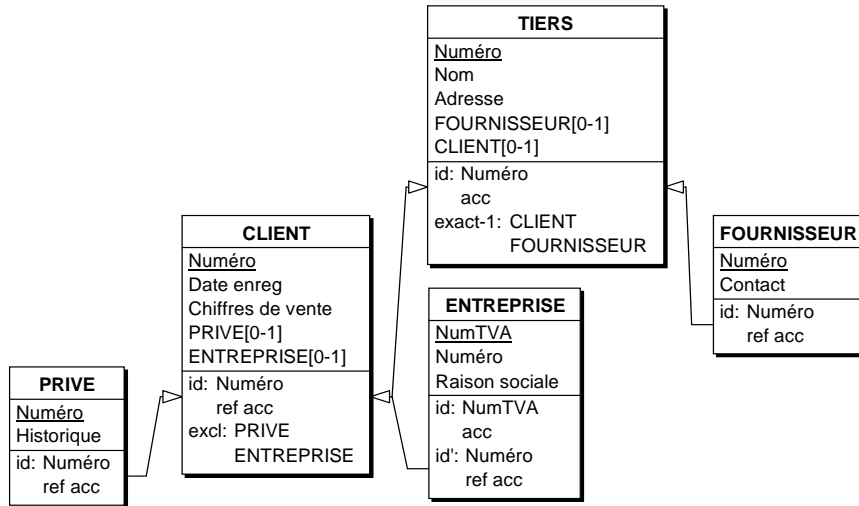


A19.5 Transformer le schéma ci-dessous en structures relationnelles selon l'approche standard (matérialisation). Produire ensuite, en utilisant les transformations par héritage descendant et héritage descendant, un schéma ne comportant qu'une seule table (TIERS), un schéma ne comportant que deux tables (CLIENT et FOURNISSEUR) puis un schéma constitué de la représentation exclusive des sous-types. Comparer ces schémas.

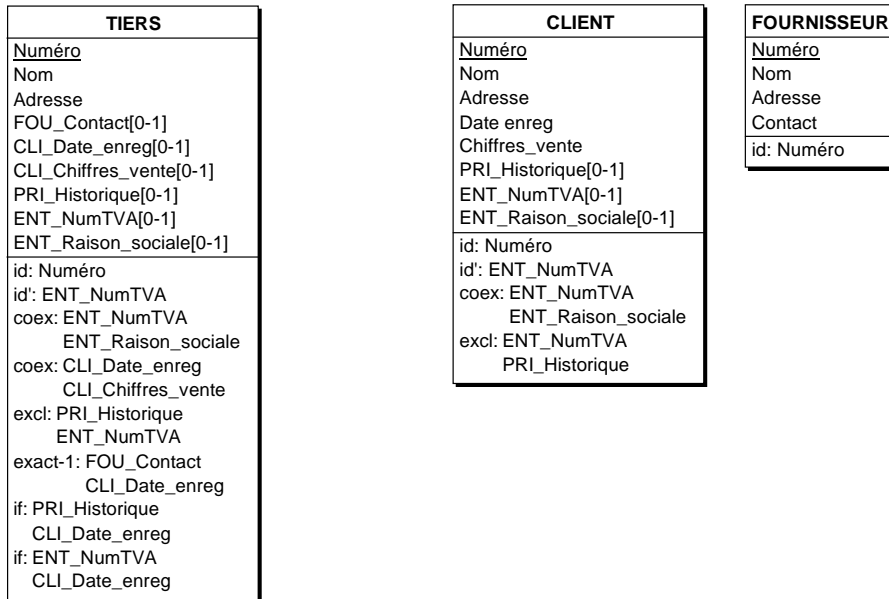


Solution

1. Approche standard (matérialisation) :



2. Représentation à 1 tables (gauche) et à 2 tables (droite) :



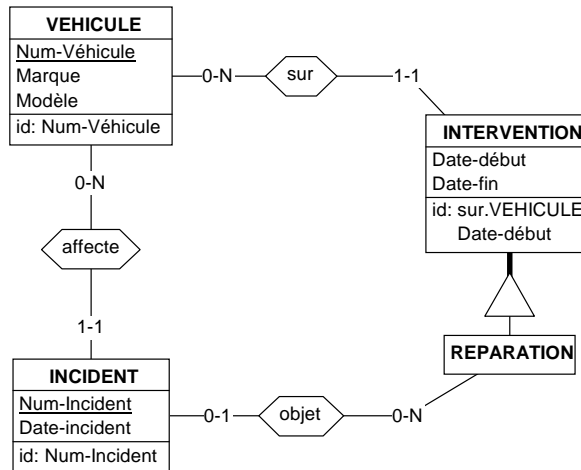
3. Représentation exclusive des sous-types :

| TIERS_CLIENT_PRIVÉ | TIERS_CLIENT_ENTREPRISE | TIERS_CLIENT_AUTRES | TIERS_FOURNISSEUR |
|--------------------|-------------------------|---------------------|-------------------|
| <u>Numéro</u> | <u>Numéro</u> | <u>Numéro</u> | <u>Numéro</u> |
| Nom | Nom | Nom | Nom |
| Adresse | Adresse | Adresse | Adresse |
| Date enreg | Date enreg | Date enreg | Contact |
| Chiffres de vente | Chiffres de vente | Chiffres de vente | id: Numéro |
| Historique | NumTVA | id: Numéro | |
| id: Numéro | Raison sociale | | |
| | id: Numéro | | |
| | id: NumTVA | | |

L'expression des contraintes d'intégrité additionnelles dans ces schémas est laissée à l'initiative du lecteur.

A19.6 Transformer le schéma suivant en schéma relationnel.

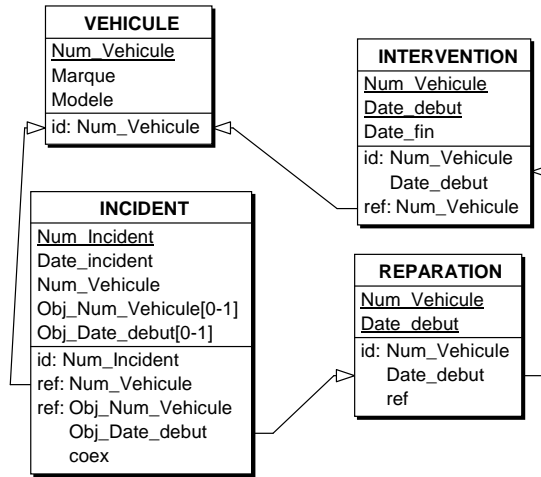
Note : ce problème, en apparence banal, est plus complexe qu'il n'y paraît. Il illustre la difficulté de construire un plan de transformation qui fonctionne dans tous les cas.



$INCIDENT.objet.REPARATION.sur.VEHICULE \subseteq INCIDENT.affecte.VEHICULE$

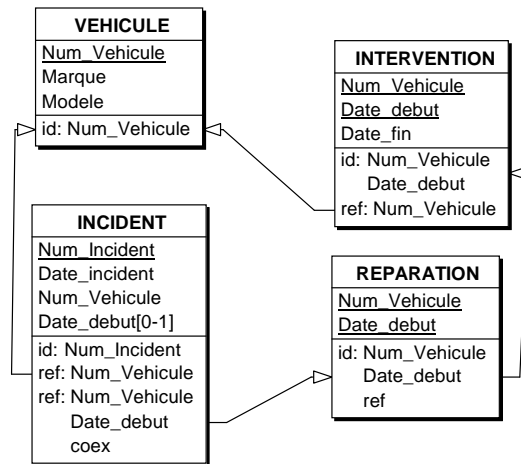
Solution

L'application stricte des règles de la conception logique relationnelle conduit au schéma suivant. La contrainte d'égalité des véhicules se traduit par une contrainte d'égalité des colonnes Num_Vehicule et Obj_Num_vehicule de la table INCIDENT.



INCIDENT.Num_vehicule = INCIDENT.Obj_Num_vehicule

L'utilisation d'une seule colonne permet de supprimer cette dernière contrainte. On obtient alors le schéma simplifié ci-dessous.

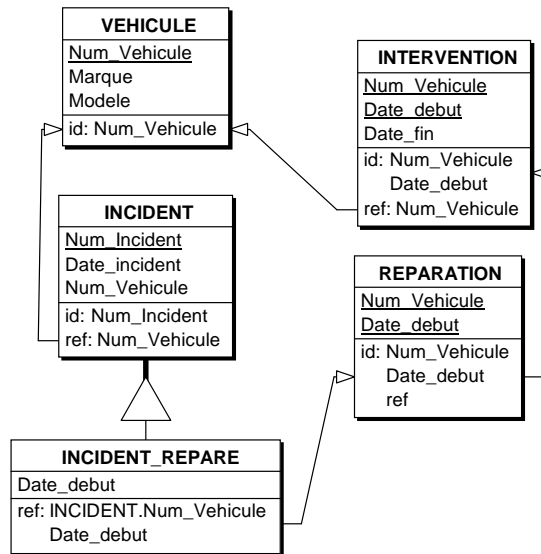


Ce schéma pose deux problèmes :

- la contrainte de coexistence est invalide, un de ses composants étant obligatoire;
- une clé étrangère est semi-facultative, constituée d'un composant obligatoire et d'un composant facultatif; cette construction est invalide dans le modèle relationnel de base tel que nous l'avons défini.

On résout le premier problème par la suppression de la contrainte de coexistence. Le second problème est plus complexe à traiter.

Dans un premier temps, on identifie une classe spécifique d'incidents : les *incidents réparés*, dont la particularité est d'être doté d'un attribut *Date_debut* obligatoire. C'est la table *INCIDENT_REPARE* de cette dernière classe qui référence *REPARATION*, via une clé étrangère constituée de l'attribut hérité *Num-Vehicule* et de l'attribut propre *Date_debut*.

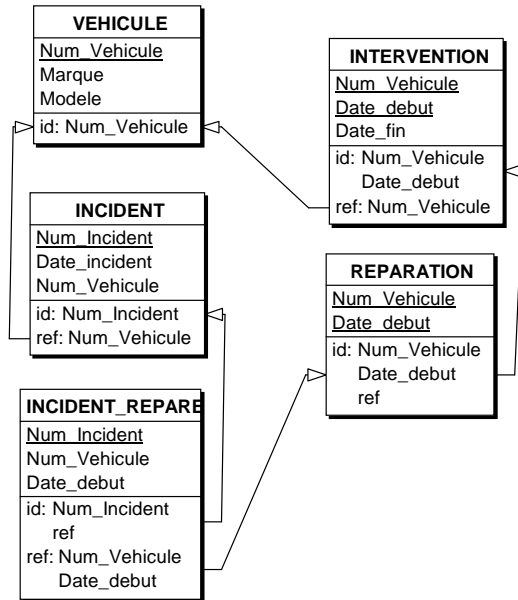


Il nous reste à transformer la relation *is-a*. Nous la transformons d'abord en type d'associations 1:1. Malheureusement, ce dernier ne peut à son tour être traduit sous la forme d'une clé étrangère de manière classique¹ : l'identifiant de *INCIDENT_REPARE* comporte un composant hérité qui ne lui appartient pas en propre. Il faudrait dès alors ajouter à cette table une colonne qui serait une copie de *INCIDENT.Num_Vehicule* et qui serait assortie d'une contrainte de redondance. On obtiendrait ainsi le schéma ci-après.

Il n'est pas certain que ce schéma soit plus clair que le premier, obtenu par la méthode classique. Tous deux contiennent une redondance et une contrainte qui exprime celle-ci. Dans la dernière solution, le schéma comporte une table supplémentaire (ce qui n'est pas un défaut en soi), une colonne héritée et une contrainte de redondance.

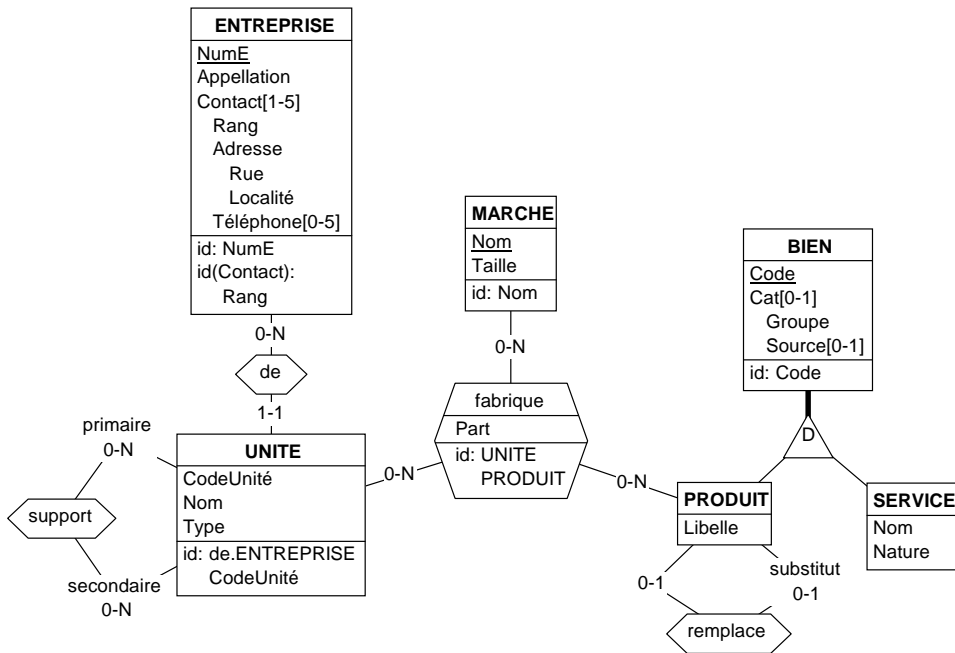
Conclusion : rien n'est parfait en ce bas monde.

1. En SQL2 du moins. En SQL3, le schéma peut être conservé tel quel.

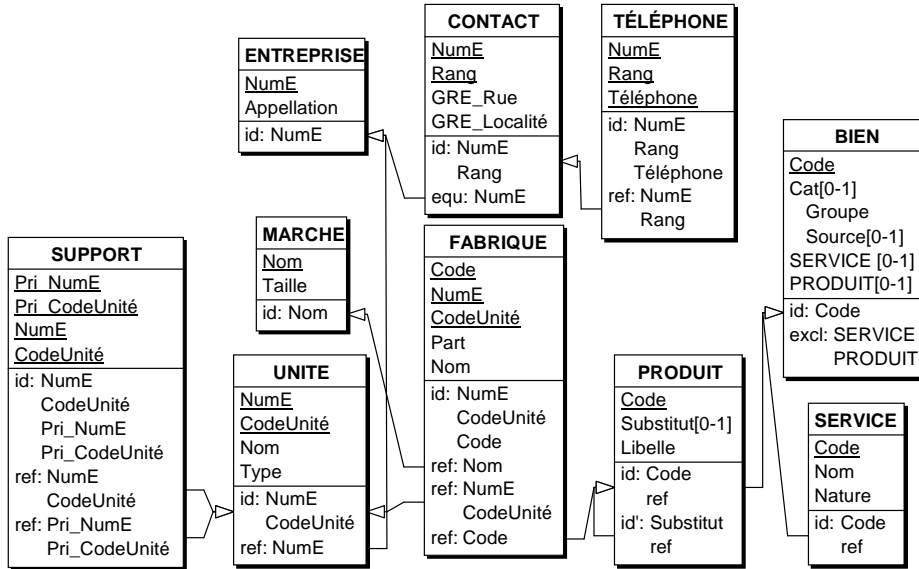


$\forall ir \in \text{INCIDENT_REPARE}, \forall i \in \text{INCIDENT},$
 $ir.\text{Num_Incident} = i.\text{Num_Incident} \Rightarrow ir.\text{Num_vehicule} = i.\text{Num_vehicule}$

A19.7 Transformer le schéma suivant en schéma relationnel.

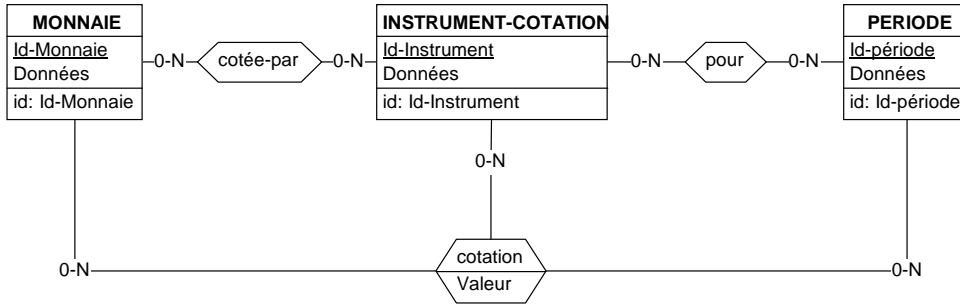


Solution



Il semble que le concepteur ait oublié de traiter une construction non conforme à SQL2. Achever le travail.

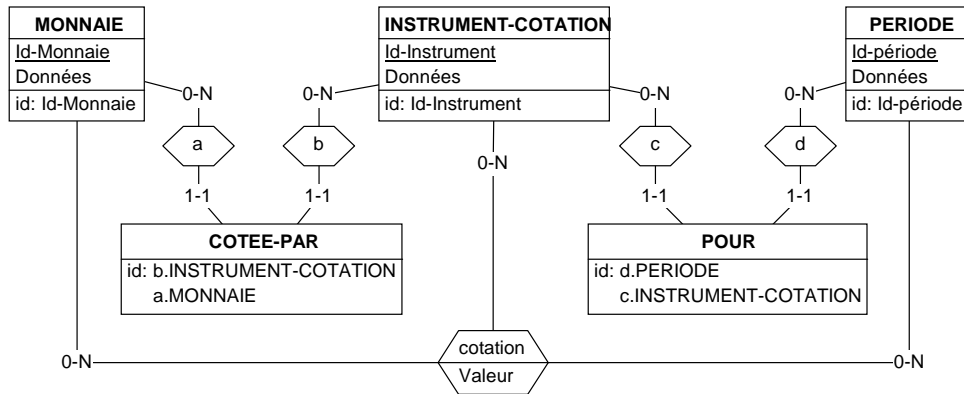
A19.8 Transformer le schéma suivant en schéma relationnel.



cotation[MONNAIE, INSTRUMENT-COTATION] \subseteq cotée-par
 cotation[INSTRUMENT-COTATION, PERIODE] \subseteq pour

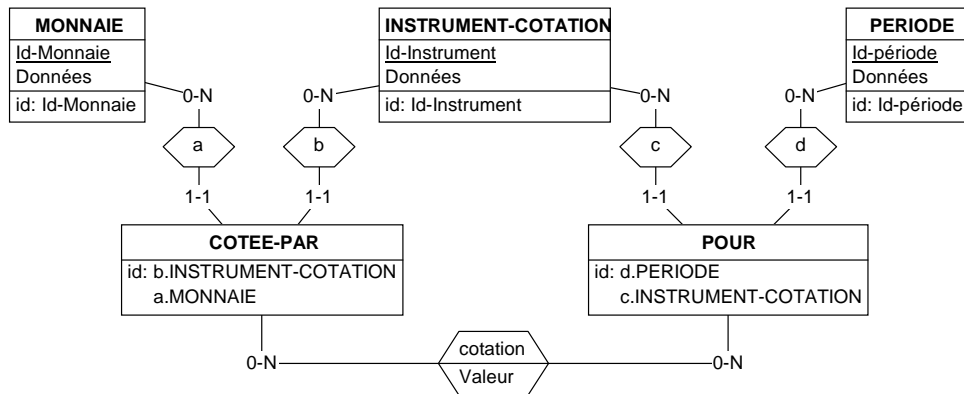
Solution

On transforme d'abord les types d'associations cotée-par et pour en types d'entités.



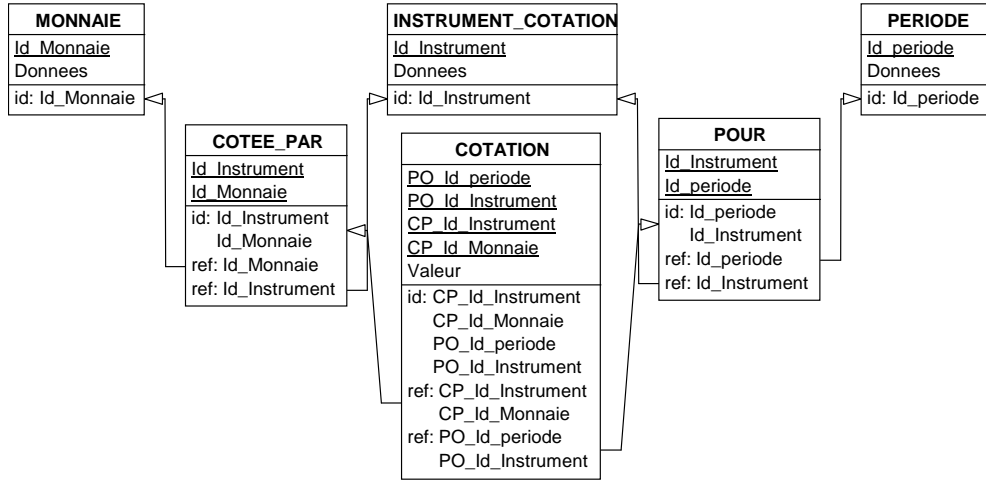
$$\begin{aligned} \text{cotation}[\text{MONNAIE}, \text{INSTRUMENT-COTATION}] &\subseteq a*b[\text{MONNAIE}, \text{INSTRUMENT-COTATION}] \\ \text{cotation}[\text{INSTRUMENT-COTATION}, \text{PERIODE}] &\subseteq c*d[\text{INSTRUMENT-COTATION}, \text{PERIODE}] \end{aligned}$$

Le type d'entités COTEE-PAR représentant les associations cotée-par (et de même pour POUR), on peut restructurer ce schéma comme ci-dessous. La contrainte exprime le fait que, pour une instance de cotation, les deux instances de INSTRUMENT-COTATION accessibles via COTEE-PAR et POUR sont identiques.

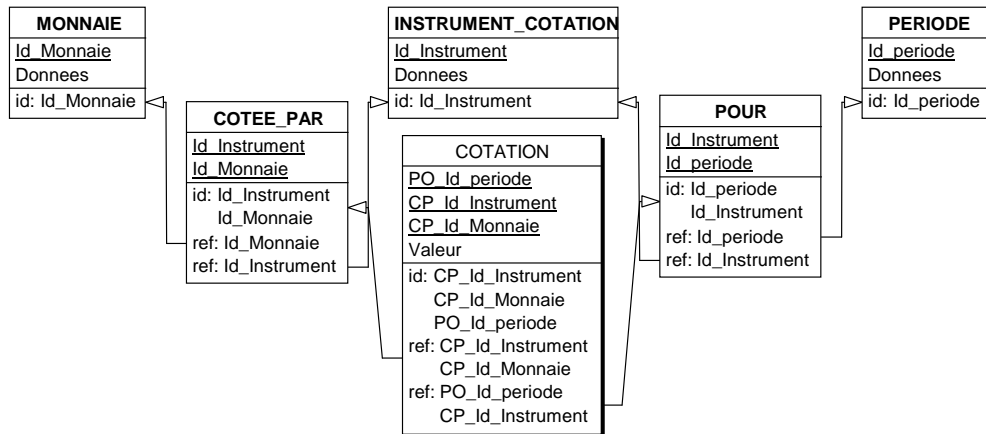


$$\forall (cp,p) \in \text{cotation}[\text{COTEE-PAR}, \text{POUR}], \text{COTEE-PAR}=cp)[\text{INSTRUMENT-COTATION}] = c(\text{POUR}=p)[\text{INSTRUMENT-COTATION}]$$

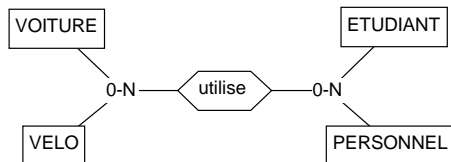
La traduction s'achève par la transformation du type d'associations cotation. La contrainte s'exprime par l'égalité des valeurs de deux colonnes.



En réduisant ces deux colonnes en une seule, on obtient le schéma final.

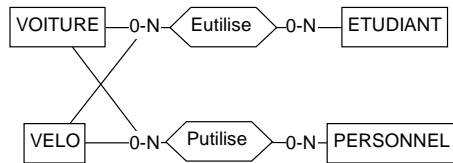


A19.9 Transformez le schéma ci-dessous en un schéma de tables.

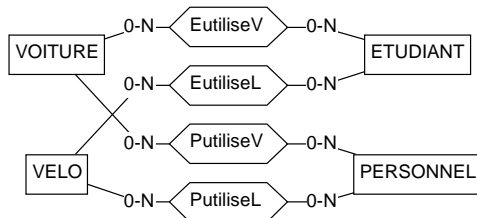


Solution

La technique par multiplication proposée à la section 18.5.2 permet de remplacer un premier rôle multi-types :

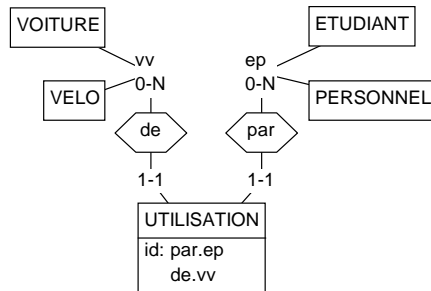


On fait apparaître deux nouveaux rôles multi-types, qu'on transforme successivement pour obtenir le schéma suivant :



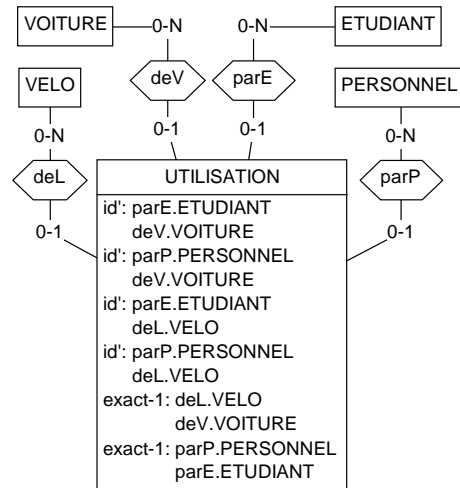
Sa traduction dans le modèle SQL2 ne pose pas de problème, mais le résultat est relativement complexe² (4 tables, 4 identifiants et 8 clés étrangères).

Selon la seconde technique de cette même section, on transforme d'abord le type d'associations en type d'entités :



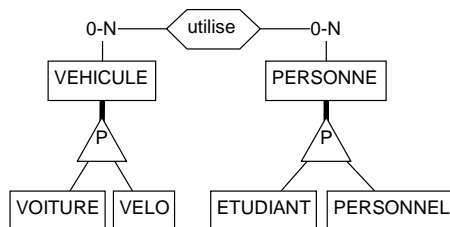
Puis on traite chaque rôle multi-types :

2. Juste pour se faire peur, on pourrait imaginer un type d'associations n-aire dont tous les rôles sont multi-types !



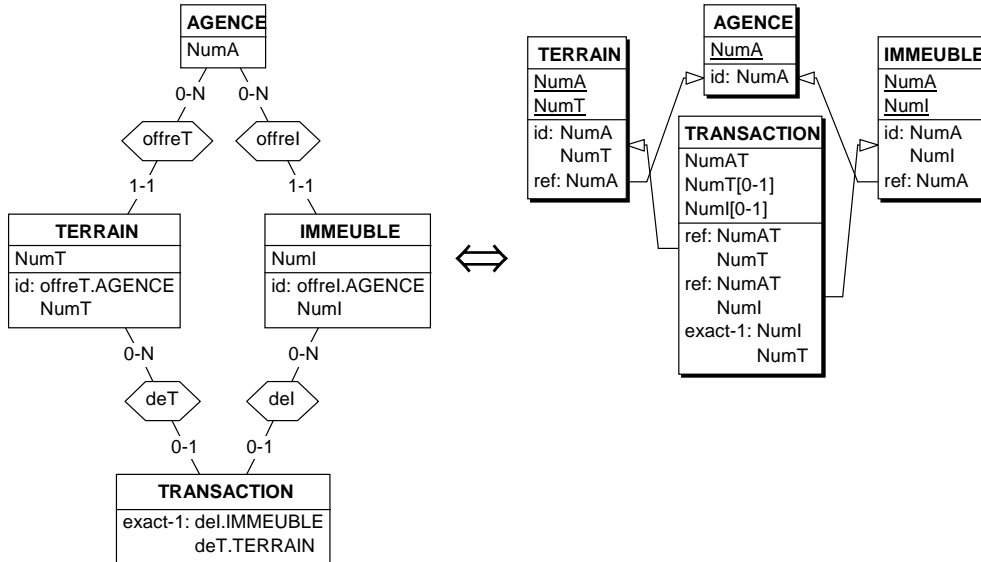
On observe l'intervention de rôles multi-types dans la multiplication des identifiants du type d'entités associations UTILISATION.

Face à la complexité des ces deux approches, on peut explorer une autre voie, qui part de l'observation de traits communs à VELO et VOITURE d'une part, et à ETUDIANT et PERSONNEL. Le premier groupe est constitué de *véhicules* et le second de personnes. On pourrait donc redessiner ce schéma à l'aide de relations *is-a* :



La suite des transformations est standard. L'applicabilité de cette troisième solution dépend de l'existence d'autres relations *is-a* impliquant les types d'entités de ce fragment de schéma.

A19.10 Un développeur propose la traduction suivante lors de la production d'un schéma relationnel. sa proposition est-elle correcte ?



A19.11 Développer un plan de transformation pour les diagrammes de classes UML.

A19.12 Compléter le plan de transformation de la figure 18.30 de sorte qu'il traite les rôles multi-types.

A19.13 Construisez un script DB-MAIN qui traduit le plan de transformation de la question A17.12. Ce script est-il idempotent ?

