Test de Case Studio 2.18

par Jean-Alain Baeyens

Date de publication : 21/06/2005

Dernière mise à jour :

Vous trouverez ci-dessous un test relativement complet de l'outil Case Studio 2.18. Case Studio est un logiciel de modélisation de base de données. Ce genre d'outils est destiné à tout les développeurs. (Version PDF)

I - Fiche produit

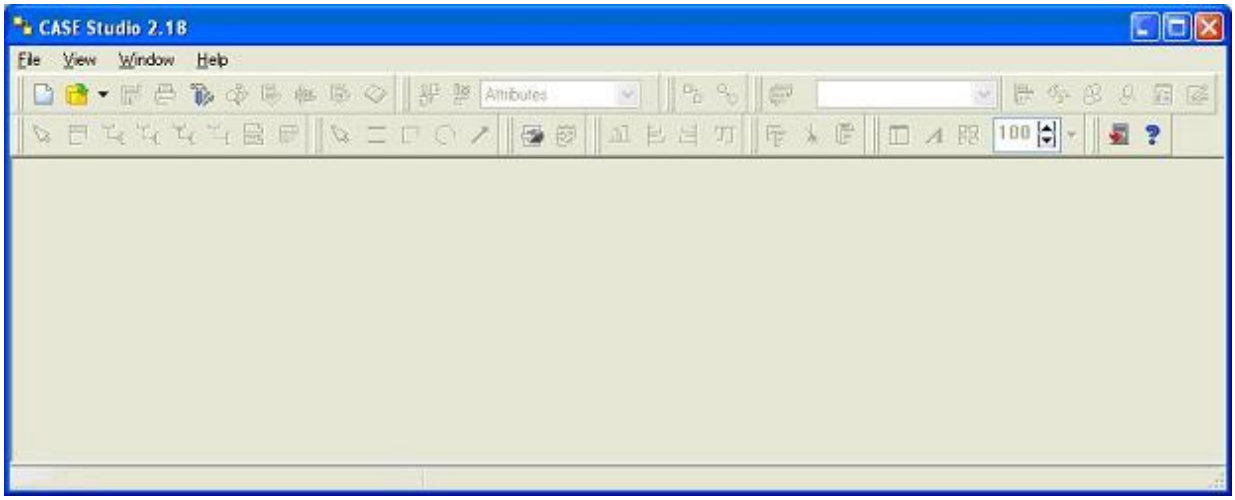
Case Studio 2.18 est un outil de modélisation de base de données. Il supporte les principales bases de données du marché (Access, DB2, Informix, Ingres, InterBase, SQL Server, MySQL, Oracle, Paradox, Sybase, PostgresSQL et bien d'autres). C'est un produit de la société CharonWare. Il se décline en deux versions, une version full et une version Lite. Les fonctionnalités principales sont décrites dans cet article mais il n'est toutefois pas exhaustif. Je vous invite à consulter le site officiel www.casestudio.fr pour de plus amples informations. Par rapport à ce qui est décrit ici, la version Lite est privée du reverse engineering, du gestionnaire de version, de la gestion des utilisateurs et du diagramme de flux de données. Vous pourrez trouver sur le site une version de démonstration. Cette version n'est pas limitée dans le temps mais bien au niveau de ses fonctionnalités.

Vous retrouverez également un lien vers Case Studio sur notre page outils sgbd.developpez.com/outils

II - Avertissement

Cet article n'est pas un tutoriel sur "Case Studio 2.18" mais un test du produit que je découvre avec vous. Nous allons tenter de décortiquer les fonctionnalités les plus importantes et de voir dans quelle mesure le produit est facile et efficace. J'estime qu'un bon moyen de tester la convivialité d'un produit est de partir directement à la découverte et de faire appel à l'aide ou la documentation quand le besoin s'en fait sentir. L'approche est évidemment différente quand il s'agit de l'apprentissage exhaustif d'un logiciel.

III - Premières impressions.



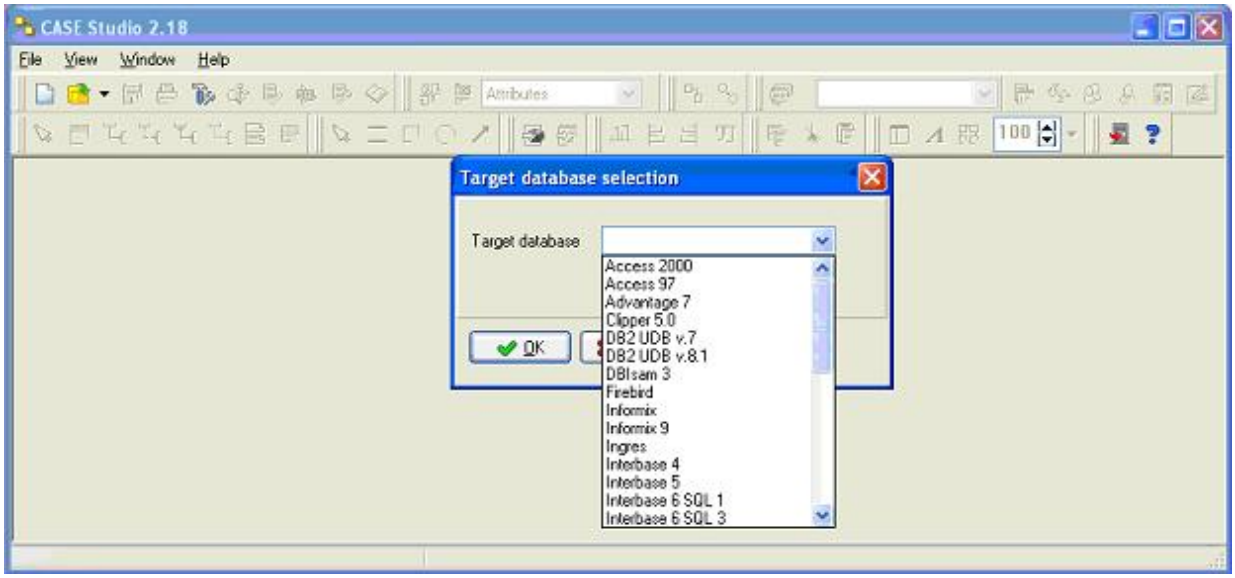
La toute première impression est plutôt favorable. L'interface est agréable et on est rapidement capable d'utiliser les principales fonctionnalités. Pour le placement des composants sur l'écran, il s'agit de cliquer sur le composant voulu dans la barre d'outils et ensuite de cliquer dans la fenêtre à l'endroit voulu. Il est inutile de maintenir le bouton de la souris enfoncé. Les déplacements dans la fenêtre se font par cliquer, glisser.

IV - Une première approche.

Le mieux pour tester une application reste encore d'effectuer un exercice. Comme exercice, nous allons faire simple. Il s'agit de décrire une base de données de produits finis. La base de données doit également fournir la composition pour chaque produit fini. Pour le cas qui nous occupe, nous considérerons que le produit fini est construit uniquement sur base de composants simples.

V - Création du modèle

Dans Case Studio, nous créons un nouveau modèle. Nous devons d'entrée choisir le gestionnaire de base de données qui servira de support. Cela peut paraître surprenant de lier d'emblée un modèle à un gestionnaire de base de données et non de définir celui-ci lors de la génération du code mais cela permet d'optimiser l'interface de manière à permettre des options différentes selon le serveur. Une fonction de migration nous permettra le cas échéant de changer de gestionnaire de base de données.

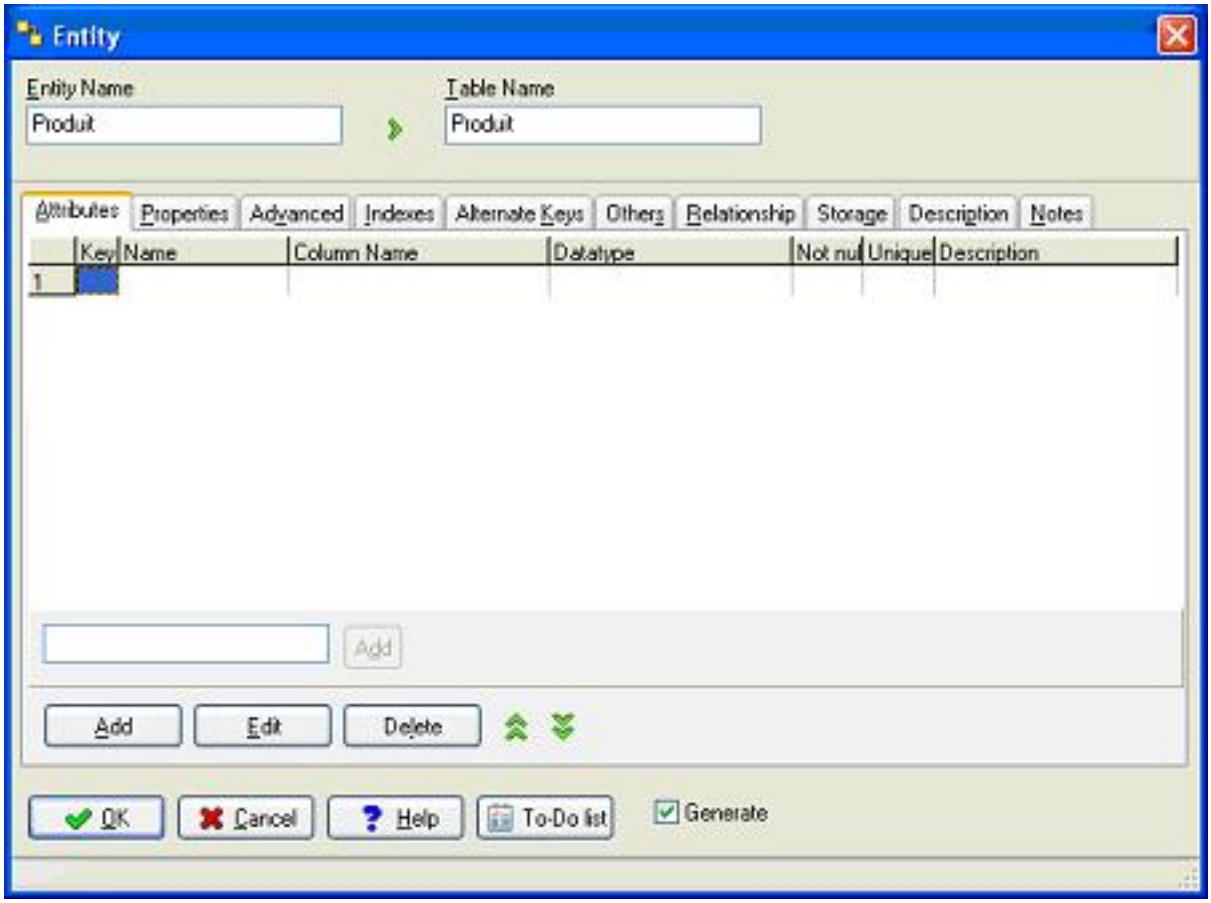


Ici j'ai choisi MS SQL 2000. Une nouvelle fenêtre est maintenant ouverte et nous pouvons constater qu'il est possible de définir des modèles de type ERD (Entity Relationship Diagram) et des modèles de type DFD (Data Flow Diagram). Nous reviendrons ultérieurement sur les modèles DFD.

VI - Diagramme ERD

A - Création d'une table

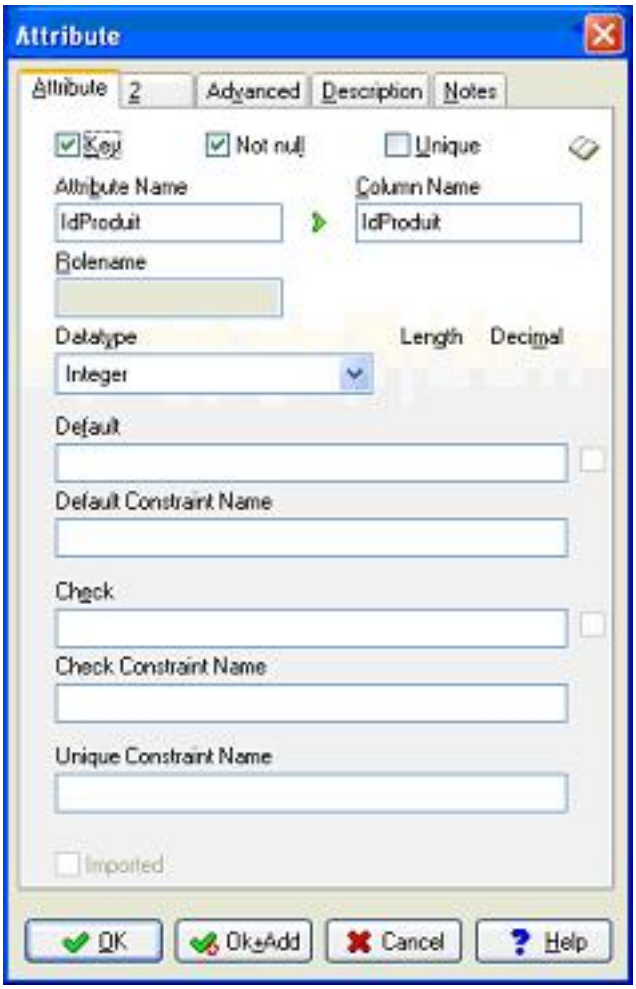
J'ajoute une nouvelle entité et directement j'ouvre la fenêtre de modification (clique droit, "Edit entity" ou simplement un double clique sur la table).



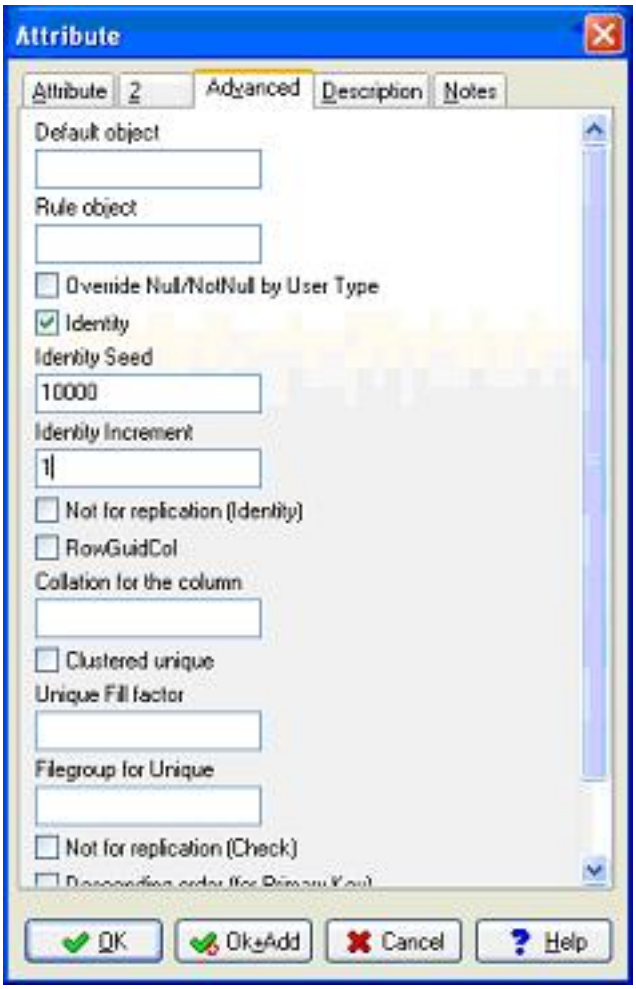
Nous sommes maintenant dans la fenêtre permettant de définir une table.

Pour définir les champs d'une table, il suffit d'ajouter des attributs via le bouton "Add". Les boutons "Edit" et "Delete" permettent respectivement de modifier et de supprimer l'attribut sélectionné.

Nous allons commencer par ajouter la référence du produit. Je choisis ici une référence numérique automatique.



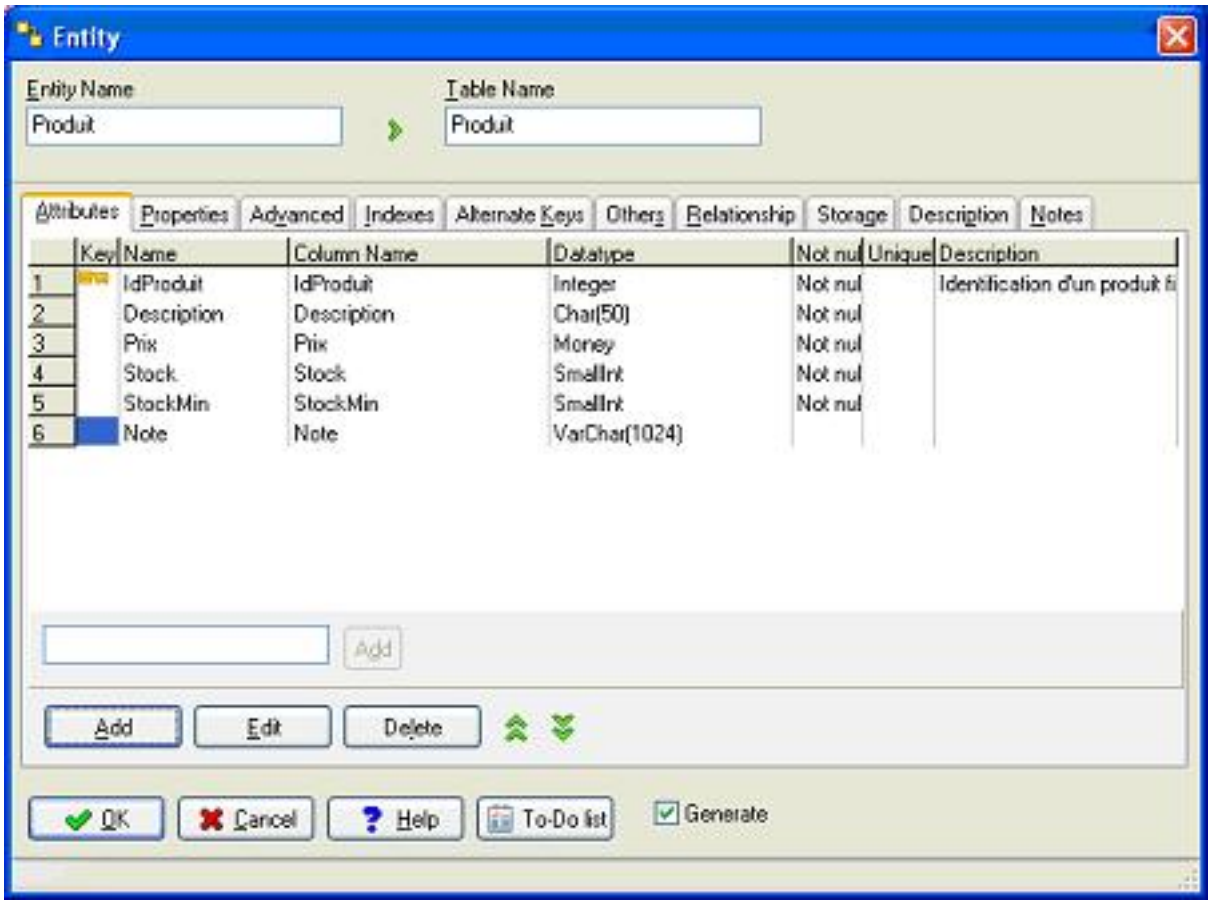
Notez au passage qu'il suffit de choisir "key" pour ajouter le champ à la clé primaire.



Dans l'onglet "Advanced", nous pouvons définir que le champ est de type "Identity" (autonumber). Le contenu de l'onglet "Advanced" dépend du gestionnaire de base de données précédemment choisi. L'onglet "description" défini le commentaire associé qui sera inscrit dans la base de données alors que l'onglet "Note" permet d'ajouter des informations destinées uniquement aux utilisateurs de Case studio.

Bien que nous soyons sur le troisième onglet, le fait de faire "Ok+Add" nous restitue une fenêtre vierge correctement positionnée sur l'onglet "Attribute". Si cela semble trivial, ce n'est pourtant pas toujours le cas. En revanche, j'aurai apprécié un système permettant de dupliquer la définition d'un champ pour les tables contenant un grand nombre de champs similaires si ce n'est le nom.

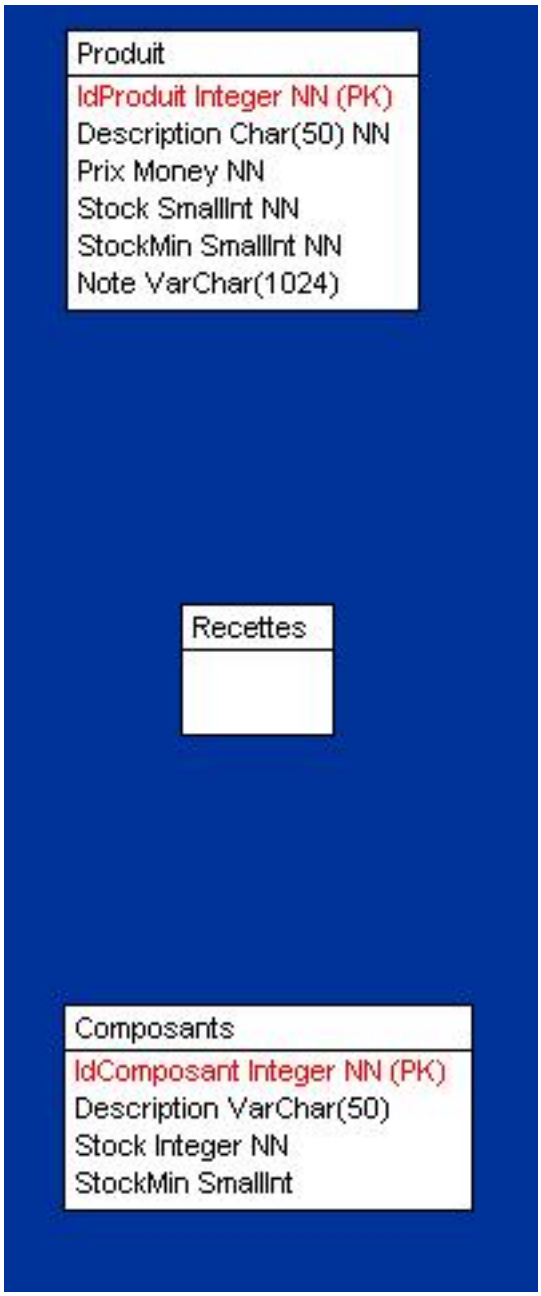
Comme vous pouvez le voir, la structure définie est clairement présentée dans la fenêtre.



La table (Entity) est maintenant affichée sur le bureau. Seul la clé primaire définie est représentée. En fait, il existe 4 modes d'affichage différents auxquels il faut encore ajouter les options "Physical view" et "Display index".

Notre modèle étant simple, nous pouvons nous permettre de choisir en permanence l'affichage le plus détaillé. Je choisis donc "Attributes" et "Physical view". A ce stade, "Display index" n'apporte rien mais vous pouvez aussi le sélectionner.

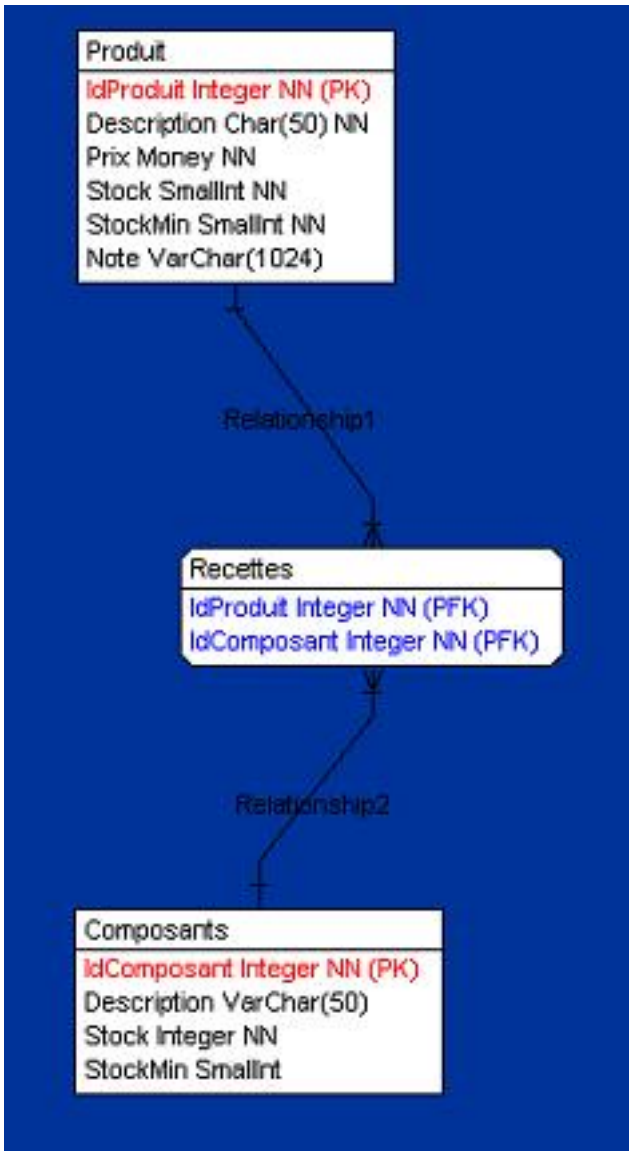
Après l'ajout des tables "Composants" et "Recettes", notre diagramme se présente comme ceci :



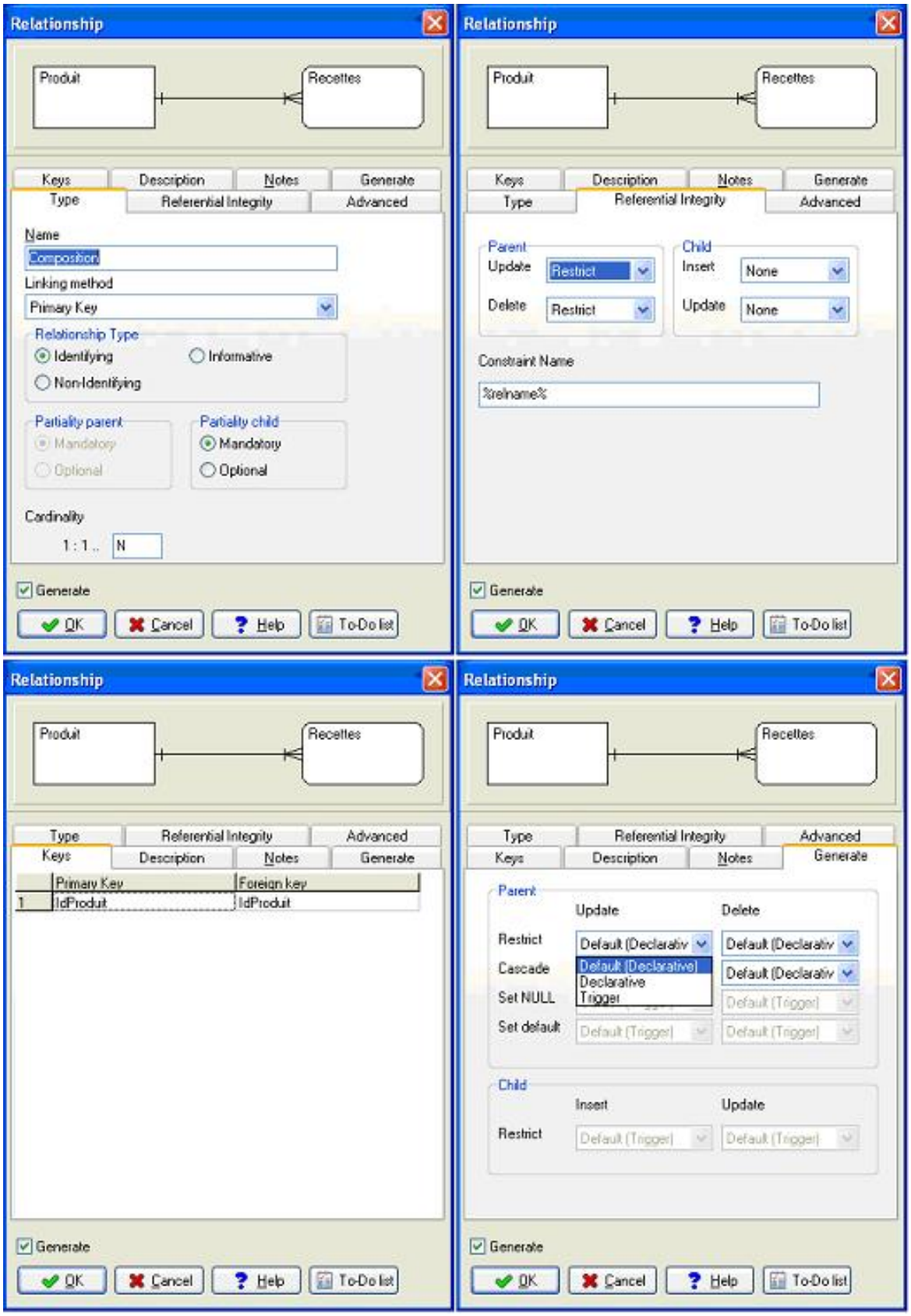
Vous remarquerez que la table Recettes est actuellement vide.

B - Création des relations

Avec l'icône  de la barre d'outils, je crée une relation identifiée de "Produits" vers "Recettes" et une de "Composants" vers "Recettes". La clé primaire, grâce au choix d'une relation identifiée, et les clés étrangères sont automatiquement créées. Les champs nécessaires sont ajoutés dans la table "Recettes".



En double cliquant sur une relation, nous obtenons les propriétés de celle-ci.



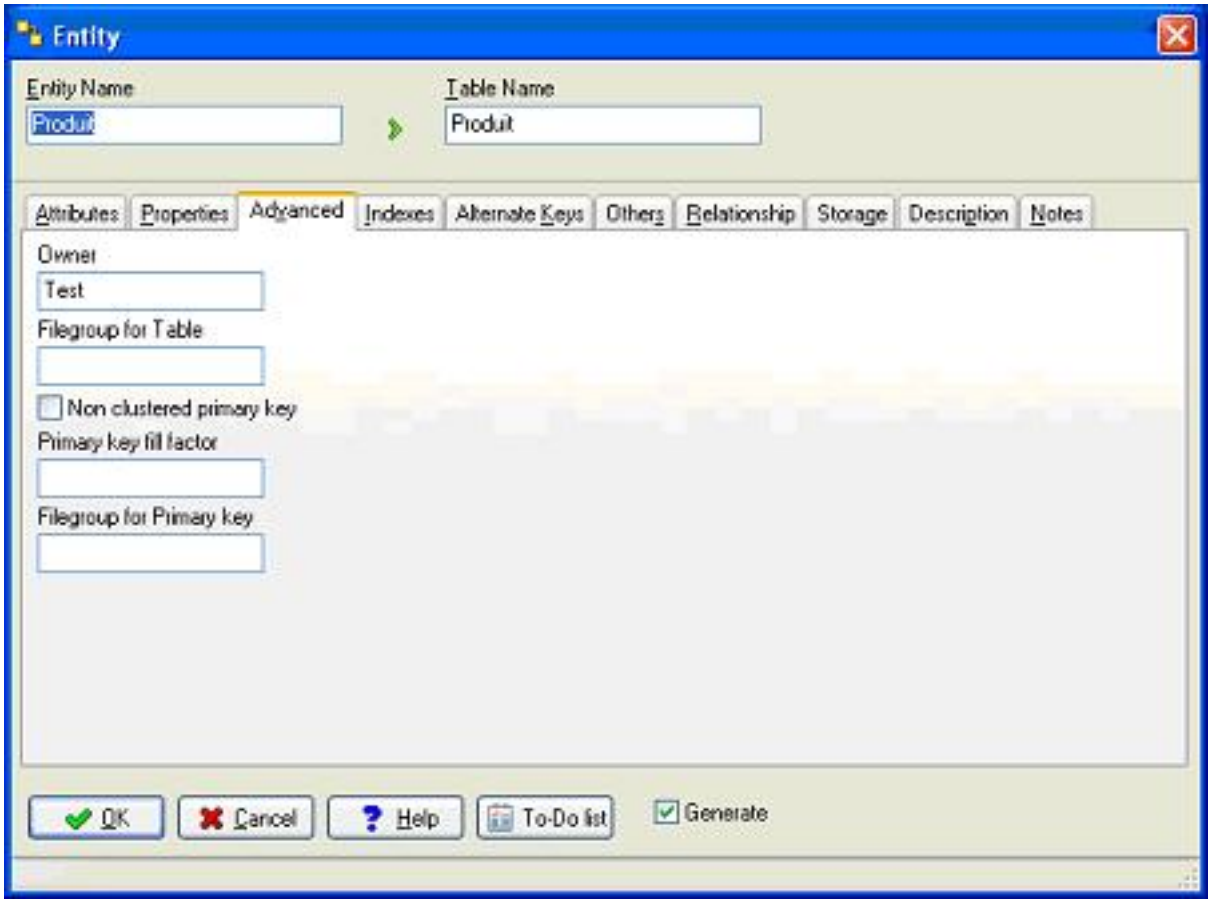
Il est alors possible de modifier finement la relation. Vous pouvez ainsi insérer 4 types de relations :
- Identifiée, la clé primaire migre vers la table enfant et participe à sa clé primaire.
- Non identifiée, la clé primaire migre vers la table enfant mais sans participer à sa clé primaire.
- Informative, définit l'existence d'une relation mais sans déterminer une clé étrangère.
- M-N, va automatiquement générer la création d'une table intermédiaire réalisant la relation.

Jusqu'ici j'ai toujours parlé d'une relation avec la clé primaire du parent mais il est également possible de faire une relation sur une clé secondaire ou sur un champ unique.

Il est aussi possible de créer une relation d'une table sur elle-même. Dans ce cas, vous ne pourrez évidemment pas choisir une relation de type identifiée.

C - Module de vérification

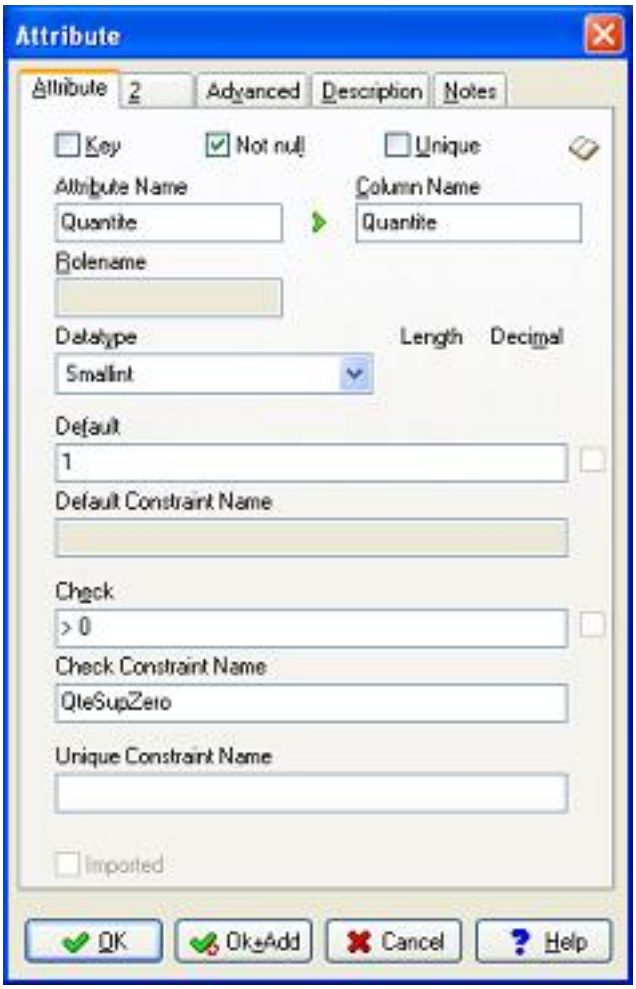
A ce stade si nous utilisons le module de vérification du modèle, nous allons recevoir des "warning". En effet pour les tables sous SQL Server, nous devons définir un propriétaire (owner) pour chaque table (entité).



D - Contraintes

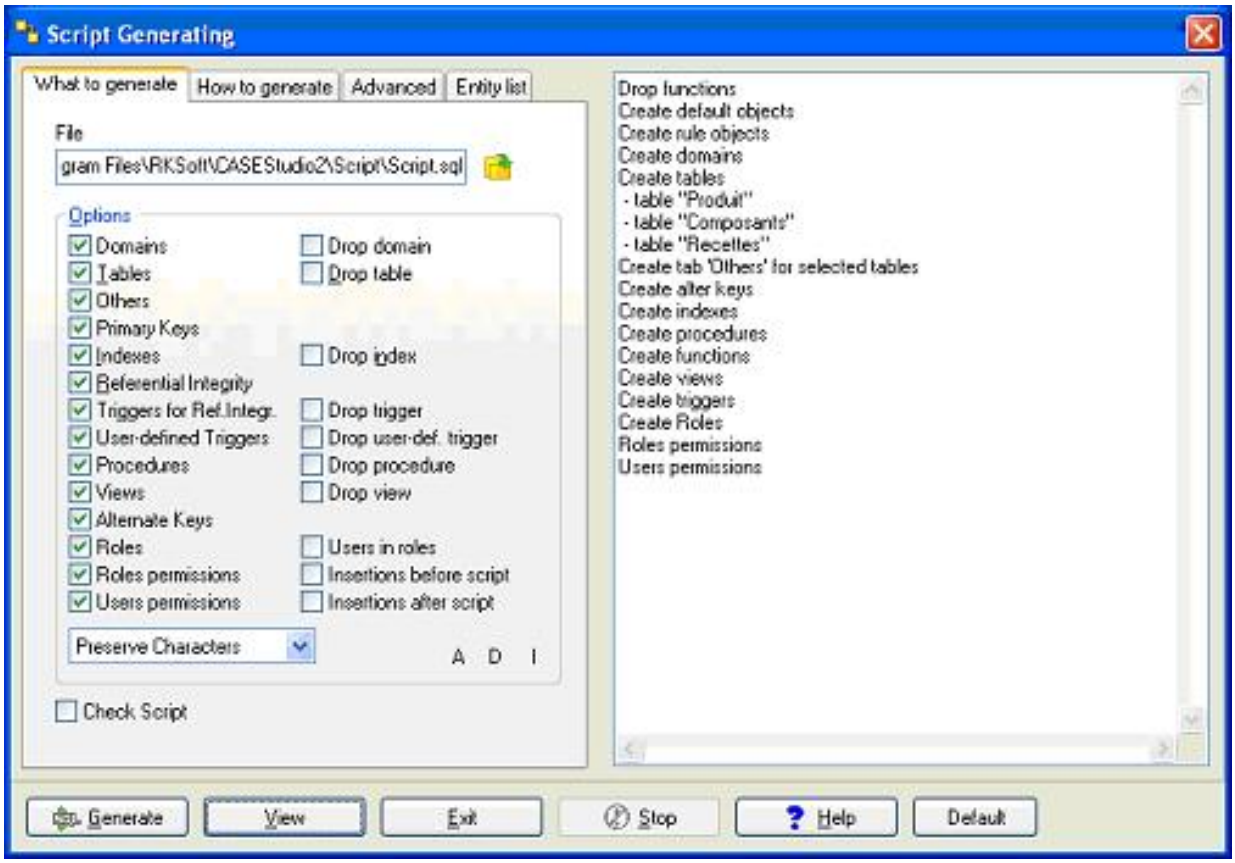
Outre les contraintes des relations déjà vues plus haut, il est possible de définir des contraintes sur les champs. Dans l'exemple ci-dessous, on définit une contrainte nommée "QteSupZero" qui impose un nombre supérieur à

zéro.

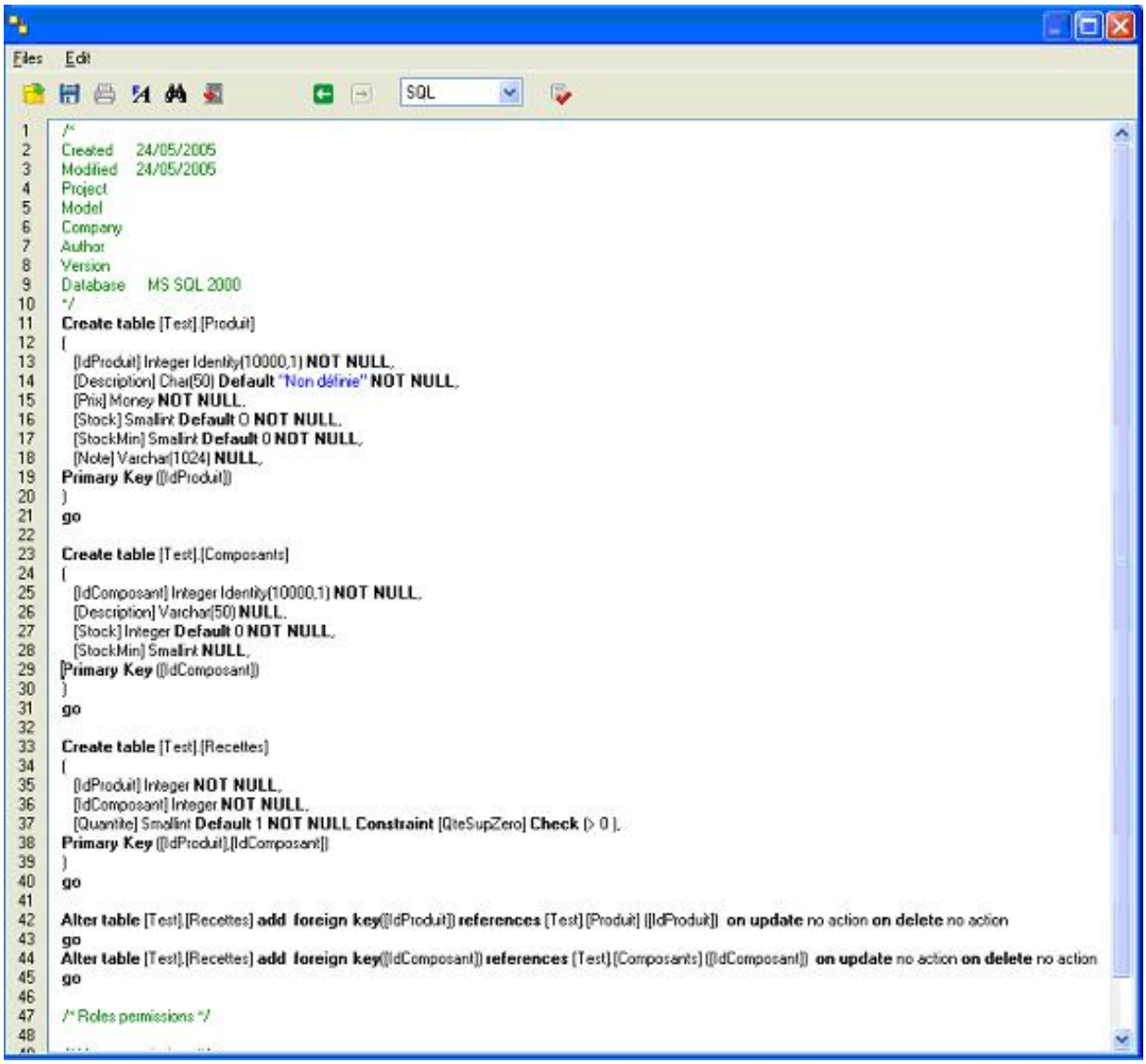


E - Génération du script

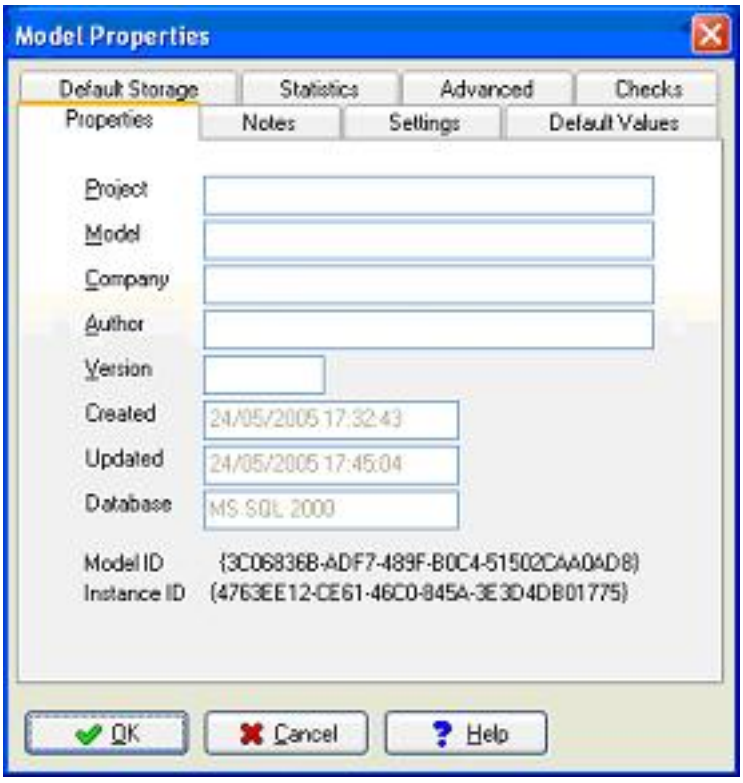
Nous pouvons maintenant passer à la génération automatique du script.



Voyons à quoi ressemble le script généré.

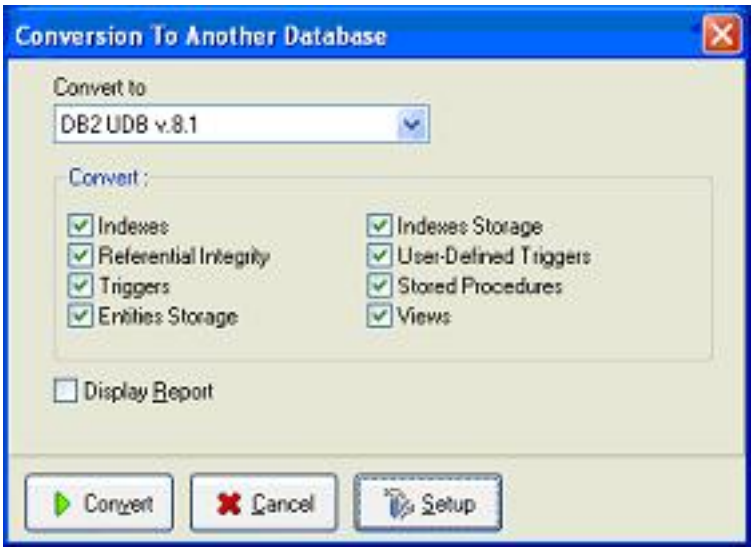


Le script peut être directement modifié dans le viewer et ensuite enregistré pour un usage ultérieur. Les commentaires sont automatiquement complétés grâce aux propriétés du modèle.

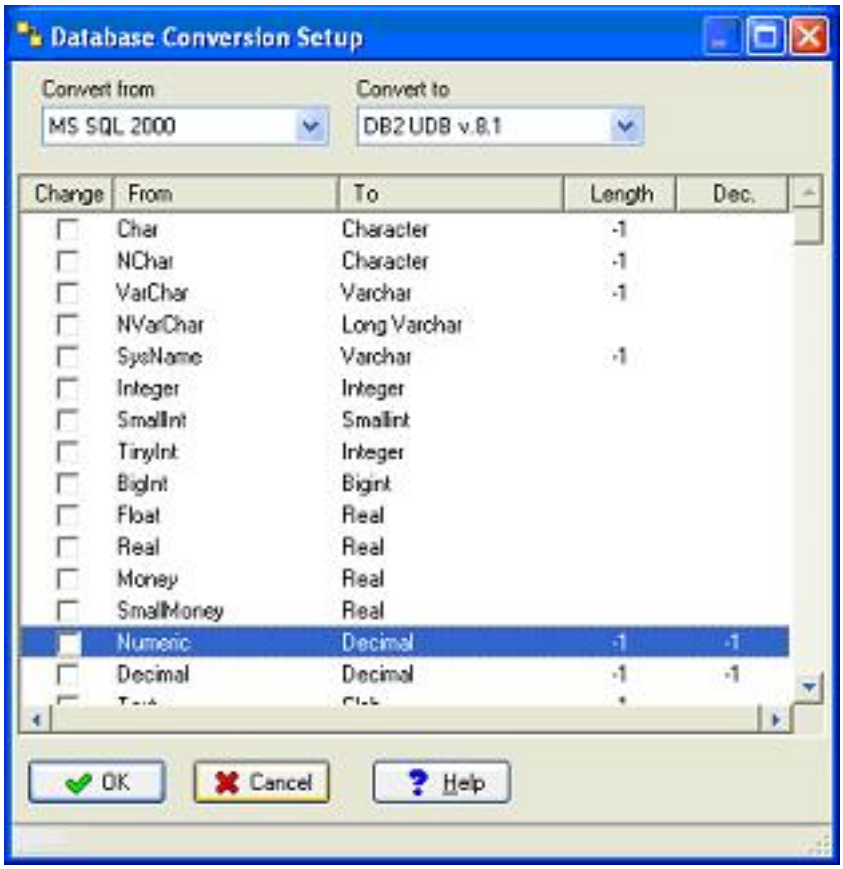


F - Conversion vers un autre gestionnaire de base de données

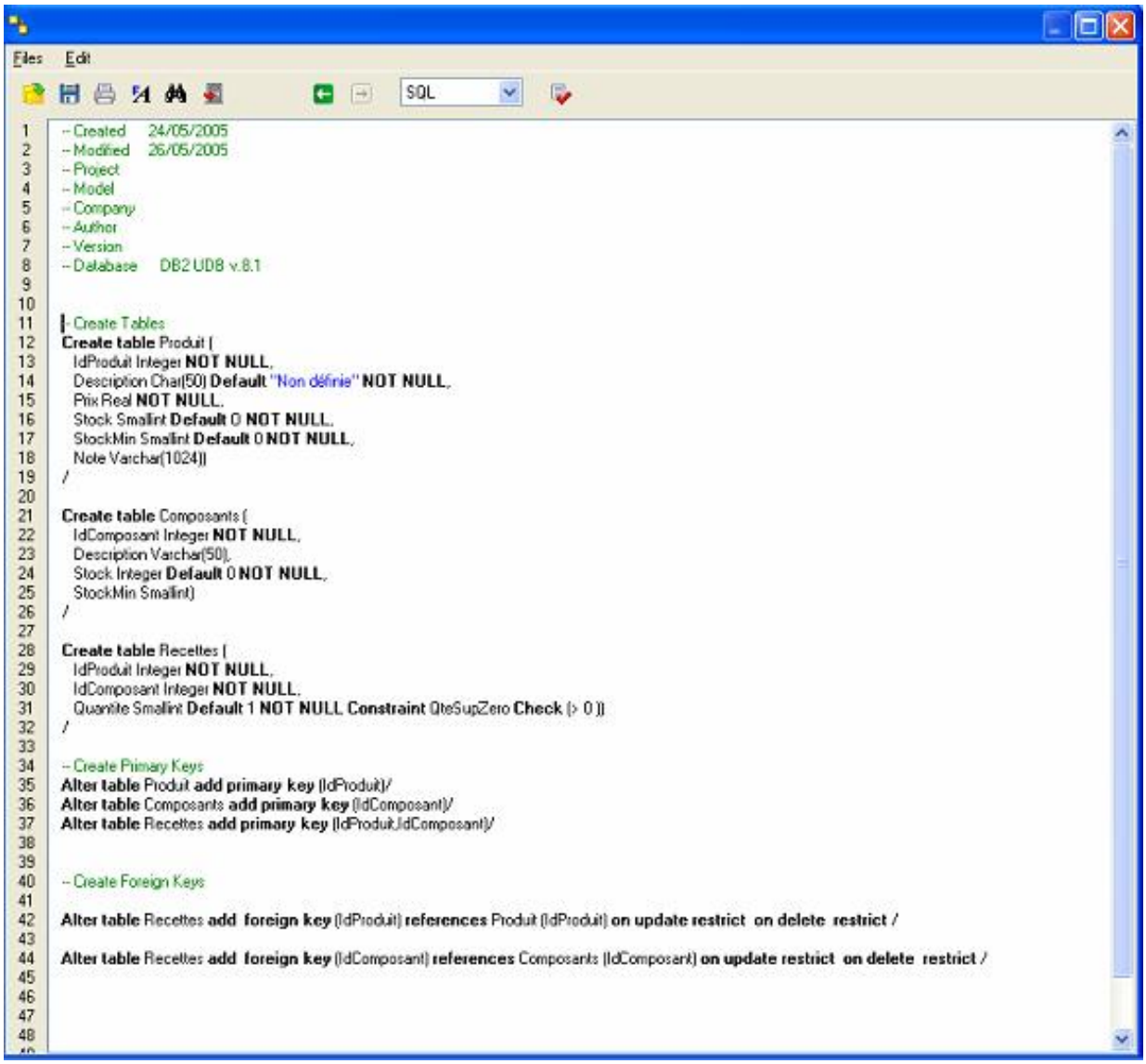
Si vous souhaitez changer de gestionnaire de base de données, le logiciel vous fournit un module de conversion.



Le setup permet de modifier la conversion des types de données.



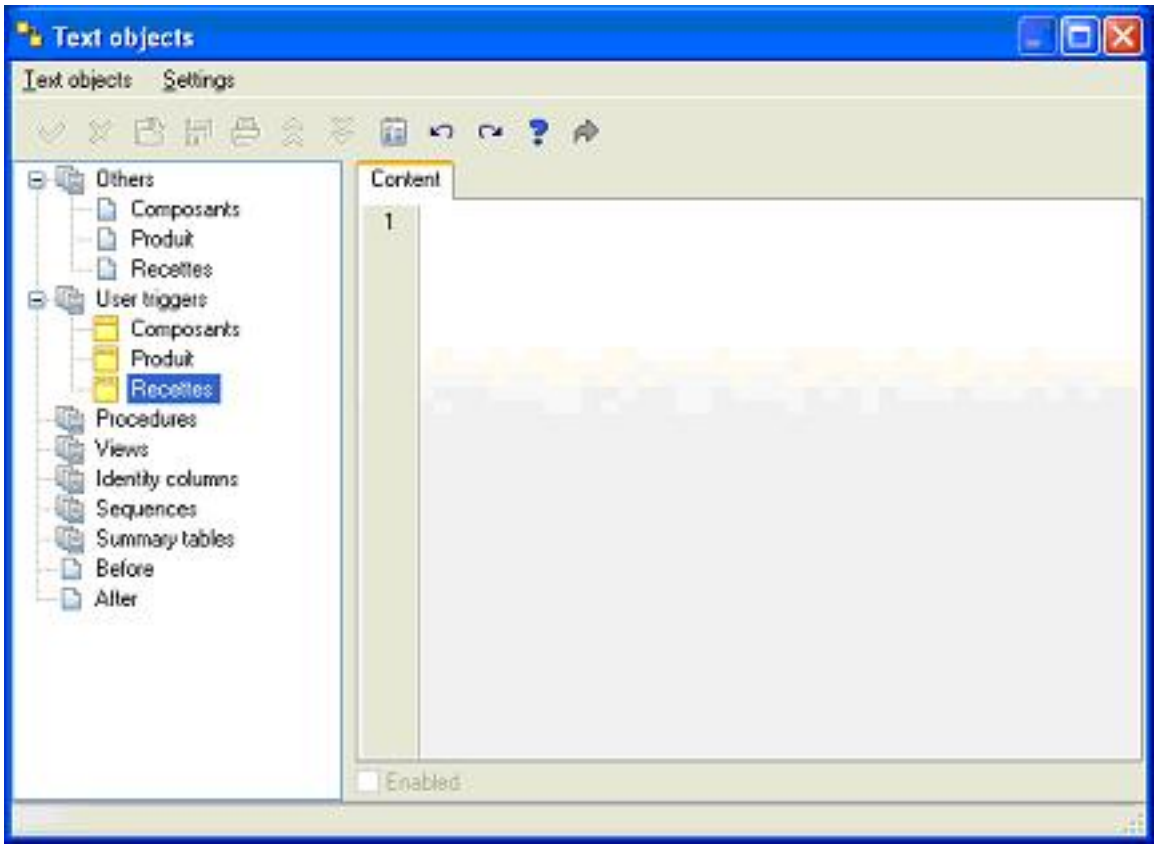
Dans notre exemple, je fais la conversion de MS-SQL vers DB2. Si nous générons le script, il donne maintenant :



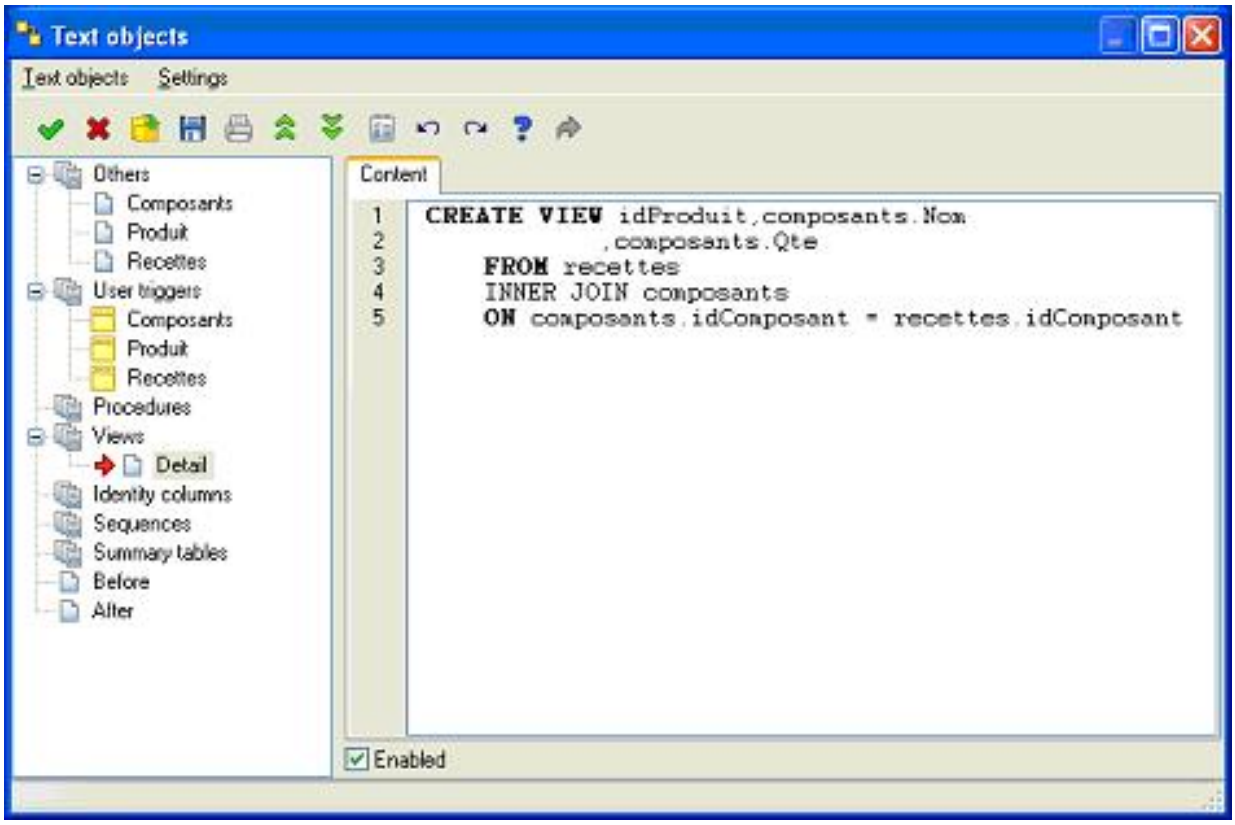
Le script a clairement été adapté pour refléter la syntaxe DB2. Il faut tout de même faire attention aux fonctions spéciales. Si nous regardons l'onglet "Advanced" du champ "IdProduit", nous voyons que la valeur initiale de l'autonumber n'est plus 10000 mais 1 ! En fait rien de bien étonnant puisque la gestion de ce type de champs est différente d'une base à l'autre. C'est en fait le résultat du choix de l'éditeur d'offrir les options les plus poussées relatives à un gestionnaire de base de données. S'il avait pris l'option de rester plus générique, l'écueil aurait pu être évité. Pour ma part, je suis favorable à l'option de l'éditeur dans la mesure où l'on change rarement de gestionnaire de base de données mais cependant il peut être intéressant de disposer des options avancées qui ne pourraient pas être présentes dans un modèle générique.

G - Vues, Procédures stockées, trigger, ...

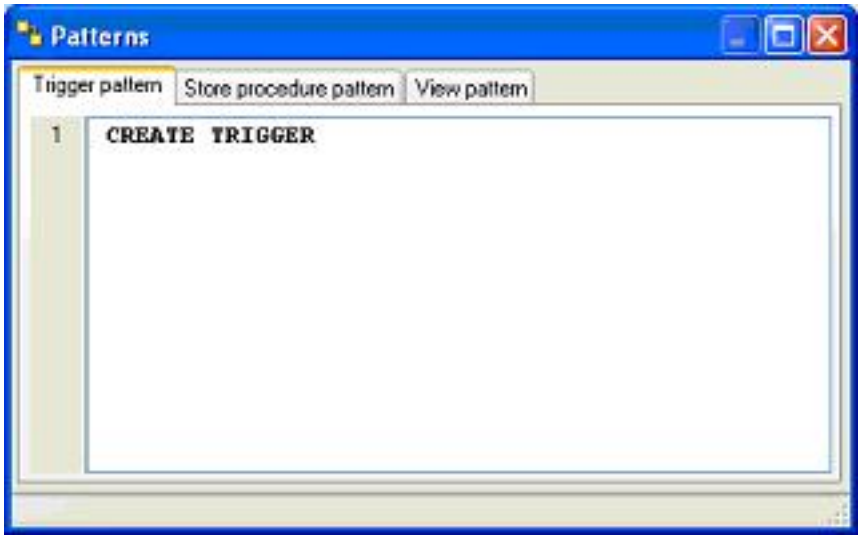
Différents objets peuvent également être ajouté aux modèles. Il s'agit d'objet de type texte. Cette option est disponible dans le menu modèle.



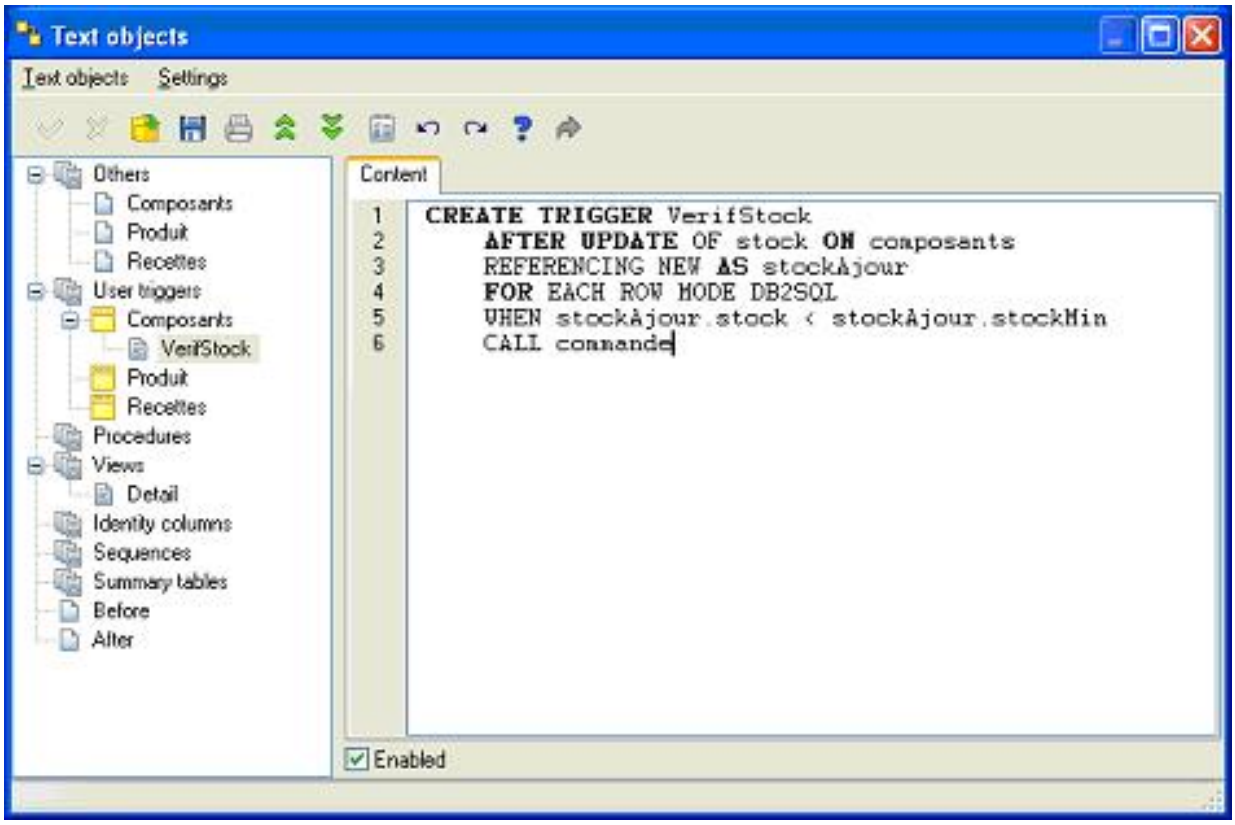
Par exemple, pour ajouter une vue, il suffit de cliquer sur "Views" avec le bouton droit et de choisir "Add". Vous donnez un nom, vous tapez votre code dans la partie droite de l'écran et vous cliquez sur "Ok" dans le menu ou sur le bouton équivalent dans la barre d'outils et la vue est ajoutée.



Pour vous faciliter le travail, il est possible d'utiliser un "pattern".



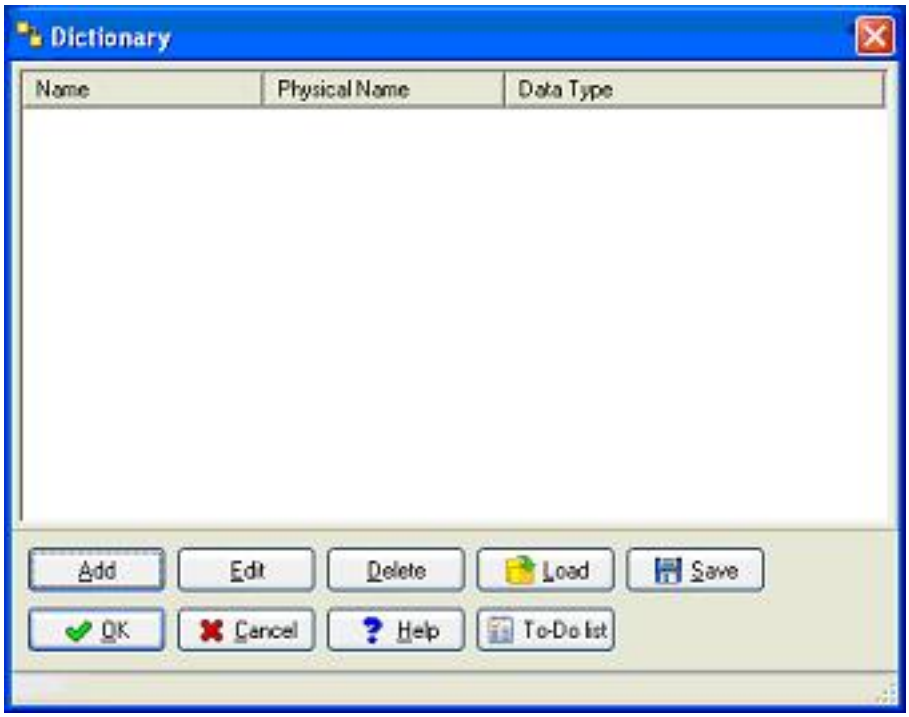
Pour un trigger, il faut cliquer avec le bouton droit sur la table sous l'option "trigger" et choisir "Add trigger". Il suffit alors de procéder comme pour la vue.

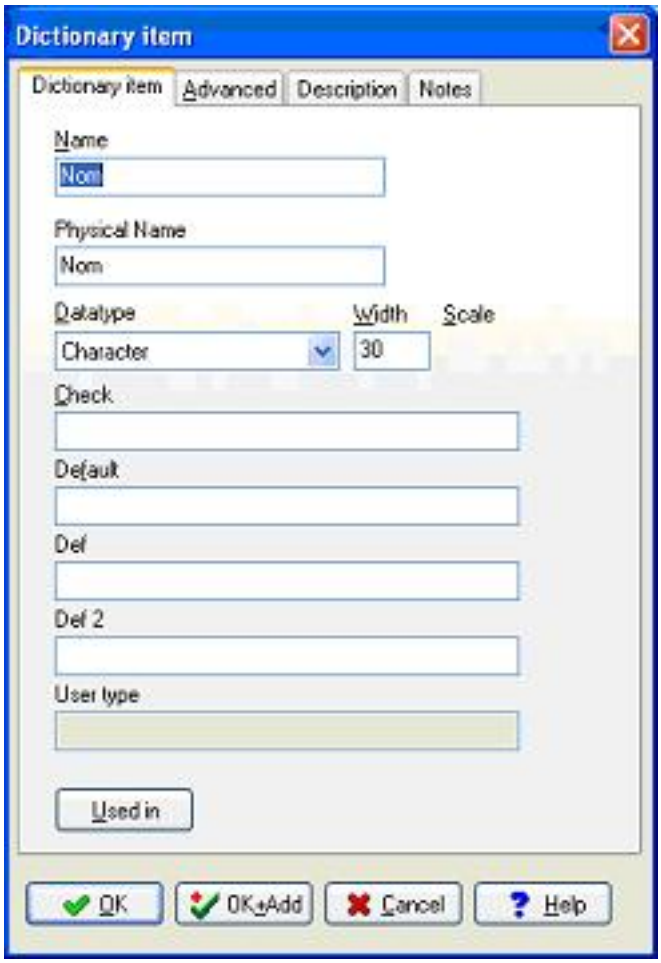


Le code sera évidemment ajouté lors de la génération du script. Bien sûr cette partie ne bénéficie d'aucune conversion en cas de changement de gestionnaire de base de données.

H - Les types utilisateurs.

Il est également possible de définir vos propres types de données. Cette fonctionnalité est disponible depuis le menu "Dictionary"

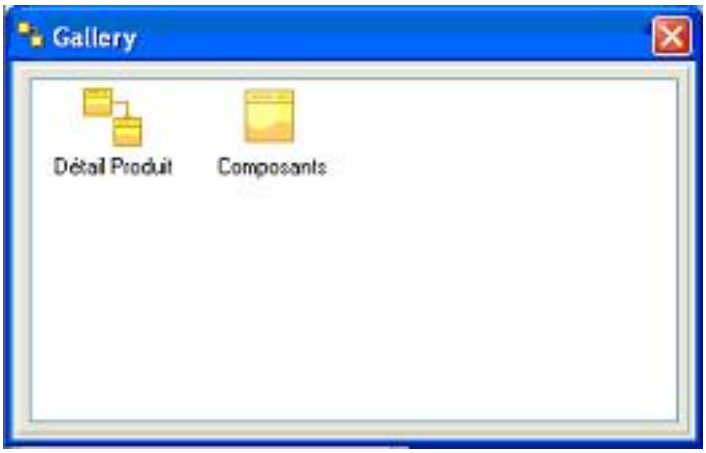




Le type ainsi défini est alors disponible lors de la définition des champs des tables et donne lieu à l'écriture du code nécessaire lors de la génération du script.

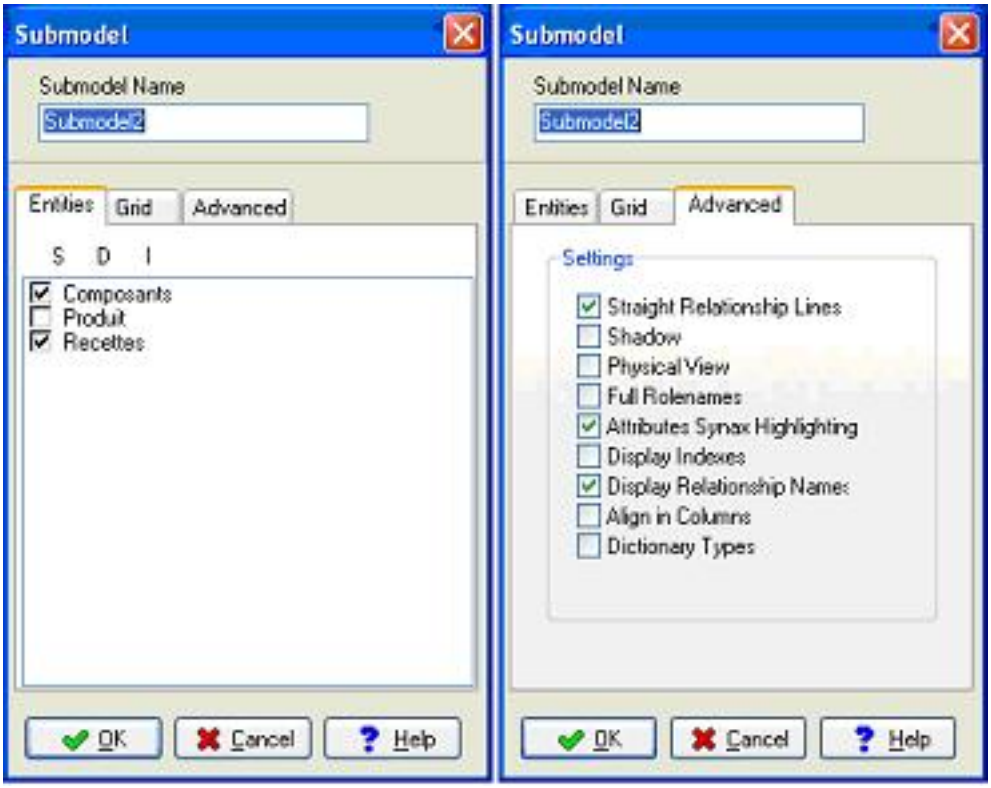
I - La galerie.

La galerie permet de stocker des tables ou des extraits de schéma qui pourront alors être réutilisés par simple copier, coller. Cette fonction est très utile pour augmenter la productivité car nous avons tous des tables qui sont régulièrement utilisées. Pour ajouter un élément à la galerie, il suffit de sélectionner la partie voulue dans le graphique et via le clique sur bouton droit, de choisir "Add to gallery" Pour les réutiliser, il suffit ensuite de faire un drag&drop.



J - Sous modèles.

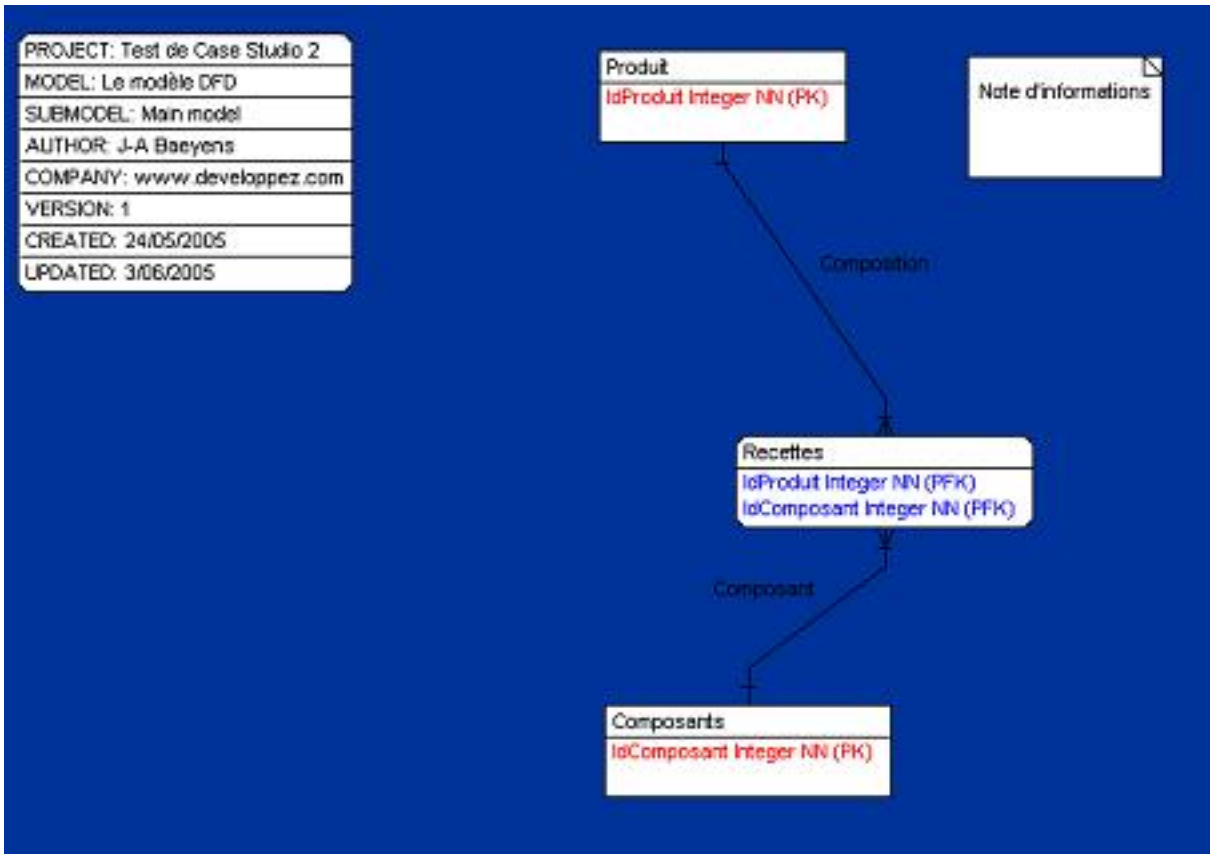
Il est possible pour des projets travaillant sur un grand nombre de tables, de créer des sous modèles. Cela permet de simplifier le travail et la vision du diagramme.



Après avoir créé votre sous modèle, vous pouvez choisir d'afficher uniquement le sous modèle ou le diagramme en entier. Vous pouvez choisir pour le diagramme de n'afficher que les entités (avec ou sans les clés) et pour le sous modèle d'afficher un maximum d'information. Le diagramme général est ainsi nettement simplifié. Ceci est rendu facile car Case studio conserve le paramètre d'affichage pour chacun des modèles. On peut regretter que les concepteurs n'aient pas poussé l'idée jusqu'à permettre l'affichage des sous modèle comme une entité particulière dans le diagramme général.

K - Informations complémentaires dans le diagramme.

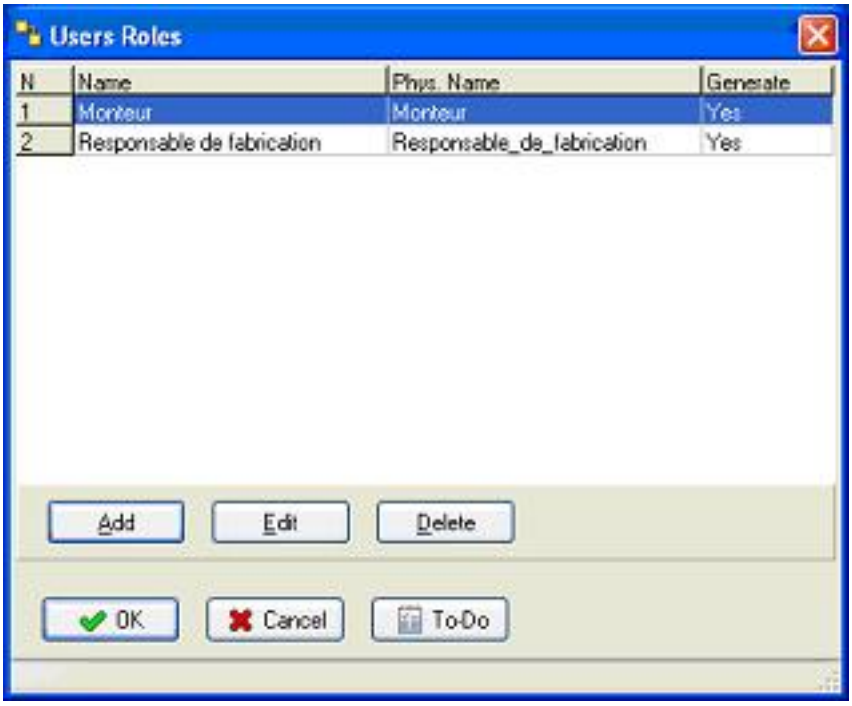
Il est possible d'introduire une étiquette reprenant les données du modèle mais également d'inclure des notes.

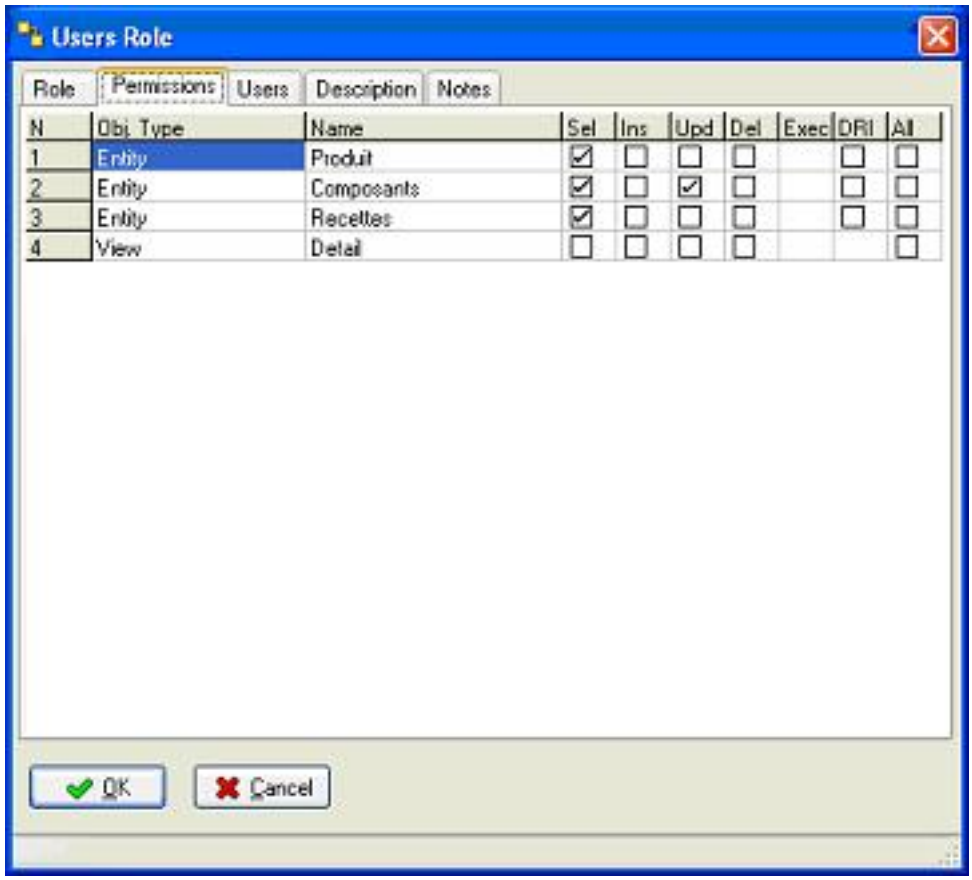


Il est dommage que les notes ainsi introduites ne puissent être liées visuellement à un des composants du diagramme. De même, il n'y a pas de possibilité d'afficher sous cette forme les notes introduites au niveau des différents composants du modèle. Une option sous forme d'une case à cocher aurait pu être prévue à cet effet.

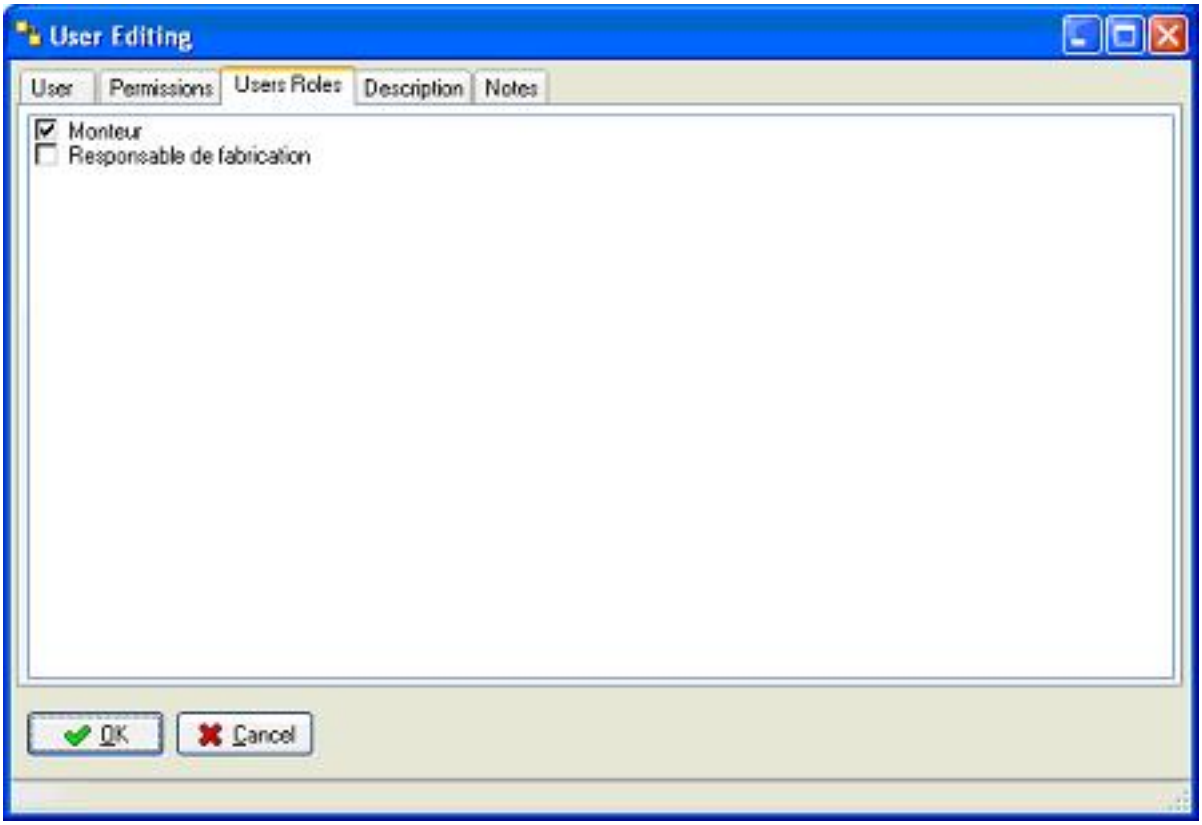
VII - Gestion des rôles et des utilisateurs

Il est possible également de réaliser la gestion des rôles et des utilisateurs depuis Case studio. Les rôles sont en principe stables, il est donc très intéressant de les inclure dans le schéma car cela permet d'obtenir un script complet.



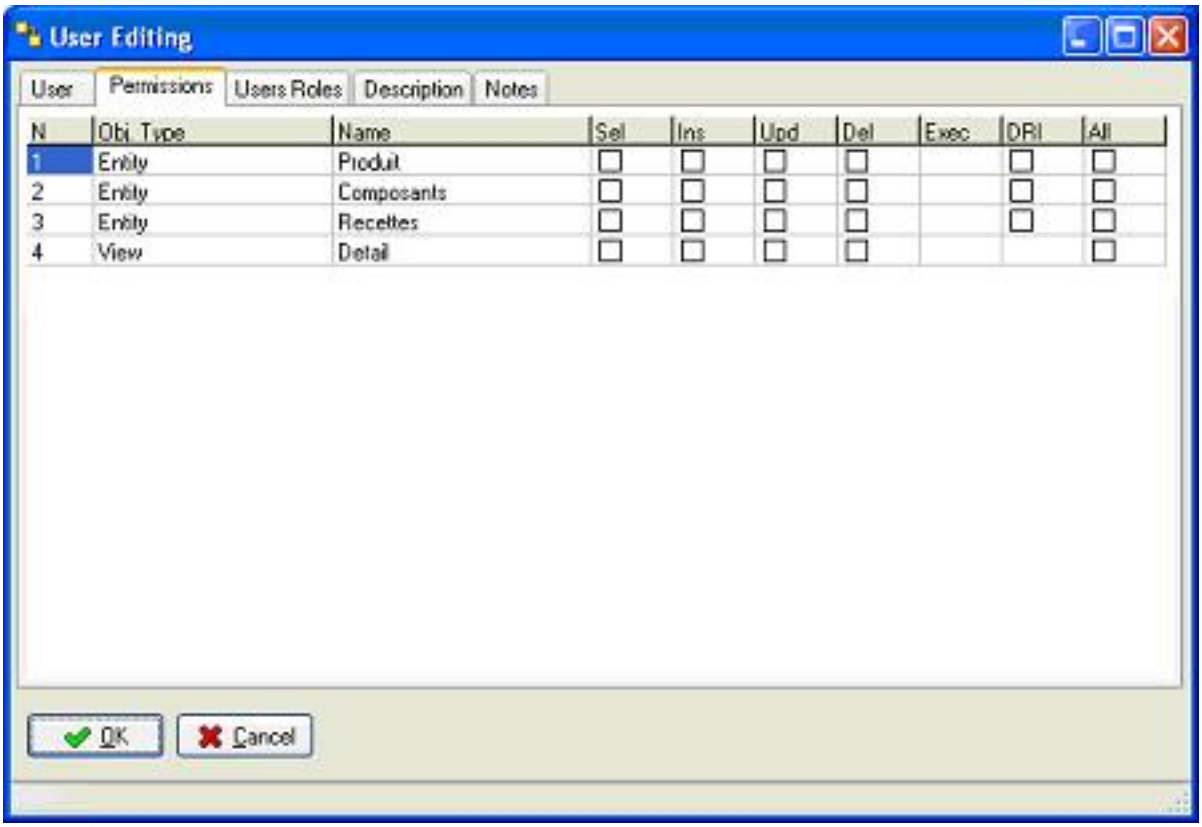


Vous pouvez même inclure les utilisateurs individuellement. Evidemment la liste des utilisateurs varie fortement dans le temps. On peut donc se poser la question de savoir si oui ou non la liste doit être gérée depuis case studio.

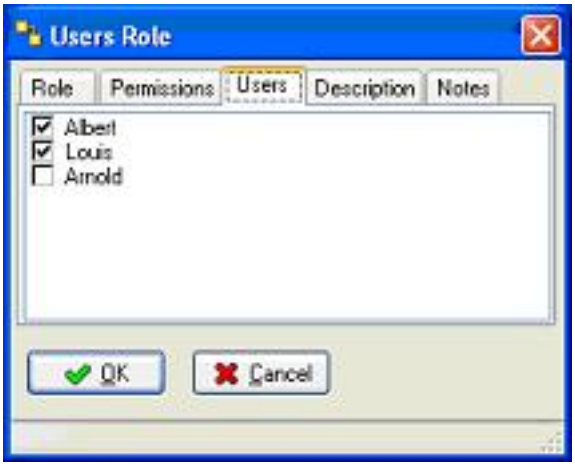


Comme vous pouvez le voir dans l'image ci-dessus, vous pouvez bien sûr associer un utilisateur à un ou des rôles

mais vous pouvez également lui attribuer des droits individuellement.

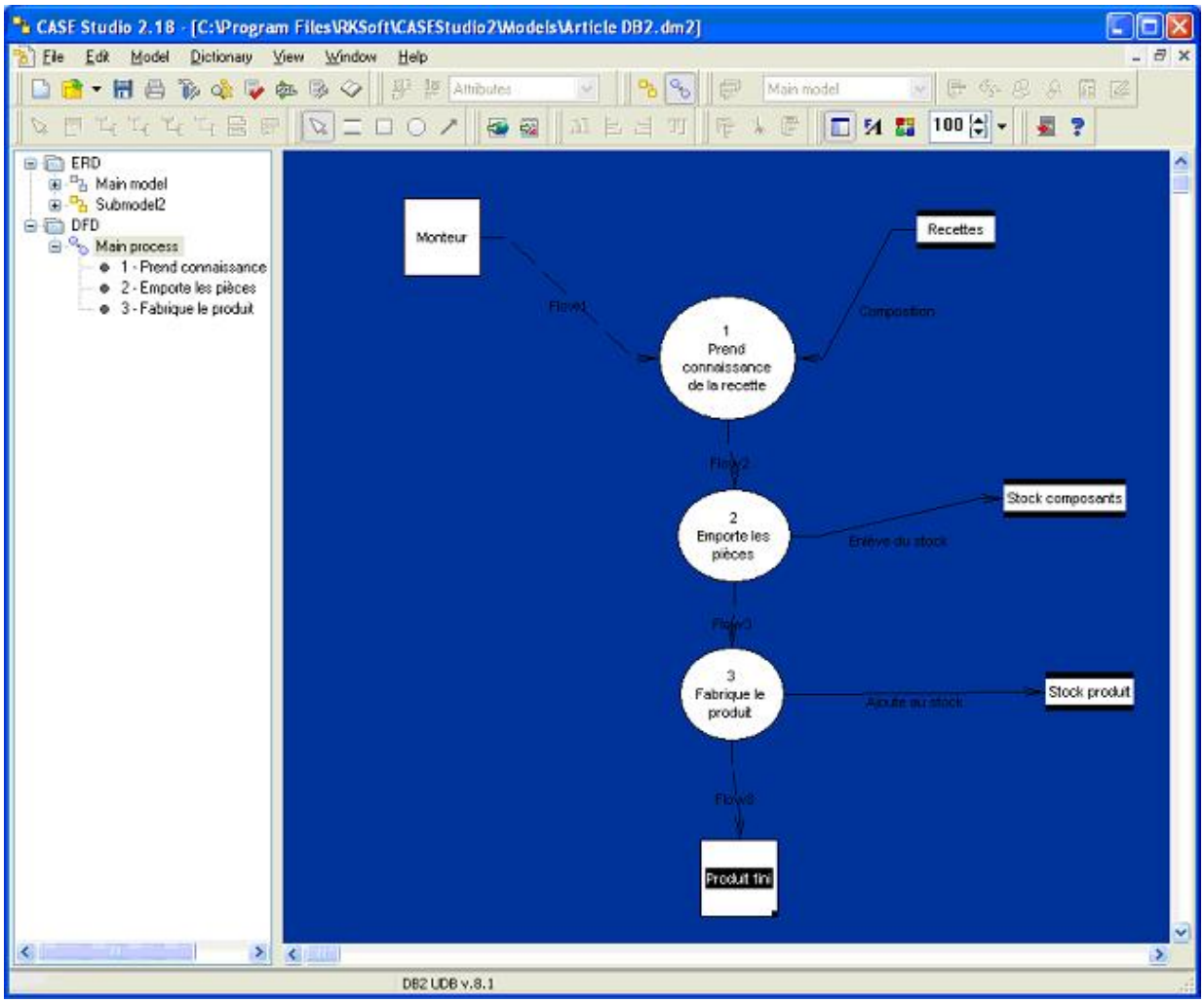


Quand les utilisateurs sont créés, vous pouvez également leurs attribuer un rôle depuis la fenêtre du rôle.



VIII - Diagramme DFD

Ce type de diagramme ne m'est pas très familier. Je vais donc juste en dessiner un et vous le présenter.

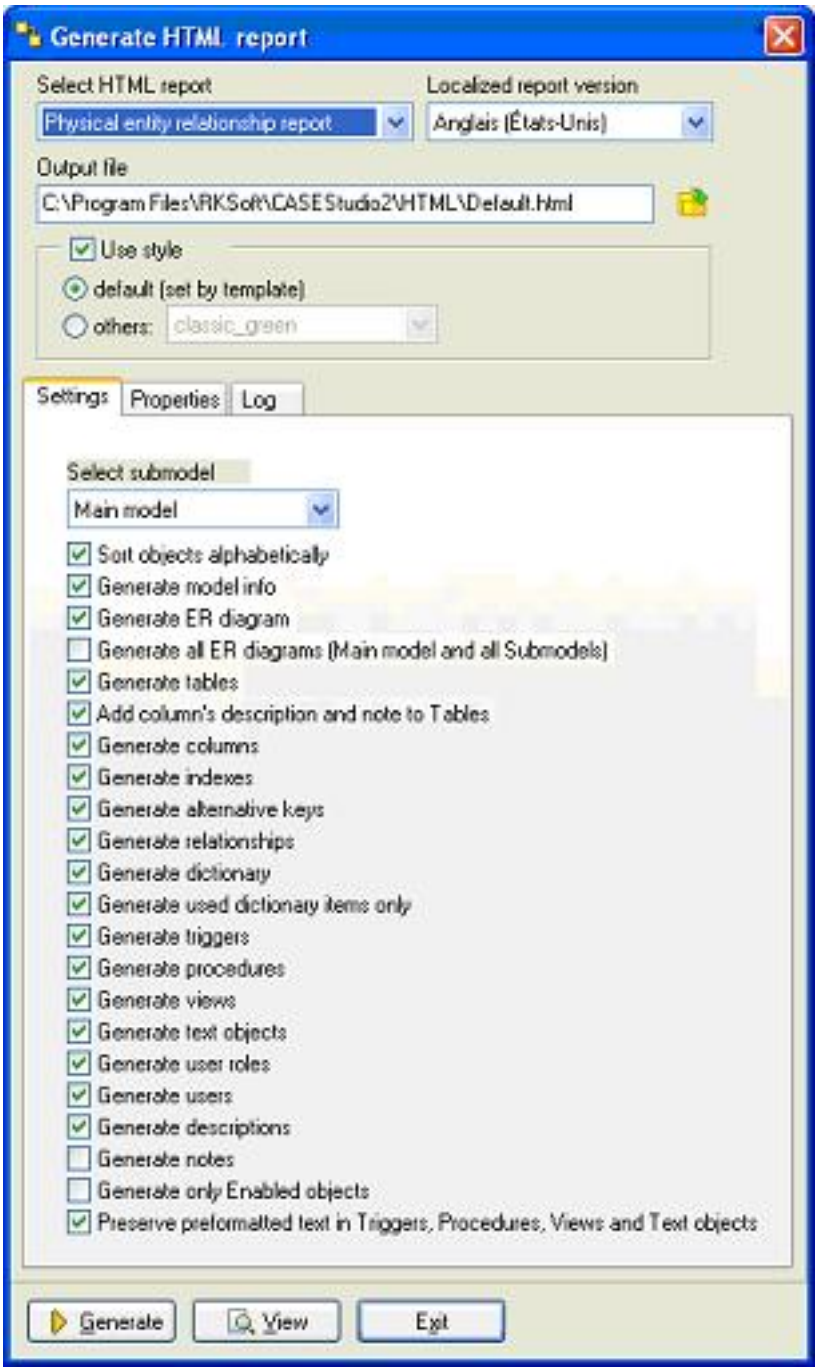


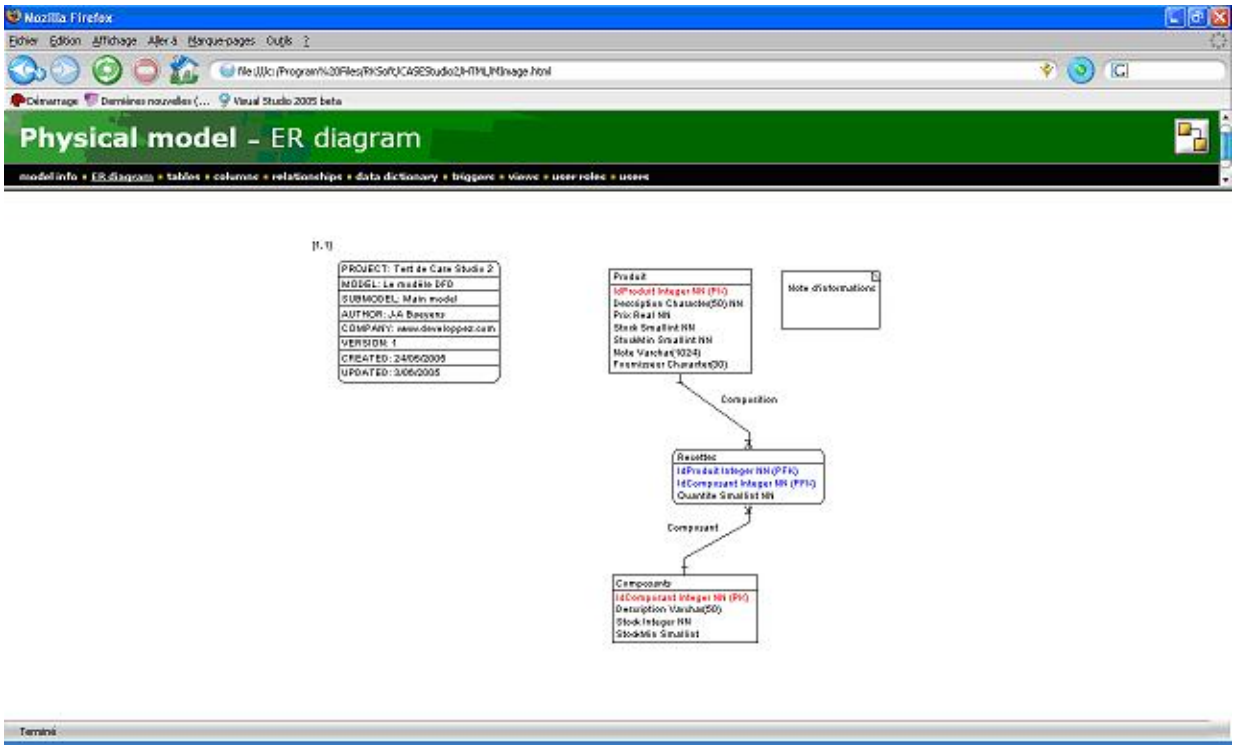
Ce diagramme a bien sûr son intérêt mais je l'attendais plutôt dans un outil d'analyse que dans un outil de design de base de données. Cela reste toutefois un plus pour celui qui ne dispose pas d'un autre outil.

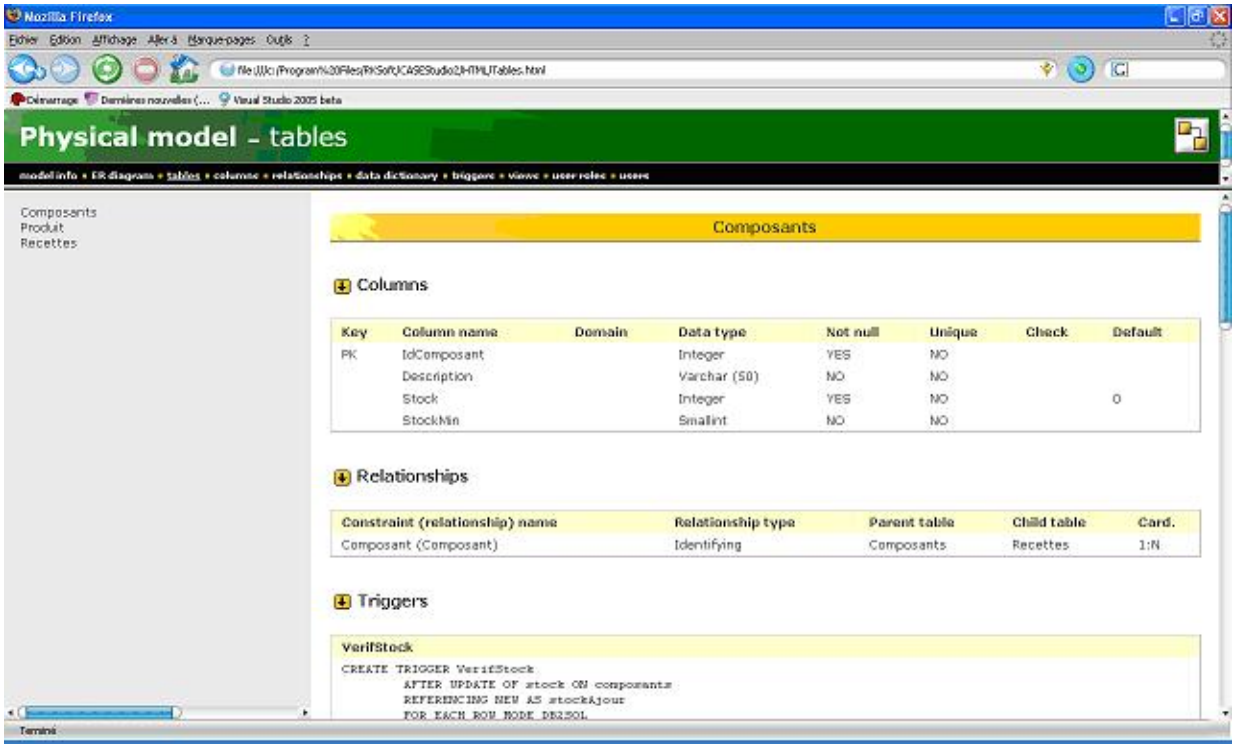
IX - Rapport

Vous pouvez éditer un rapport documentaire de votre modèle. Ce rapport peut être plus ou moins détaillé selon les options que vous choisissez. Le rapport peut être édité au format rtf ou au format HTML. Attention, les options diffèrent selon le type de rapport choisi. Pour ma part j'ai fortement apprécié le format HTML. Il existe 4 différents:

- Todo list report
- Data flow report
- Physical entity relationship report
- Logical entity relationship report
- User permission report



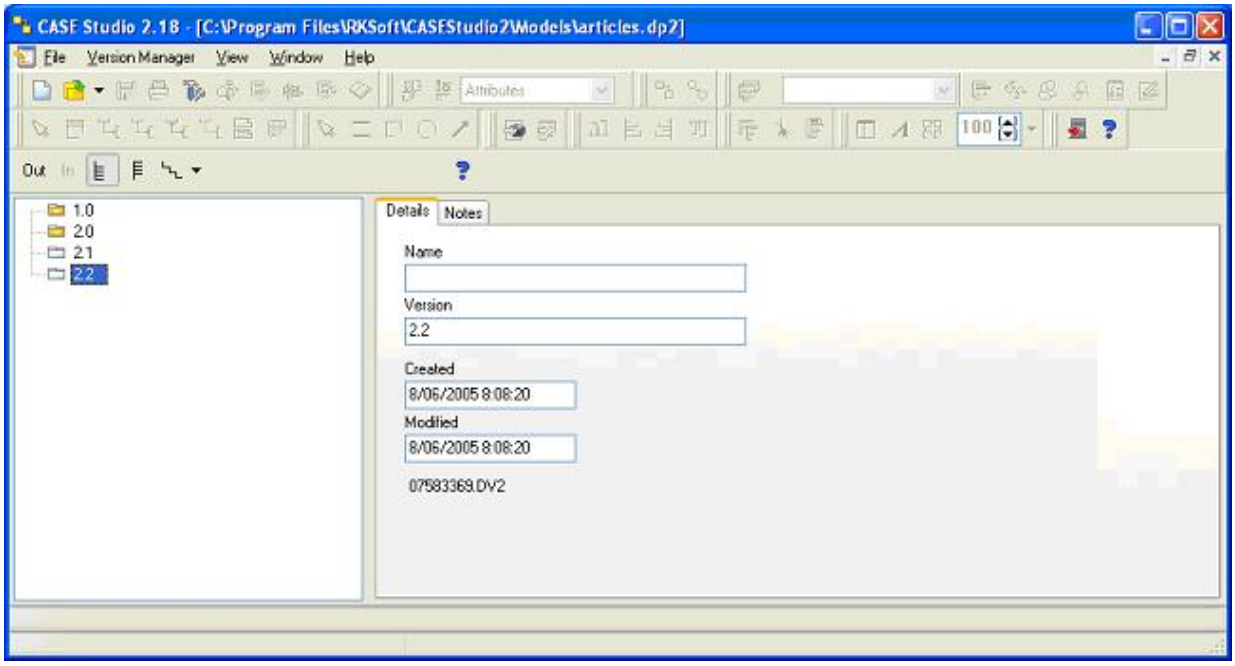




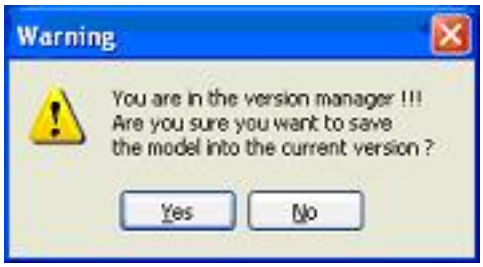
X - La gestion des versions

Pour ajouter un modèle dans le gestionnaire de version, il suffit de choisir l'option "Insert model into version manager"

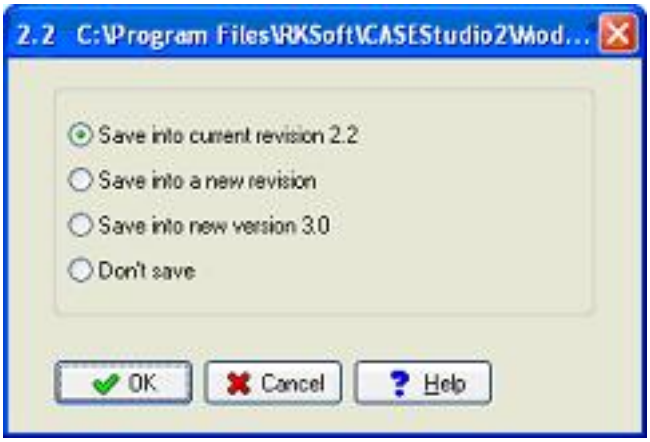
Vous disposez alors d'un fichier projet. Fermez votre modèle et à la place, ouvrez votre projet. La fenêtre gestion de projet s'ouvre automatiquement.



Vous choisissez la version que vous désirez obtenir et vous cliquez sur "Out". L'écran bascule sur la fenêtre habituelle du modèle. Mais si vous utilisez l'option "Save", vous recevrez le message suivant :

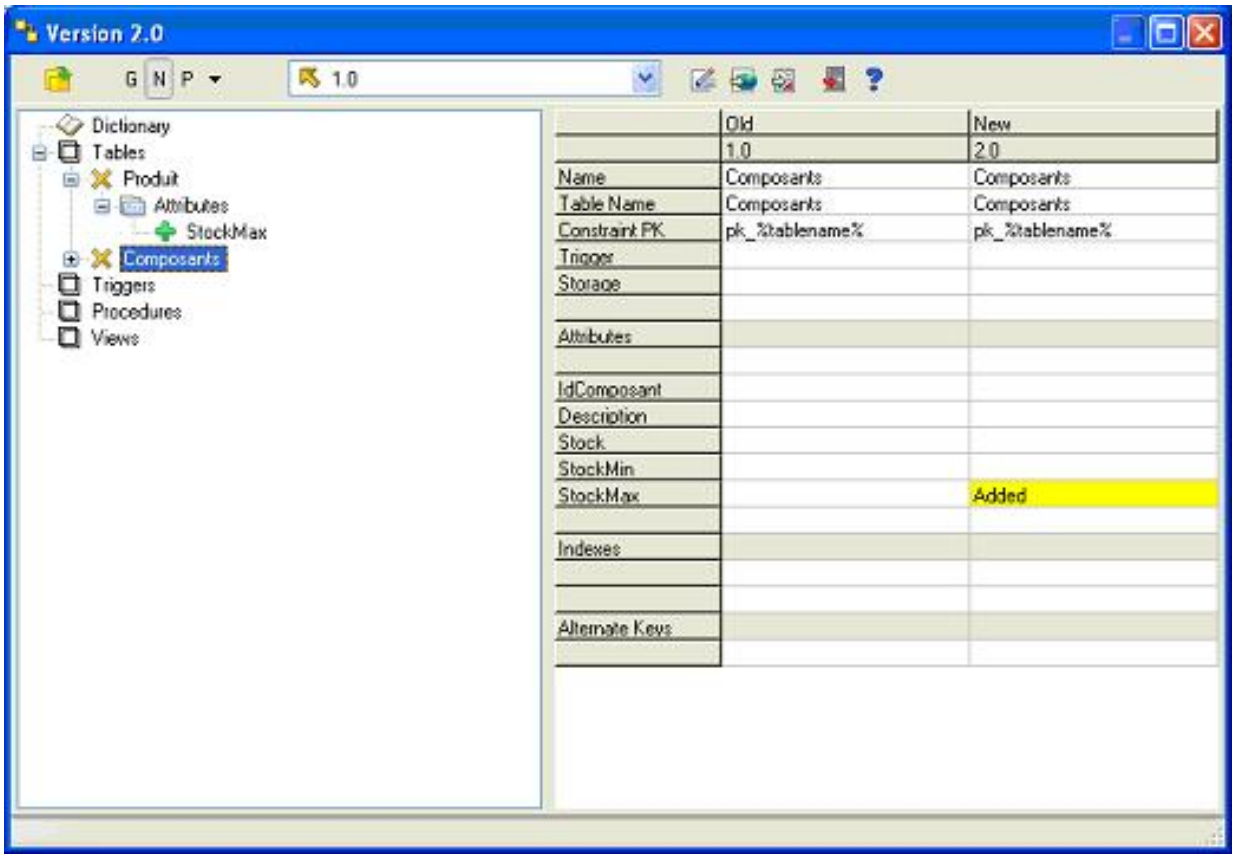


Si vous désirez sauver une nouvelle version, vous devez revenir dans l'écran de gestion des versions pour y faire un "IN". Pour revenir à cet écran, j'ai mis un certain temps à comprendre mais il suffit d'aller dans l'option " Windows ". La fenêtre du gestionnaire est en fait restée ouverte. Après avoir fait le "IN" vous aurez la possibilité d'enregistrer les modifications dans la version courante, de créer une nouvelle version ou de créer une nouvelle version majeure. Il semble que la division des versions soit limitée à 2 niveaux.



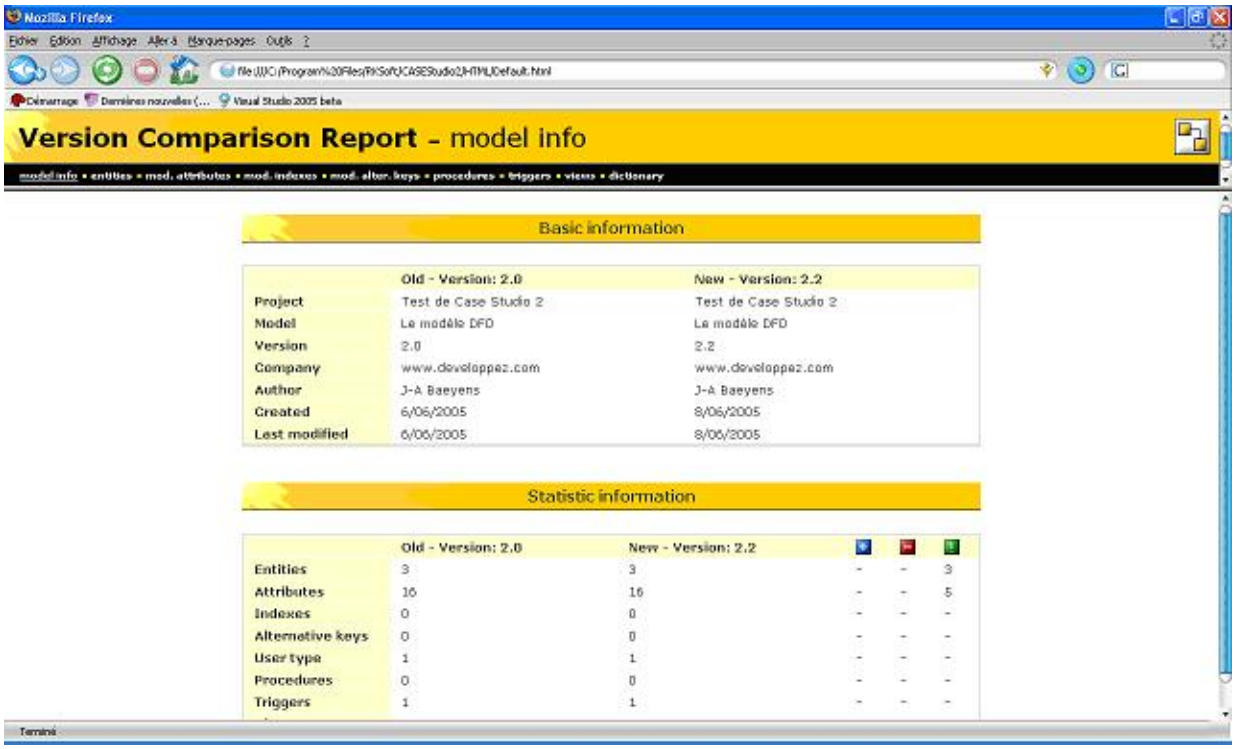
Il est tout à fait possible d'ouvrir simultanément plusieurs versions différentes du même modèle. Attention, après un transfert du modèle dans le gestionnaire de version, toutes les versions sont enregistrées dans le fichier d'extension dp2. Si vous aviez préalablement sauvé votre modèle dans un fichier d'extension dm2, celui-ci ne sera plus mis à jour. Il vaut donc mieux le supprimer pour éviter les confusions.

L'outil de comparaison de versions vous permet une visualisation aisée des différences.



Il est dommage que ce module ne permette pas de générer un script SQL permettant l'adaptation d'une version vers une autre. En effet, dans le cas où il s'agit d'une révision d'une base de données déjà en production, vous devrez vous-même écrire vos commandes SQL pour adapter la base de données.

Par contre vous pouvez éditer un rapport de comparaison et comme pour le modèle, ce rapport peut être au format HTML ou au format RTF. Le format RTF étant plus adapté pour l'imprimer et le mettre dans un dossier alors que comme précédemment, le format HTML est beaucoup plus agréable à consulter. Comme pour les rapports précédents, vous disposez d'options pour préciser ce que vous souhaitez voir apparaître dans votre rapport.



XI - Remerciements

Je remercie "MD Software" pour la relecture de cet article.

XII - Conclusion

En conclusion, il s'agit d'un excellent outil qui vous permettra très facilement de bien modéliser vos bases de données. Le reverse engineering vous permettra de récupérer vos anciens projets. Il est très intuitif et facile d'utilisation. Le plus gros reproche que je puisse lui faire concerne l'aide que j'ai trouvée insuffisante (trop peu exhaustive et une recherche d'informations pas toujours très aisée) mais franchement je n'en ai quasiment pas eu besoin. Il ne manque vraiment que la génération des scripts permettant la migration de la base de données d'une version à l'autre pour satisfaire pleinement le développeur que je suis. Le logiciel s'est montré stable pendant toute la durée des tests excepté un plantage sans conséquence en quittant l'application.

Je vous invite également à consulter notre page outils sgbd.developpez.com/outils