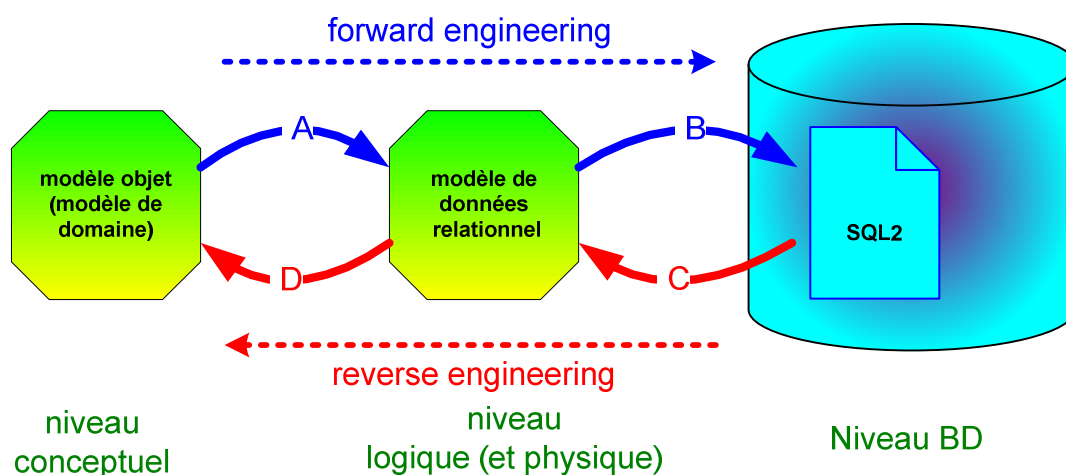


Tutoriel de transformation d'un modèle objet en schéma et SQL sous Rose

Table des matières

1	INTRODUCTION.....	1
2	PREPARATION DU MODELE OBJET.....	2
2.1	MARQUEZ LES CLASSES PERSISTANTES	2
2.2	DESIGNEZ L'OBJECT IDENTITY DES CLASSES.....	3
3	TRANSFORMATION DU PACKAGE EN SCHEMA	4
1.	LE SCHEMA CIBLE.....	4
3.1	RESULTAT	5
4	AMELIORATION DU SCHEMA.....	5
4.1	MODELE DE DONNEES BRUT	6
4.2	REORGANISATION DU MODELE DE DONNEES	6
4.3	MODIFICATION DU MODELE DE DONNEES	7
4.3.1	Exemple avec le cas Mutin.....	7
4.3.2	Remarque sur la transformation	8
5	LA TRANSFORMATION EN SCRIPT SQL2.....	9
5.1	LES ECRANS	9
5.2	LE CONTENU DES FICHIERS SQL.....	11
6	TRAVAIL A REALISER.....	11

1 Introduction



Ici, nous utilisons la « transformation A » d'un diagramme de classes UML niveau conceptuel (modèle de domaine) en un schéma de base de données relationnelle puis nous utilisons la « transformation B » de ce schéma en script SQL2 (transformation B).

Avant chacune de ces transformations, on prépare le modèle. Après chacune des transformations, on améliore le modèle.

Dans le schéma (logical view) de Rose, pour travailler plus facilement, on utilise un diagramme qui s'appelle un « modèle de données ».

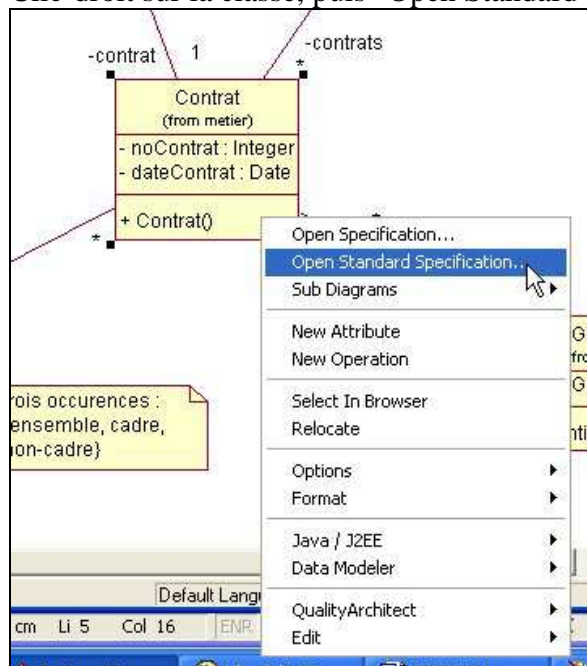
2 Préparation du modèle objet

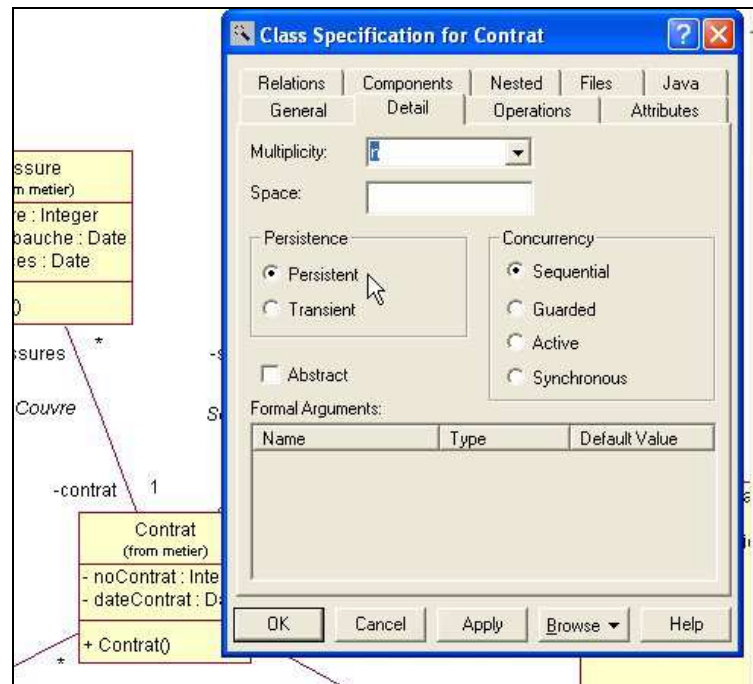
Notez les remarques suivantes (elles sont appliquées dans les rubriques suivantes) :

1. Dans le compartiment « Component View », il faut disposer d'un composant de type Database (**voir documents précédents**) et dans « logical view », il faut disposer d'un schéma associé à cette Database.
2. Le diagramme de classes UML doit avoir les caractéristiques suivantes :
 - Toutes les classes concernées par la transformation doivent être marquées Persistent (clic droit sur la classe, Open Specification, onglet Détail).
 - Ces classes persistantes doivent être groupées dans un package du compartiment Logical View car l'outil transforme un package vers un schéma.
 - Les classes concernées par la transformation doivent avoir, si possible, leur "object identity" fixé.

2.1 Marquez les classes persistantes

Clic-droit sur la classe, puis "Open Standard Specification..."





2.2 Désignez l'Object IDentity des classes

(Définition des clés primaires)

Object IDentity = future clé primaire

La désignation de l'OID est optionnel.

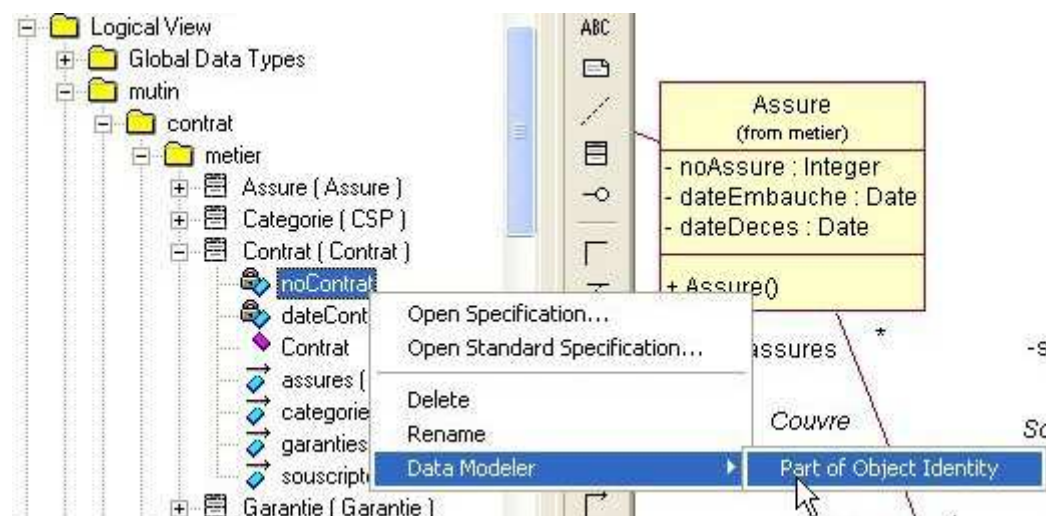


Figure : Le champ noContrat jouera le rôle de clé primaire

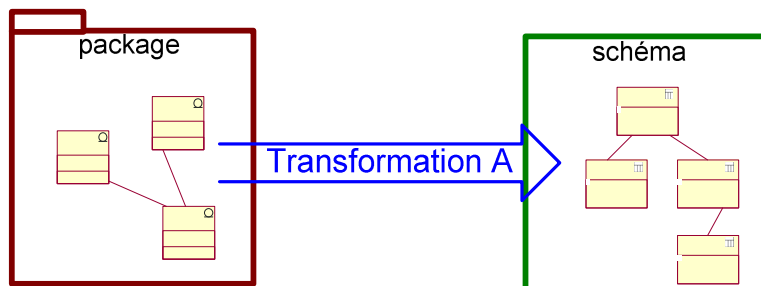
Pour identifier les clés primaires, il faut marquer les attributs concernés en tant que "Part of Object Identity" via un clic droit sur l'attribut suivi d'un clic sur le menu Data Modeler.

Pour une classe persistante, on peut ne pas choisir d'attribut pour définir la clé primaire. Dans ce cas, au moment de la transformation, l'outil Rose ajoutera automatiquement un attribut avec un nom « nomTable »_ID et un type NUMBER(5,0) (pour Oracle).

L'outil Rose applique les règles de passage du modèle conceptuel (modèle Entité/Association) au modèle logique relationnel. Par exemple, certaines colonnes marquées en tant que clés

primaires migreront automatiquement dans les tables filles et deviendront des clés étrangères lors de la transformation des associations.

3 Transformation du package en schéma

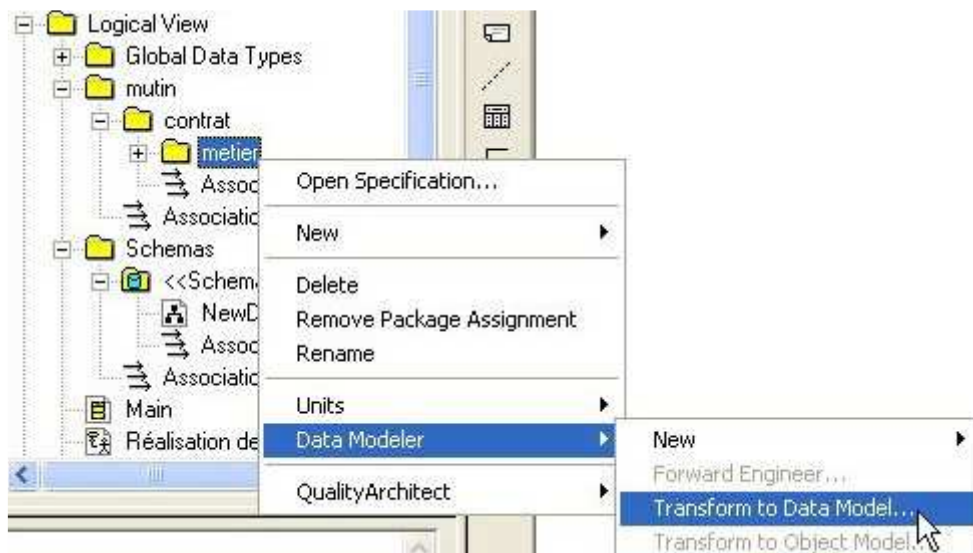


Afin de créer la base de données relationnelle qui résultera de la transformation du modèle objet, il faut effectuer les opérations suivantes :

- effectuer la transformation package vers schéma,
- Amélioration du schéma brut
 - créer un diagramme modèle de données,
 - glisser-déplacer des tables dans le diagramme modèle de données,
 - modifier le modèle de données.

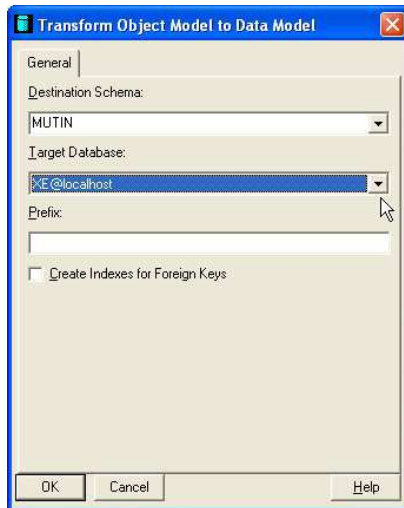
Rappel sur les documents précédents : pour créer une base de données, il faut se placer dans le répertoire Component View, puis choisir par un clic droit l'option DataModeler/New/Database ...

Pour lancer la transformation du modèle objet, il faut sélectionner le **package** contenant les classes UML à transformer (clic droit. DataModeler/Transform to DataModel...).



1. Le schéma cible

Il faudra ensuite indiquer le nom du schéma cible et le nom de la base de données cible. Si aucune base n'est sélectionnée, la transformation sera ANSI-92 (SQL2).

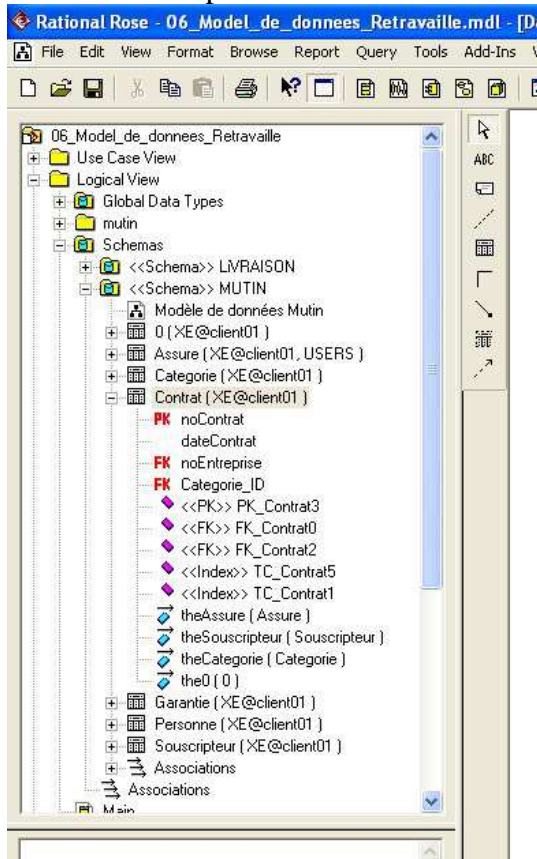


Enfin, on peut créer des index pour les colonnes clés étrangères.

3.1 Résultat

La transformation génère des tables placées sous le schéma. Ces tables comportent des clés primaires et étrangères, car la transformation a suivie les règles de passage du modèle conceptuel au modèle logique.

Les tables sont placée sous le schéma mais ne sont pas placées dans le diagramme.

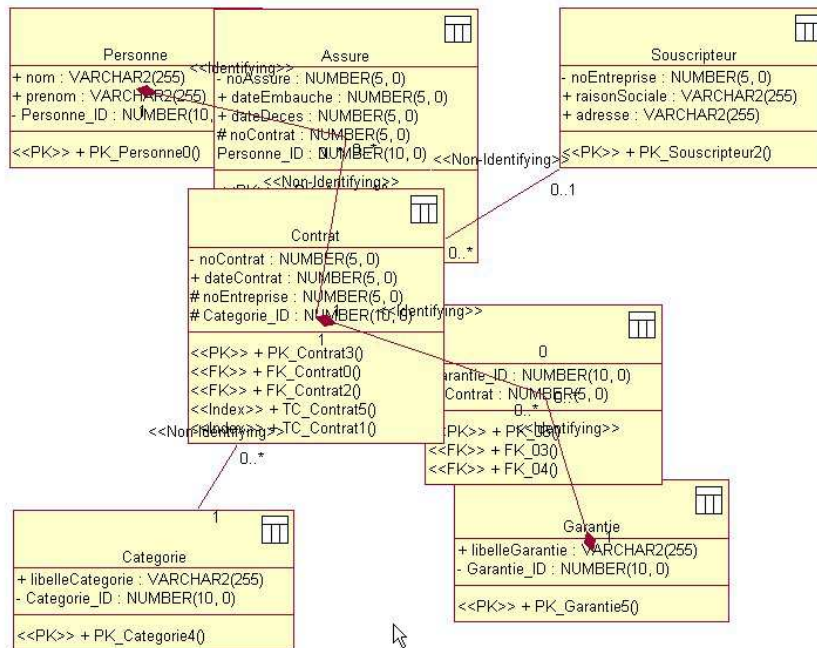


4 Amélioration du schéma

Pour illustrer le schéma, on utilise un modèle de données. Le modèle de données n'est qu'une fenêtre sur le schéma.

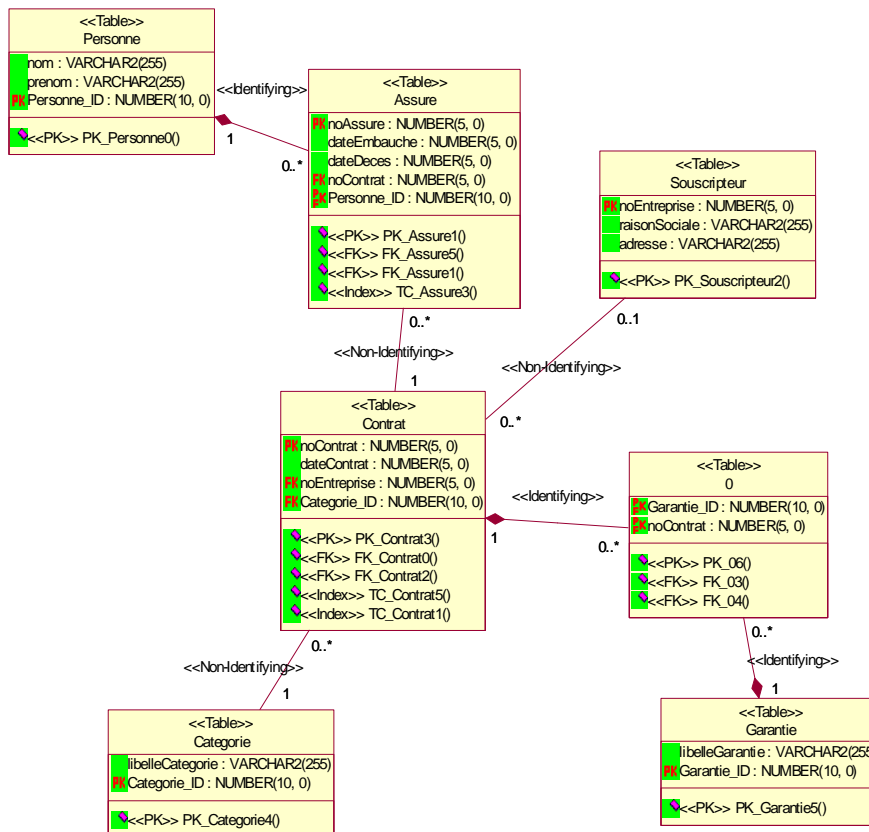
4.1 Modèle de données brut

Après un glisser-déplacer des tables dans le diagramme modèle de données, on obtient le diagramme brut suivant:



4.2 Réorganisation du modèle de données

Arranger le modèle.



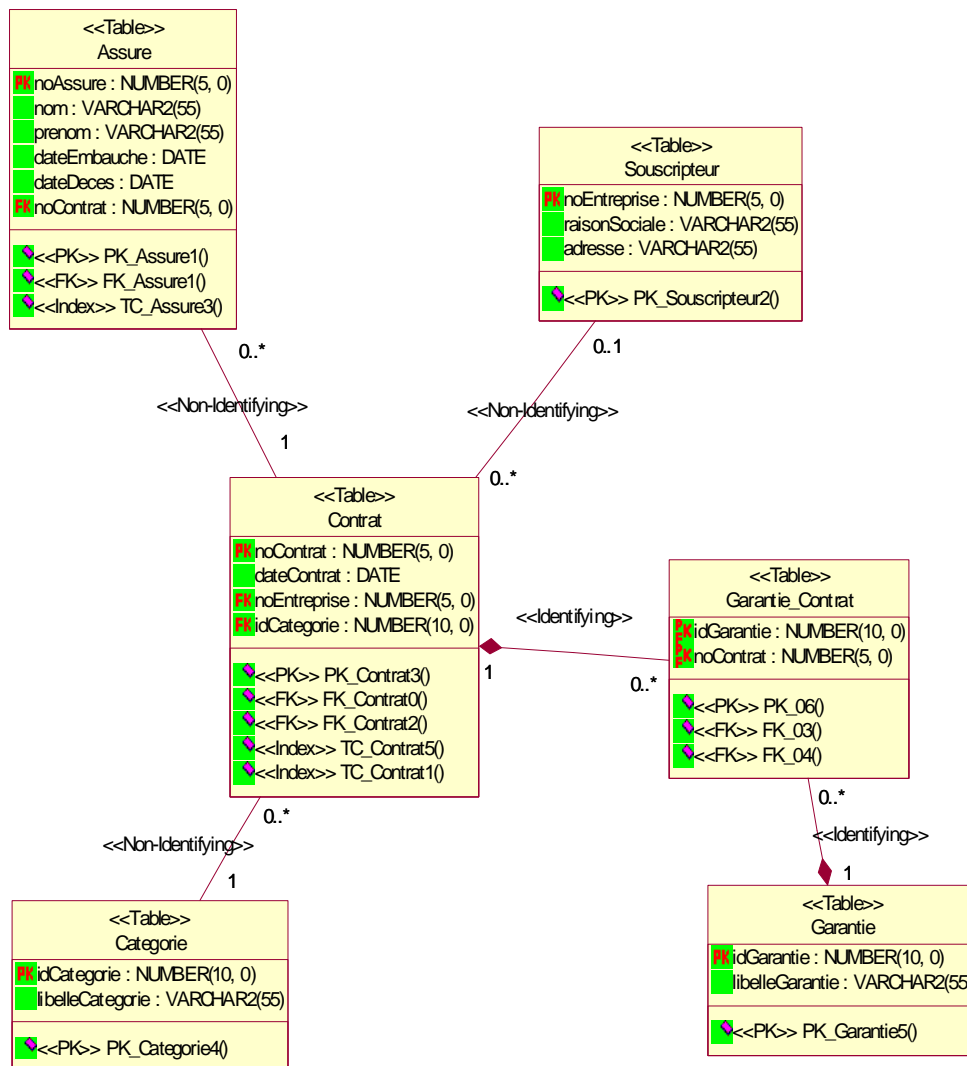
4.3 Modification du modèle de données

Vous pouvez modifier certaines caractéristiques du Data Model Diagram avant de générer le script SQL2 équivalent (noms des contraintes, noms des colonnes, utilisation des domaines, etc.). Dans ce cas, il convient d'être prudent car il y aura un déphasage avec le modèle objet. Alors, vous pourrez régénérer un modèle objet à partir d'un modèle de données.

Voici quelques exemples de modification qui peuvent être apportées au modèle:

- renommer les tables d'association,
- renommer éventuellement les clés primaires générées par l'outil,
- placer les clés primaires en haut du compartiment des attributs.

Dans le paragraphe plus loin, on décrit les modifications effectuée.



4.3.1 Exemple avec le cas Mutin

Par exemple, dans le cas Mutin avec CU Gérer les contrats, on effectue les modifications suivantes:

- On supprime la table Personne (héritage), de la manière suivante:
 - glisser-déplacer "nom" et "prenom" dans la table Assure

- supprimer la clé Personne_ID
- supprimer la table Personne

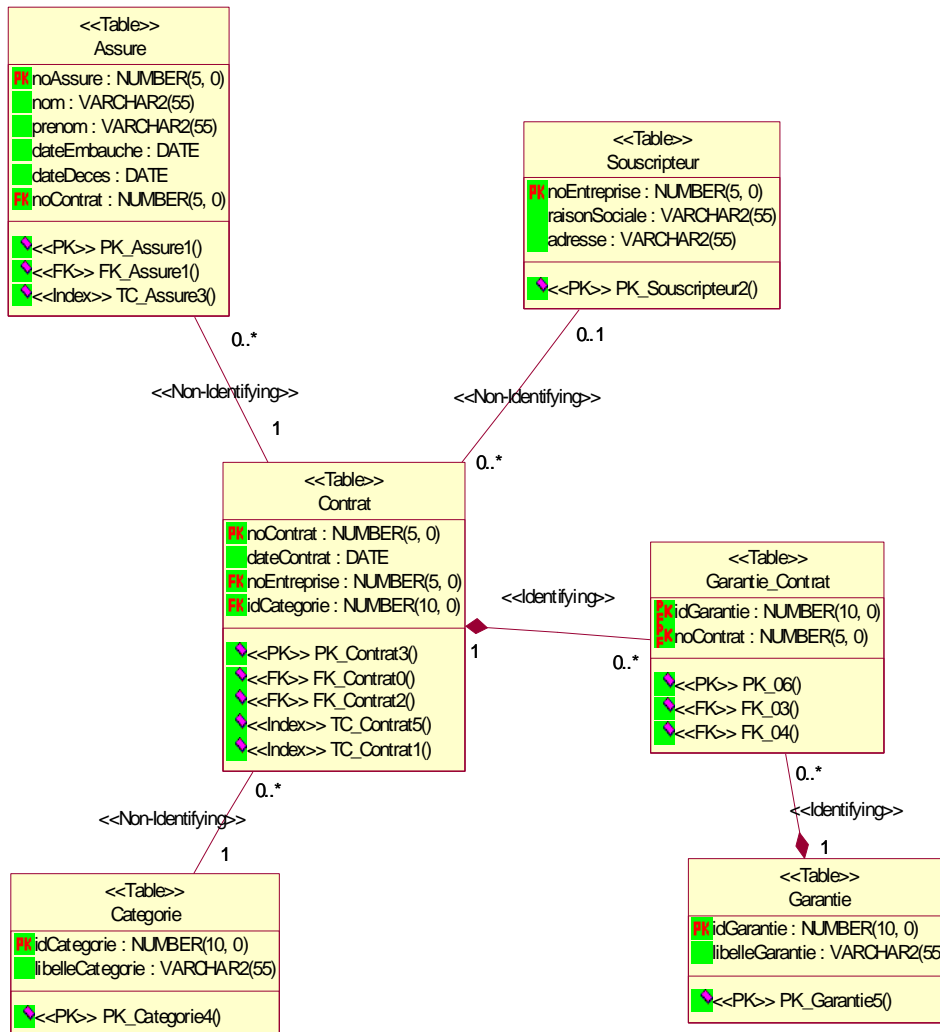
Renommer la table d'association "0" en "Garantie_Contrat"

Dans Categorie et Contrat, changez Categorie_ID en "idCategorie".

Dans Garantie et Garantie_Contrat, changez Garantie_ID en "idGarantie".

Renommer les contraintes PK des tables.

Pour le typage des champs, on met des DATE et on passe VARCHAR2(255) en VARCHAR2(n) le mieux adapté.



4.3.2 Remarque sur la transformation

Tous les éléments du modèle objet ne se transforment pas forcément en classes de stéréotype «Table» au niveau logique. La documentation stipule qu'il est évidemment plus prudent d'examiner en détail le résultat de la transformation de manière à s'assurer de la bonne structure de la base. On retrouve ici l'idée essentielle, à savoir, la maîtrise des concepts pour une meilleure utilisation de l'outil.

Chaque package se transforme en un schéma situé dans le compartiment

Logical_View/Schémas. La transformation d'un package de modèle objet dans un modèle de

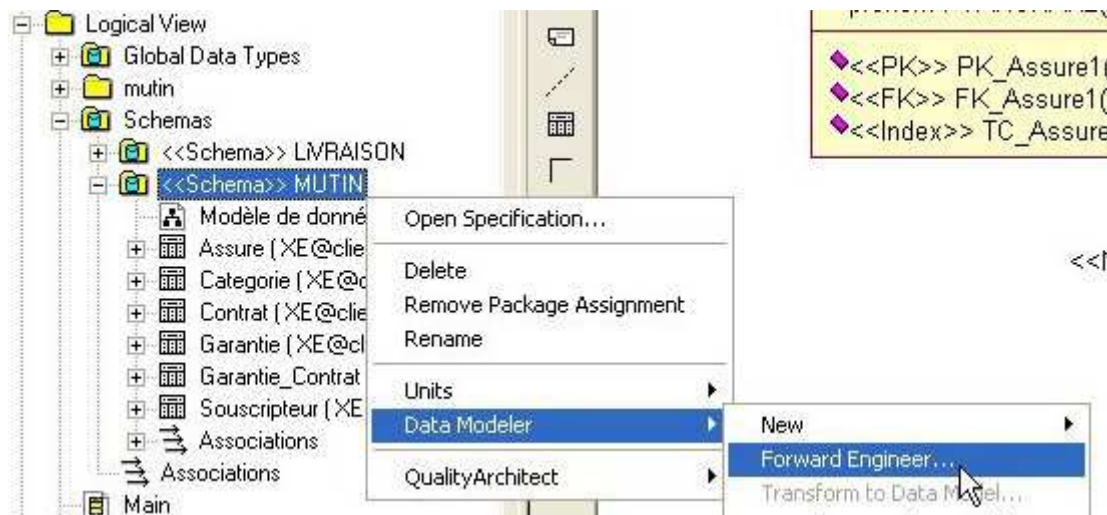
données existant modifie le modèle de données par l'ajout des nouveaux composants mais ne détruit ni ne modifie les éléments du modèle de données existants.

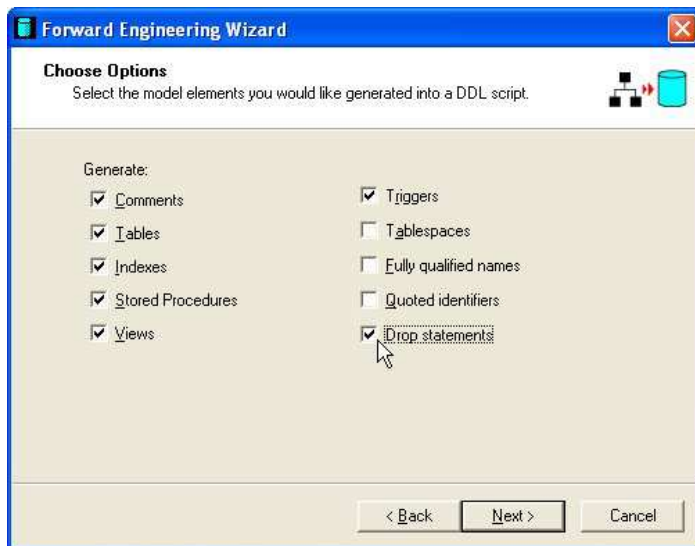
5 La transformation en script SQL2

Nous allons générer un script SQL.(Transformation B)

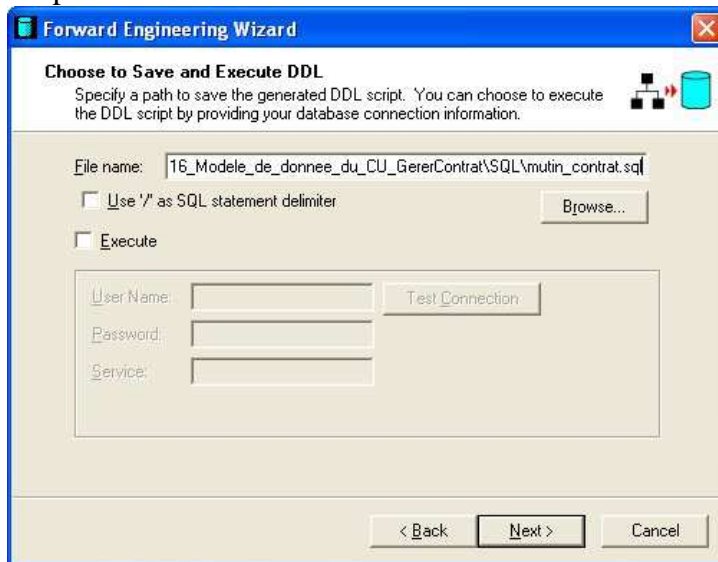
5.1 Les écrans

Après les modifications éventuelles du Data Model Diagram, on peut générer le script SQL2 pour Oracle en sélectionnant le schéma par un clic droit, puis sélectionner DataModeler/Forward_Engineer...





Ne pas mettre les index



A l'aide de l'assistant, nous avons choisi, dans notre exemple, de générer les clauses DROP de manière à pouvoir relancer le script à la demande. Vérifier la présence des contraintes de type CHECK.

5.2 Le contenu des fichiers SQL

```
ALTER TABLE Contrat DROP CONSTRAINT FK_Contrat0;
ALTER TABLE Contrat DROP CONSTRAINT FK_Contrat2;
DROP TABLE Contrat;
CREATE TABLE Contrat (
    noContrat NUMBER ( 5 ) NOT NULL,
    dateContrat DATE NOT NULL,
    noEntreprise NUMBER ( 5 ),
    idCategorie NUMBER ( 10 ) NOT NULL,
    CONSTRAINT PK_Contrat3 PRIMARY KEY (noContrat)
);
...
```

6 Travail à réaliser

Générez les scripts SQL de votre projet fil-rouge (les bibliothèques) en partant du modèle de domaine.