

Le plugin Rose Data Modeler le profil UML « Data Modeling » le modèle de données (~MPD)

Table des matières

1	DEFINITIONS	1
1.1	IBM-RATIONAL	1
1.2	ROSE	1
1.3	LES PROFILS DANS UML	1
1.4	LE PROFIL UML POUR LA MODELISATION DE DONNEES DANS ROSE.....	2
2	DIFFERENTES UTILISATIONS DE ROSE.....	2
2.1	TRANSFORMATION DE MODELE	2
2.2	CONSTRUCTION DE MODELE A PARTIR DE RIEN.....	2
3	LE PROFIL UML POUR LA CONCEPTION DE BASES DE DONNEES.....	3
3.1	DIAGRAMME DE COMPOSANTS ET ELEMENTS D'ARCHITECTURE.....	3
3.1.1	<i>Création d'un tablespace.....</i>	<i>3</i>
3.2	UN SCHEMA DANS LA BASE DE DONNEES	4
3.2.1	<i>Les éléments du schéma</i>	<i>5</i>
3.2.2	<i>Visibilité des attributs.....</i>	<i>6</i>

1 Définitions

1.1 IBM-Rational

La société Rational est à l'origine de la standardisation d'UML. Aujourd'hui, c'est une entité de la société IBM. En 2004, plus de 500 établissements d'enseignement supérieur avaient intégré les outils de la société Rational dans leurs cursus. De plus, le logiciel Rational Rose comptait plus de 500 000 utilisateurs professionnels à travers le monde. Enfin les employés de la société Rational ont écrit ou coécrit plus de 50 ouvrages.

1.2 Rose

L'outil de conception de Rational Software s'appelle "Rational Rose Data Modeler" (Rational Rose Visual Modeler), il est inclus dans Rational Rose Enterprise Edition et Rational Suite AnalystStudio. Dans la suite du document et pour ne pas compliquer les choses, nous confondrons "Rational Rose Data Modeler" avec le terme "Rose".

1.3 Les profils dans UML

Un profil (profile en anglais) est une proposition d'une communauté et regroupe un ensemble d'éléments UML (composants, stéréotypes, icônes, propriétés...) qui s'appliquent à un contexte particulier et qui conservent le métamodèle d'UML intact. Quand un tel profil est valide au niveau de la notation, il est dit « bien formé ». Par exemple, Rational a proposé à Sun, un profil UML pour la gestion des EJB (Enterprise Java Beans), ce profil a été accepté par la suite.

Rose inclut un **profil** convenant aux bases de données et s'appuyant sur SQL2 : "UML profile for Data Modeling"

1.4 Le profil UML pour la modélisation de données dans Rose

Le profil UML pour la modélisation de données (UML profile for Data Modeling) que vous allez utiliser dans Rose permet de modéliser et d'implanter une base de données à l'aide notamment de classes UML possédant des stéréotypes prédéfinis (Table, RelationalTable, View,...,etc).

2 Différentes utilisations de Rose

Nous allons utiliser Rose Data Modeler avec trois modèles distincts:

Le type de modèle	Formalisme	Niveau
un modèle objet	diagramme de classes	niveau conceptuel
un modèle de données relationnel	diagramme de classes	niveau logique
un modèle de stockage des données	scripts SQL2 du SGBD cible	niveau physique

Il faut avoir de bonnes notions d'UML pour utiliser le premier modèle et connaître les principes de conception de bases de données pour utiliser correctement les autres modèles.

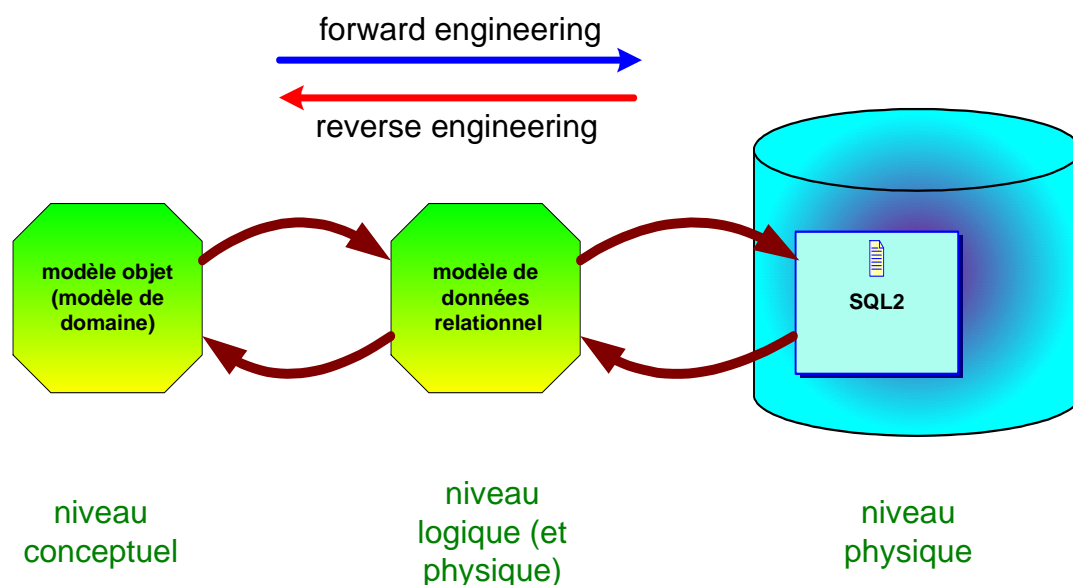


Figure: les transformations possibles

2.1 Transformation de modèle

L'outil permet les transformations de modèle suivantes:

- extraire un modèle logique à partir d'une base en exploitation (reverse engineering),
- créer un modèle logique à partir d'un diagramme de classes UML ou inversement,
- générer un script SQL2 à partir d'un modèle logique (forward engineering).

L'outil permet aussi de comparer et synchroniser des schémas de mêmes modèles.

2.2 Construction de modèle à partir de rien

En plus de ces transformations de modèles et en partant de rien, on peut construire un modèle de données relationnel ou un modèle de données physique en ajoutant des éléments (voir TPs à venir).

Rose prend en charge la plupart des SGBD du marché (IBM DB2 MVS et UDB 5 à 7, Oracle 7.x et 8.x, Microsoft SQL Server, Sybase Adaptive Server 12.x, etc.).

Rose ne permet pas encore de traduire automatiquement un diagramme de classes en un script SQL3 incluant des caractéristiques comme des pointeurs et des collections.

Pour ce faire, il faudra modifier manuellement le modèle de données ou le script SQL.

Dans le paragraphe suivant, nous présentons les caractéristiques principales du profil UML pour la modélisation des données.

3 Le Profil UML pour la conception de bases de données

(UML profile for Data Modeling)

Voici une illustration des différents éléments du profil UML pour la conception de bases de données.

3.1 Diagramme de composants et éléments d'architecture

Le profil UML permet de représenter une architecture, au sens large du terme. Cette architecture est représentée dans un diagramme de composants.

L'architecture peut contenir un ou plusieurs serveurs hébergeant une ou plusieurs bases de données (instances) faisant intervenir un ou plusieurs espaces de stockage (tablespace) qui contiennent les objets (tables).

Pour définir un composant dans un diagramme, il faut faire un clic droit au niveau du répertoire Component View, puis choisir New/Component. Certains stéréotypes sont prédéfinis (comme Database), d'autres peuvent être créés (comme Server).

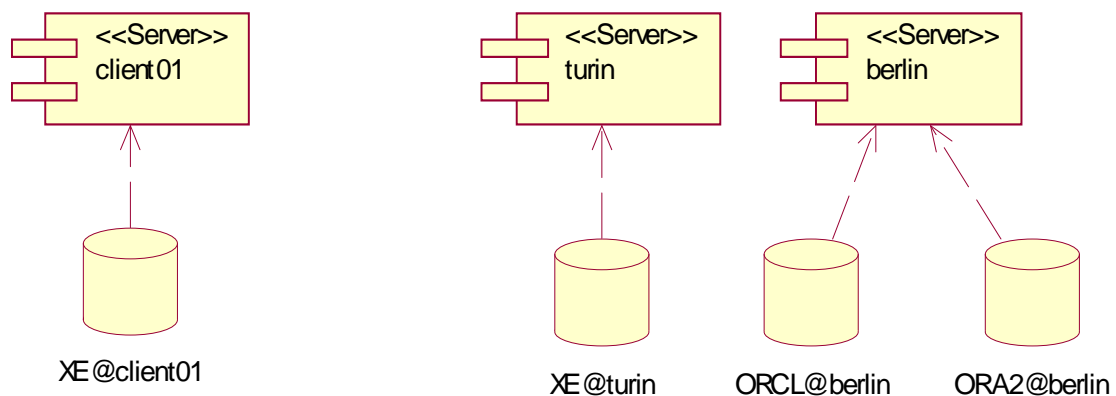


figure : Diagramme de composant illustrant les serveurs et leurs bases de données

3.1.1 Création d'un tablespace

Nous verrons plus loin que 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 créer un espace de stockage associé à une base, il faut effectuer un clic droit sur la base en question puis choisir DataModeler/New/Tablespace.

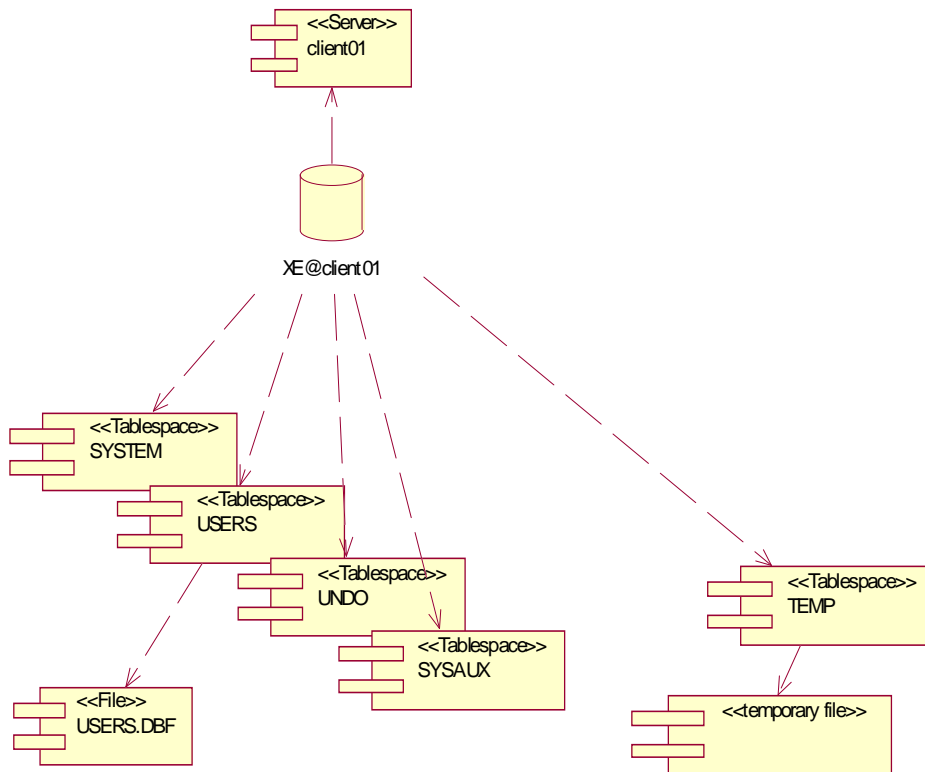
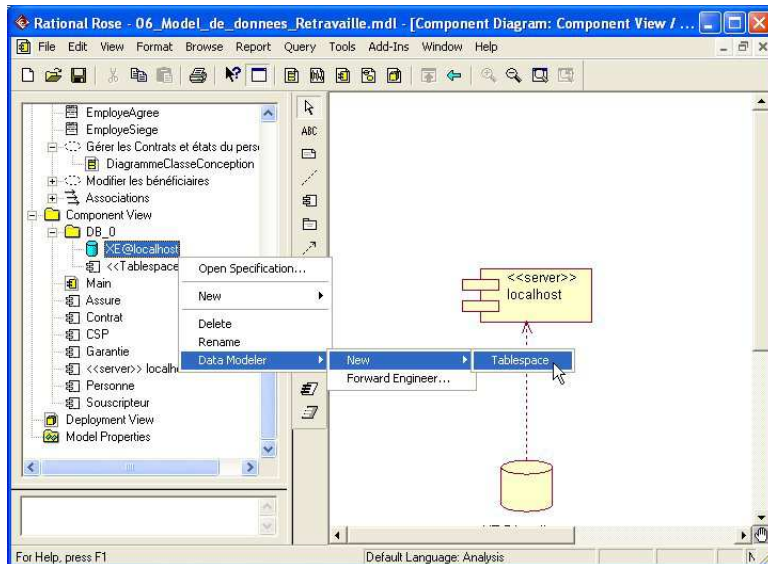
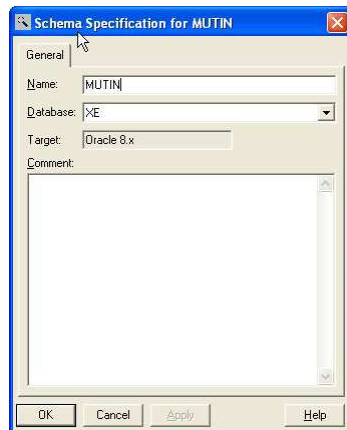
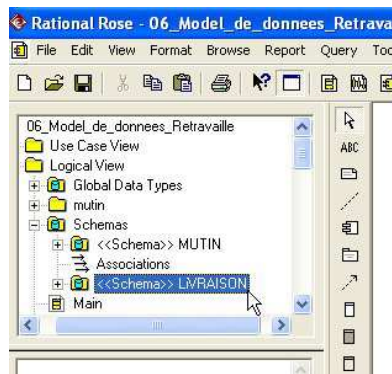


figure : diagramme de composant illustrant une base de données avec ses tablespaces

3.2 Un schéma dans la base de données

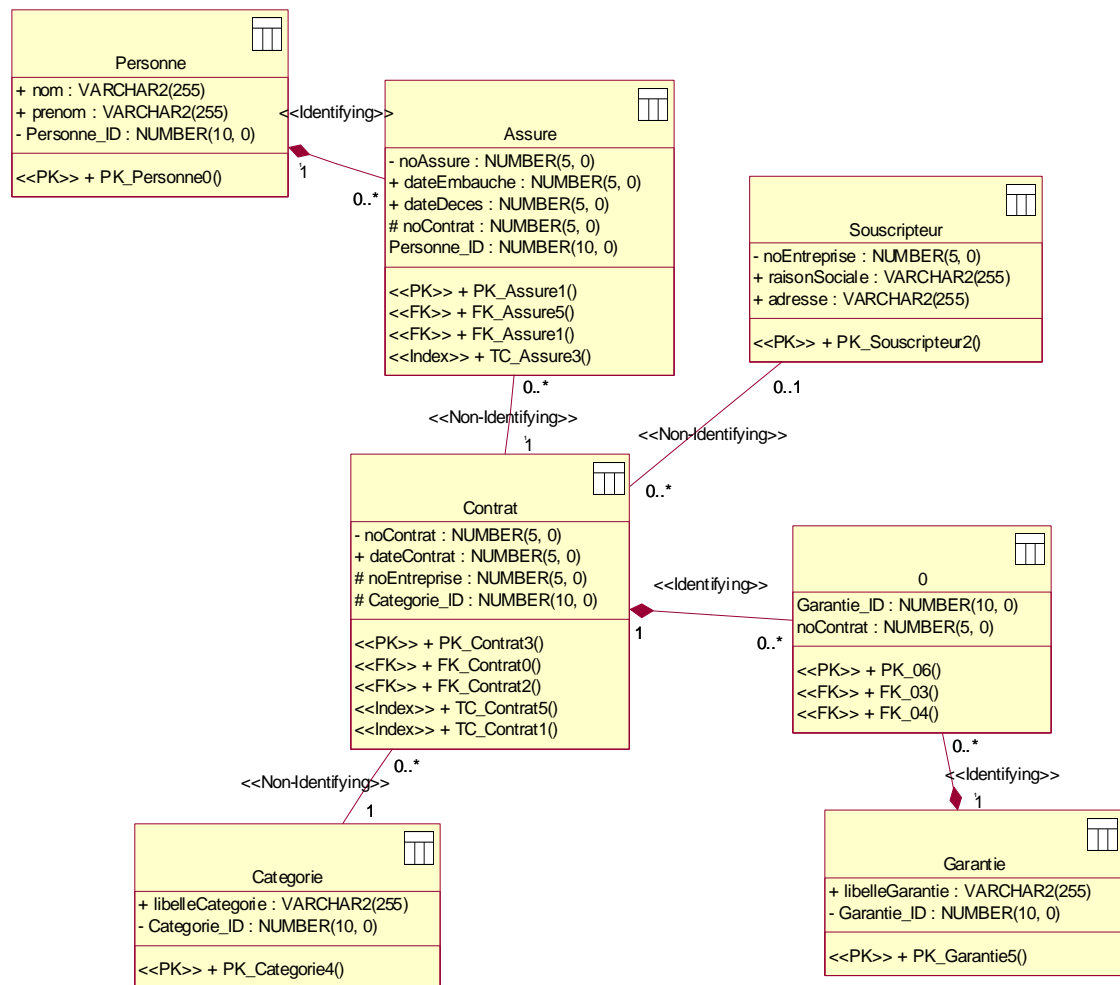
Une base de données est composée de plusieurs schémas. Ces derniers sont représentés dans le package ComponentView.



3.2.1 Les éléments du schéma

Différents éléments sont proposés pour la modélisation des tables, vues et associations. Nous verrons plus loin qu'on peut aussi représenter une table par une classe UML de stéréotype «Table». Nous expliquerons plus loin la signification des associations et le moyen de les définir correctement.

Voici un exemple de diagramme :



3.2.2 Visibilité des attributs

Avec le profil du Data Modeler, la visibilité des attributs a la signification suivante:

- "-" clé primaire
- "#" clé étrangère
- "+" champs sans contrainte
- " " (rien) champ clé primaire et clé étrangère