

Conception et déploiement d'un cube Analysis Services 2005

par [Mohamed Taslimanka Sylla](#)

Date de publication : 07/04/2008

Dernière mise à jour : 07/04/2008

Ce tutoriel a pour but de vous aider à prendre en main votre projet OLAP sous SQL Server Analysis Services 2005.

- I - Introduction
- II - Objectif
- III - Installation de la base DATAWAREHOUSE
- IV - DATAWAREHOUSE
- V - ANALYSIS SERVICES 2005
 - V-A - BIDS
 - V-A-1 - Créer la source de données
 - V-A-2 - Création de la vue de source de données
 - V-A-3 - Création du cube
 - V-A-4 - Réarrangement
 - V-A-5 - Déploiement
 - V-B - Management Studio
 - V-C - Excel
- VI - Conclusion

I - Introduction

Vous avez sans doute beaucoup entendu parler d'OLAP (OnLine Analytical Processing) et vous vous êtes peut être dit à quoi ça sert vraiment. Personnellement je suis tout à fait d'accord que cela peut sembler assez flou. Pourquoi par exemple a-t-on besoin de définir un schéma en étoile dans l'entrepôt relationnel et aussi dans Analysis Services. Ou encore pourquoi ne pas choisir de tout mettre dans une base de données relationnelle puis, choisir de faire des requêtes. Voici autant de questions qui m'amènent à réaliser ce tutoriel tout en me focalisant sur l'aspect théorique des choses.

Remarques:

Souvent, beaucoup de gens se plaignent de ne pas trouver des documents ou tutoriaux complets sur les systèmes multidimensionnelles. Sachez que l'une des principales cause à cela est la difficulté à trouver une grande base de données dans laquelle on peut fouiller. Car sachez bien que les systèmes décisionnels ont pour but de traiter des données afin d'en tirer de l'information. Donc vous vous imaginez que cela doit forcément être un gros volume de données. Donc il n'est pas très pertinent pour ceux qui suivent les tutoriaux d'observer des tendances (exemple: observer que les e-phone sont plus achetés par les jeunes de 18-25 ans habitant le plus souvent en île de france) banales. Ainsi la chose devient intéressante quand on traite de gros volume de données (pas forcément des téra-octets) vraisemblables.

II - Objectif

Tout d'abords nous allons supposer que nous disposons déjà d'un entrepôt de données de données qui a été confectionné à partir d'une base de production gérant des données sur les ventes. En réalité, cet entrepôt de données est un sous ensemble de données tirées de la base AdventureWorksDW de microsoft. Ce sous ensemble que j'ai extrait et appelé **DATAWAREHOUSE** constitue mon Entrepôt de données.

Vous pouvez télécharger mon entrepôt ici. Vous trouverez dans le zip l'entrepôt sous 2 formats:

- 1 script SQL (il suffit juste de l'exécuter dans l'analyseur)
- 2 fichiers détachés (fichiers MDF et LDF à attacher en modifiant les chemins)

L'objectif est de travailler avec cet ensemble de données plus restreint que la base initiale tout en ayant un jeu de données bien garni et permettant une analyse vraisemblable.

Sur ces données, nous allons construire un cube (avec ses dimensions) puis le déployer sur une instance SSAS pour ensuite pouvoir l'interroger à partir de Management Studio.

III - Installation de la base DATAWAREHOUSE

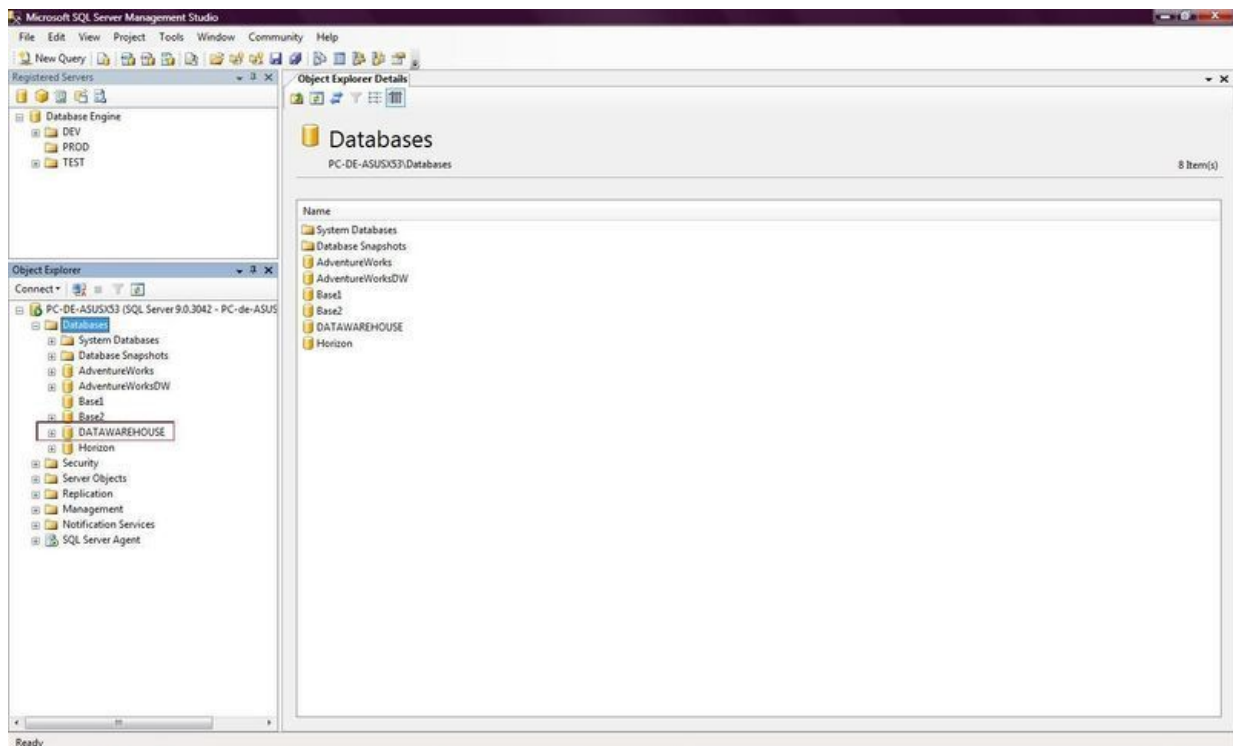
Si vous choisissez l'option script SQL, il suffit d'ouvrir le fichier **DATAWAREHOUSE.sql** et de faire F5.

Si vous choisissez d'attacher les fichiers MDF et LDF faites comme suit:

```
USE master
EXEC sp_attach_db @dbname = N'DATAWAREHOUSE',
    @filename1 = N'c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DATAWAREHOUSE_Data.mdf',
    @filename2 = N'c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DATAWAREHOUSE_log.ldf' ;
```

Bien sûr il faut savoir que **c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data**, c'est mon emplacement à moi et que chacun peut avoir le sien.

Après avoir exécuté le script ou attaché les fichiers, vous devez avoir quelque ressemblant à l'image ci-dessous.

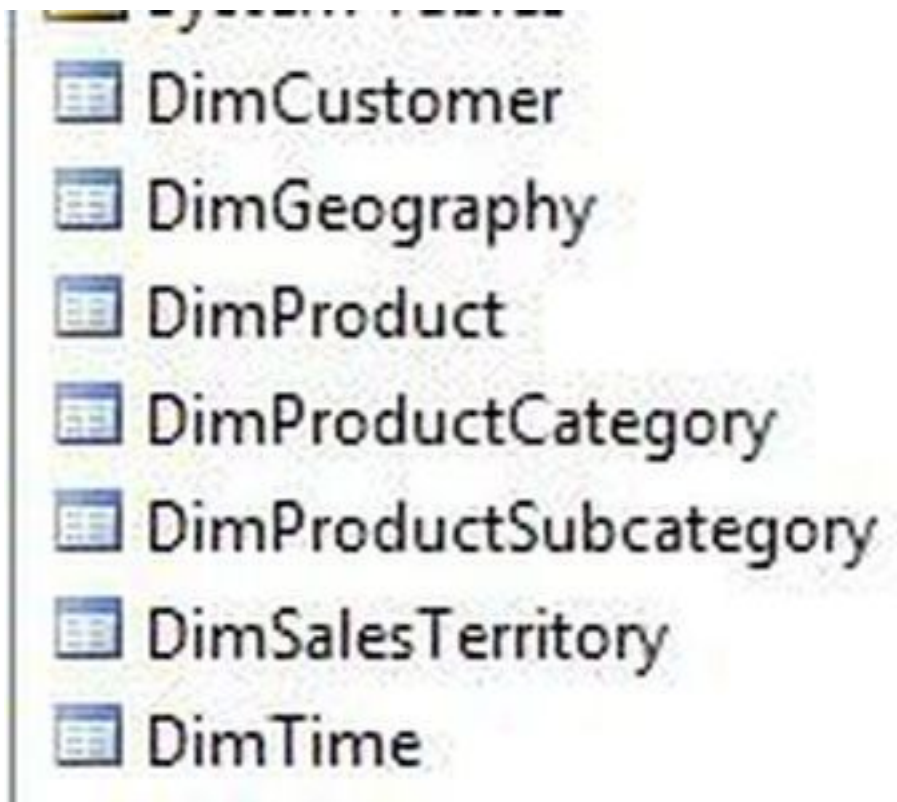


Base DATAWAREHOUSE

Dans la suite, vous allez voir ce que contient cette base de données avant de vous lancer à la conception de cubes OLAP. Car une mauvaise compréhension de l'entrepôt sous-jacente induit à une mauvaise définition de cubes.

IV - DATAWAREHOUSE

Comme je vous l'ai dit au tout départ, cette base est un sous-ensemble de la base AdventureWorksDW. Ce sous ensemble est constitué de 7 tables.

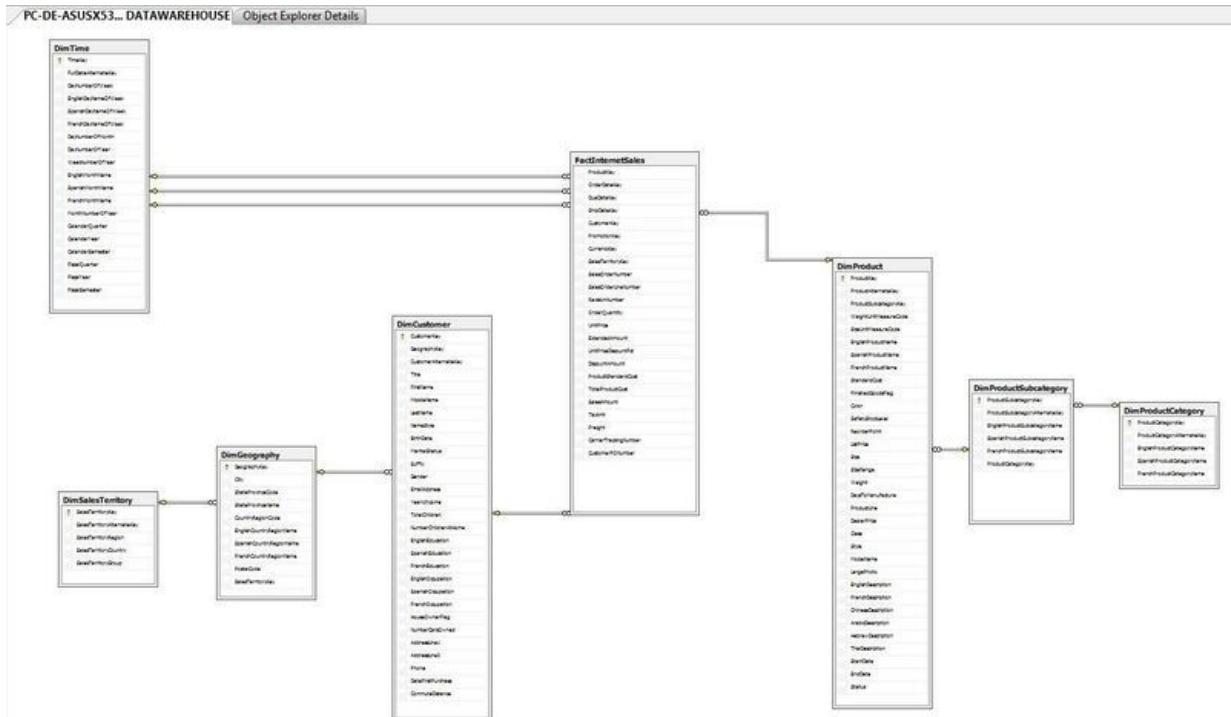


tables de la base DATAWAREHOUSE

- 1 La table DimCustomer contient des clients
- 2 La table DimGeography contient la localités des clients
- 3 La table DimProduct contient la liste des produits que la société AdventureWorks vend.
- 4 La table DimProductSubCategory contient les sous-catégories de produits
- 5 La table DimProductSubCategory contient les catégories de produits
- 6 La table DimSalesTerritory contient les lieux de ventes
- 7 La table DimTime contient l'ensemble des jours de l'année 2001 à l'année 2004

Remarquez que toutes les tables de l'entrepôt de données commencent soit par "Dim" (pour les tables de dimension) soit par "Fact" (pour les tables de faits). C'est une bonne stratégie pour adopter une bonne convention de nommage même si l'on trouve quelque fois des tables de faits qui peuvent aussi être dimension d'une autre table de fait. Bref, quoi qu'il en soit, adoptez vos propre "design pattern". L'essentiel étant que vous vous y retrouviez bien.

Ci-dessous, voici le lien entre ces tables de l'entrepôt de données **DATAWAREHOUSE**.



Liens entre les tables de DATAWAREHOUSE

Dans ce schéma, on voit bien qu'on a la table de fait centrale appelée **FactInternetSales** contenant toutes les mesures qu'on veut analyser. Et autour on voit toutes les tables de dimension. Ce schéma est appelé schéma en flocon parce que les tables de dimension ne sont pas toutes reliées directement à la table de fait.

J'attire l'attention sur l'existence des trois liens entre DimTime et FactInternetSales. Le premier lien est un lien "**date d'expédition**", le deuxième est un lien "**date de livraison**" et le troisième est un lien "**date de commande**". On pourrait même prévoir un nouveau champs calculé qui serait appelé "**délai de livraison**" et qui contiendrait (**date_de_livraison - date_de_commande**). Mais bien évidemment il est plus intéressant de gérer ça au niveau d'Analysis Services.

Maintenant que notre entrepôt de données est en place, nous allons passer sous Analysis Services pour nous occuper de notre structure dimensionnelle.

V - ANALYSIS SERVICES 2005

Enfin nous y voilà! Nous arrivons au coeur du sujet: **ANALYSIS SERVICES**. Jusqu'à présent nous avons construit une base de données relationnelle appelé **DATAWAREHOUSE** qui a un schéma non normalisé (la 3FN n'est pas respectée). Maintenant nous allons encore modéliser mais cette fois-ci sous Analysis Services. Et figurez-vous que nous allons encore créer une structure multidimensionnelle. Pourquoi?

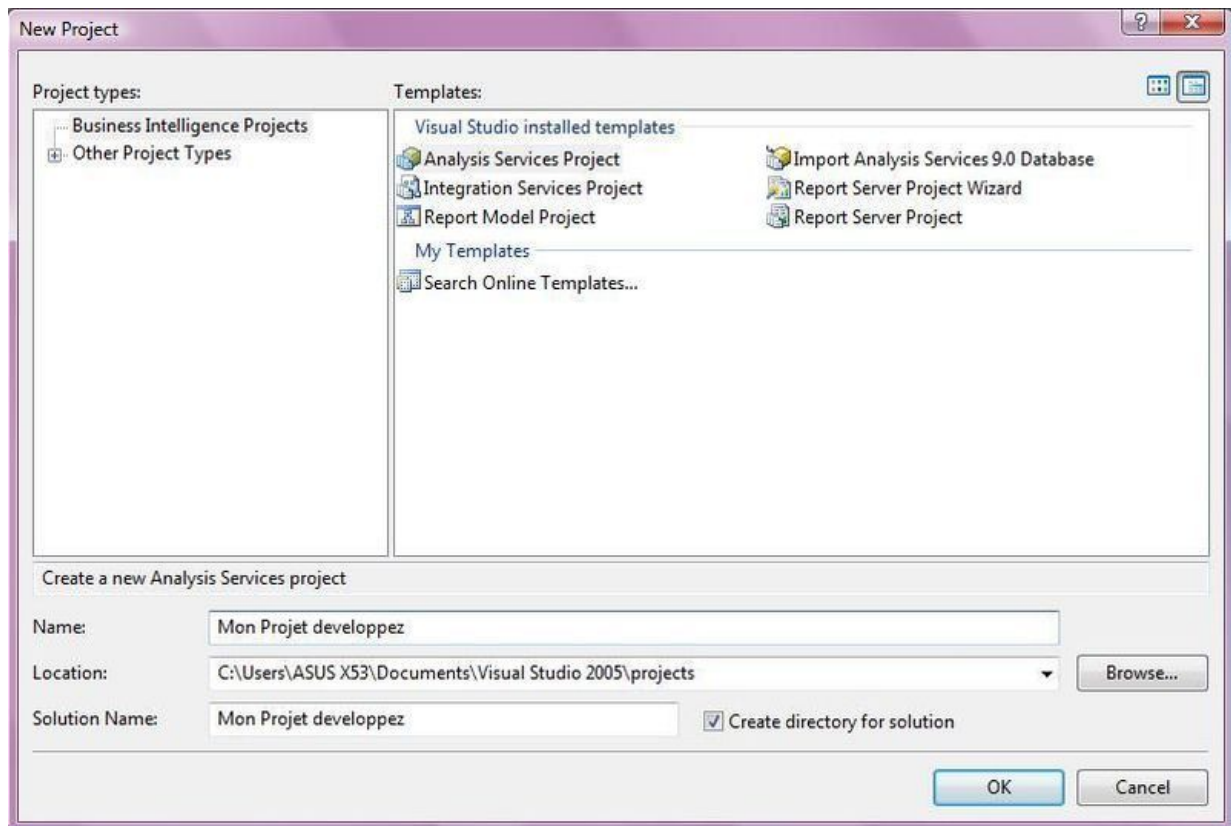
L'explication est la suivante: Notre base de données DATAWAREHOUSE a une structure multidimensionnelle mais reste cependant une base de données relationnelle. Car elle est gérée par le moteur de base de données relationnel de Sql Server (Relational DataBase Engine). Par contre nous créerons aussi une structure mutidimensionnelle (qui sera notre cube Analysis Services) sous le serveur d'analyse de Sql Server et là c'est le moteur OLAP qui intervient. Ce moteur se chargera donc de faire les pré-aggrégations et aggrégations nécessaires. Cela nous amènera donc à faire 2 travaux:

- Un premier travail d'optimisation de la base DATAWAREHOUSE (en choisissant la bonne modélisation et les bon indexes)
- Un deuxième travail d'optimisation du cube de données (en choisissant la bonne modélisation, les bonnes représentation des hiérarchies utilisateurs et les bonnes relations d'attributs etc.)

Maintenant nous allons donc entrer sous Analysis Services et construire notre cube de données ou base de données multidimensionnelle.

V-A - BIDS

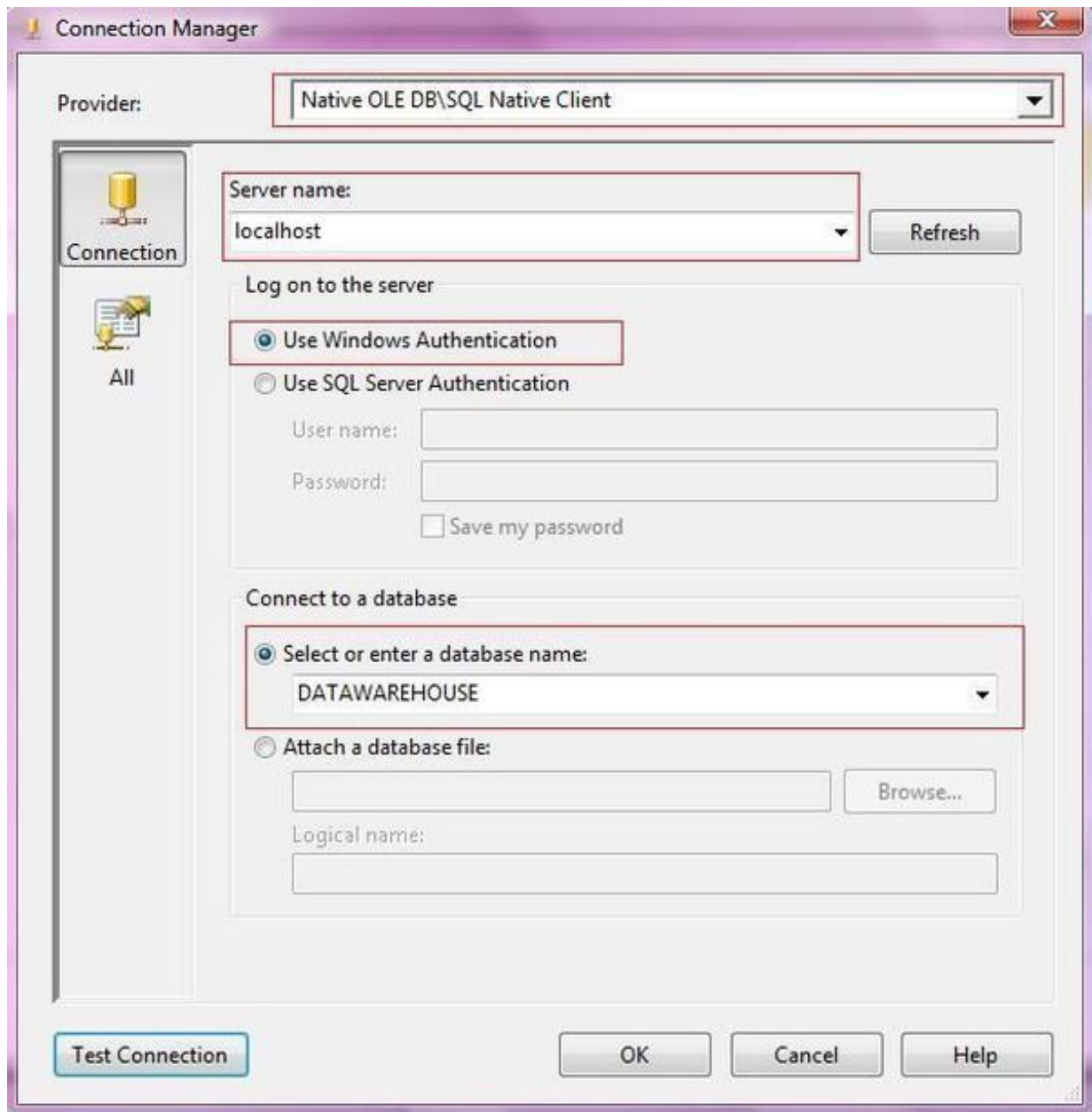
BIDS qui est l'abréviation de **Business Intelligence Development Studio** est la plateforme essentielle pour faire le développement de nos cubes de données. Nous allons donc l'utiliser. Pour ce faire, démarrez le créez un nouveau projet **Analysis Services**. Appelez le **Mon Projet developpez**.



Créer un nouveau projet Analysis Services

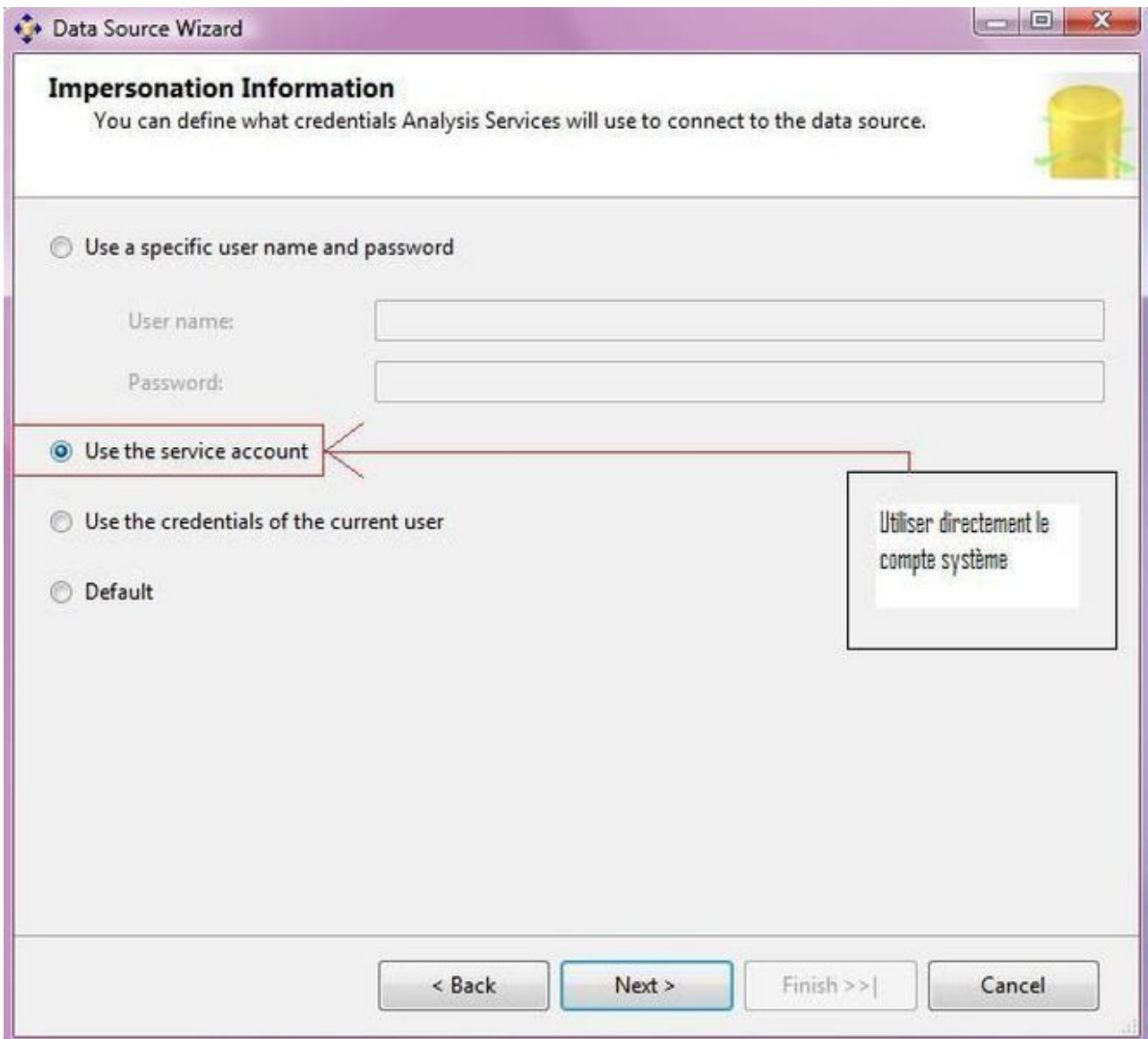
V-A-1 - Créer la source de données

La première chose à faire c'est de créer une **source de données** sur notre entrepôt de données **DATAWAREHOUSE**. Pour cela, dans la boîte de dialogue Connection Manager, choisir le fournisseur de données **Native OLE DB\SQL Native Client**, le nom du serveur contenant votre base de données DATAWAREHOUSE (dans mon cas, je l'ai mis sur mon ordinateur en **local**), le type d'authentification (moi j'ai choisi le mode **authentification windows**) et enfin, choisissez la base de données **DATAWAREHOUSE**.



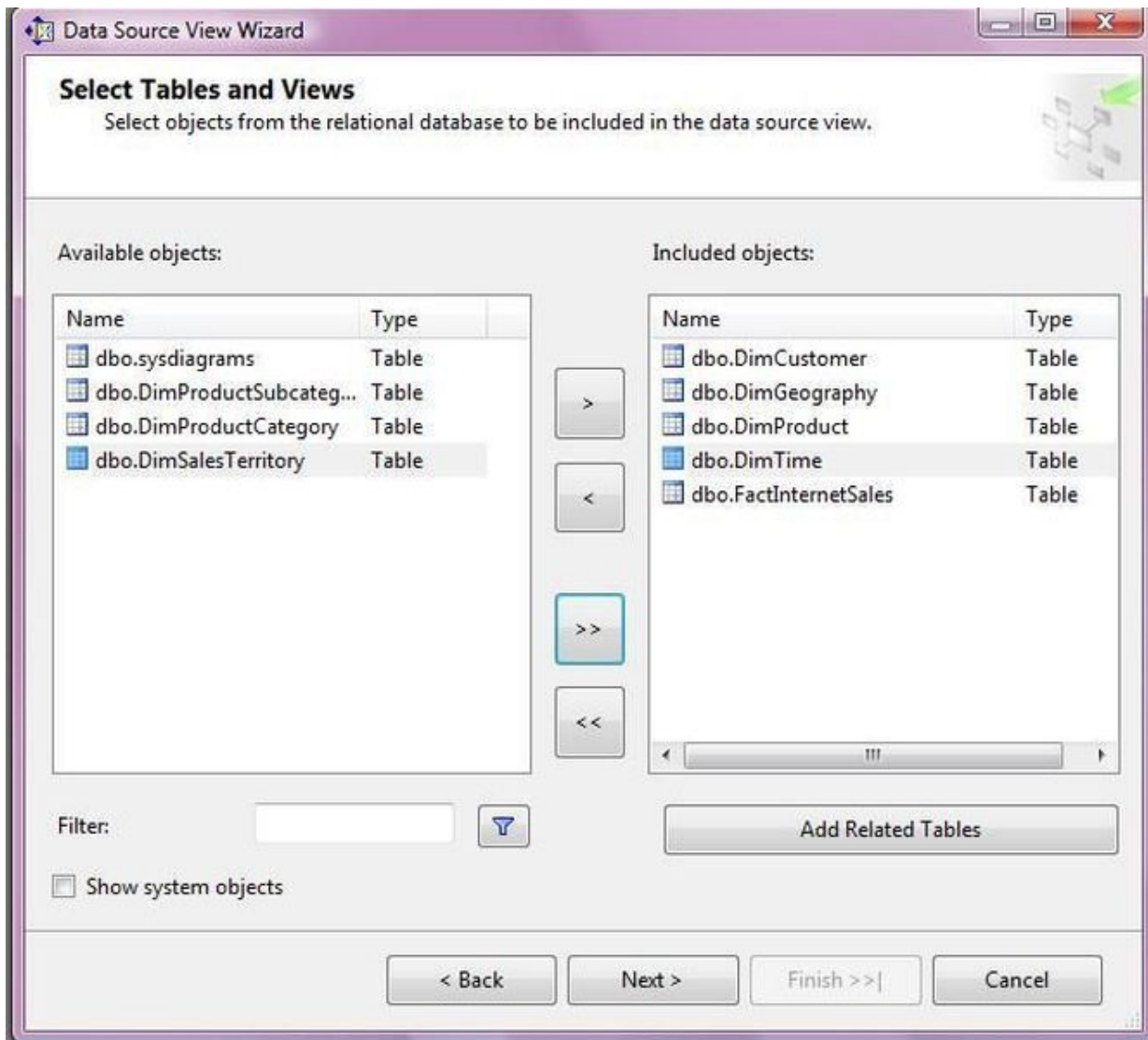
source de données

Il faut aussi choisir le login utilisé par Analysis Services pour opérer sur la base DATAWAREHOUSE (il peut s'agir du compte système local ou d'un compte windows particulier comme le vôtre).



V-A-2 - Création de la vue de source de données

La deuxième chose à faire est de créer une **vue de source de données** sur notre source de données. La vue de source de données sert à sélectionner des tables de la source de données. Ainsi pour une même source de données, on peut avoir plusieurs vues de sources de données (une par type de besoin par exemple). Nous allons donc créer une source de données qui contiendra les tables qui seront nécessaires à notre analyse (en l'occurrence les tables: Clients, Produits, Géographie, Temps ainsi que la table Vente_sur_Internet).



Choix des tables

Une fois avoir ajouté les tables dont nous avons besoin, on obtient le schéma suivant. Les liens ont été créés par Analysis Services. Ce dernier se base sur les clés étrangères définies au niveau de la base de données **DATAWAREHOUSE** pour déterminer les liens dans Analysis Services.

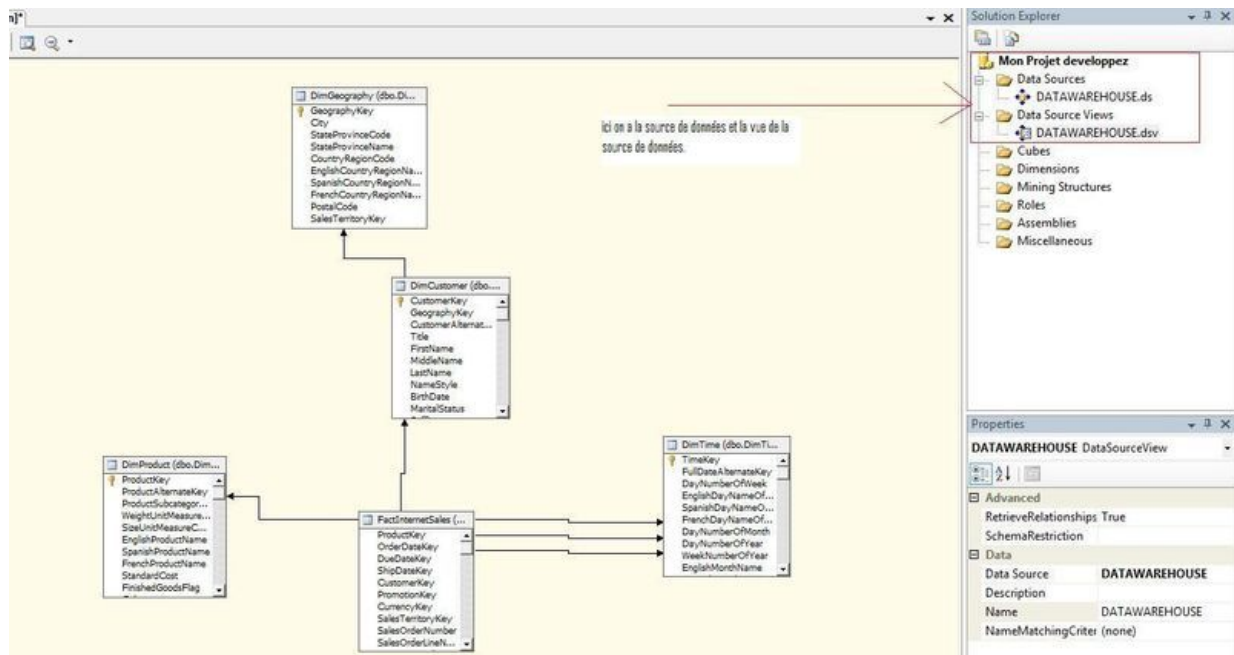
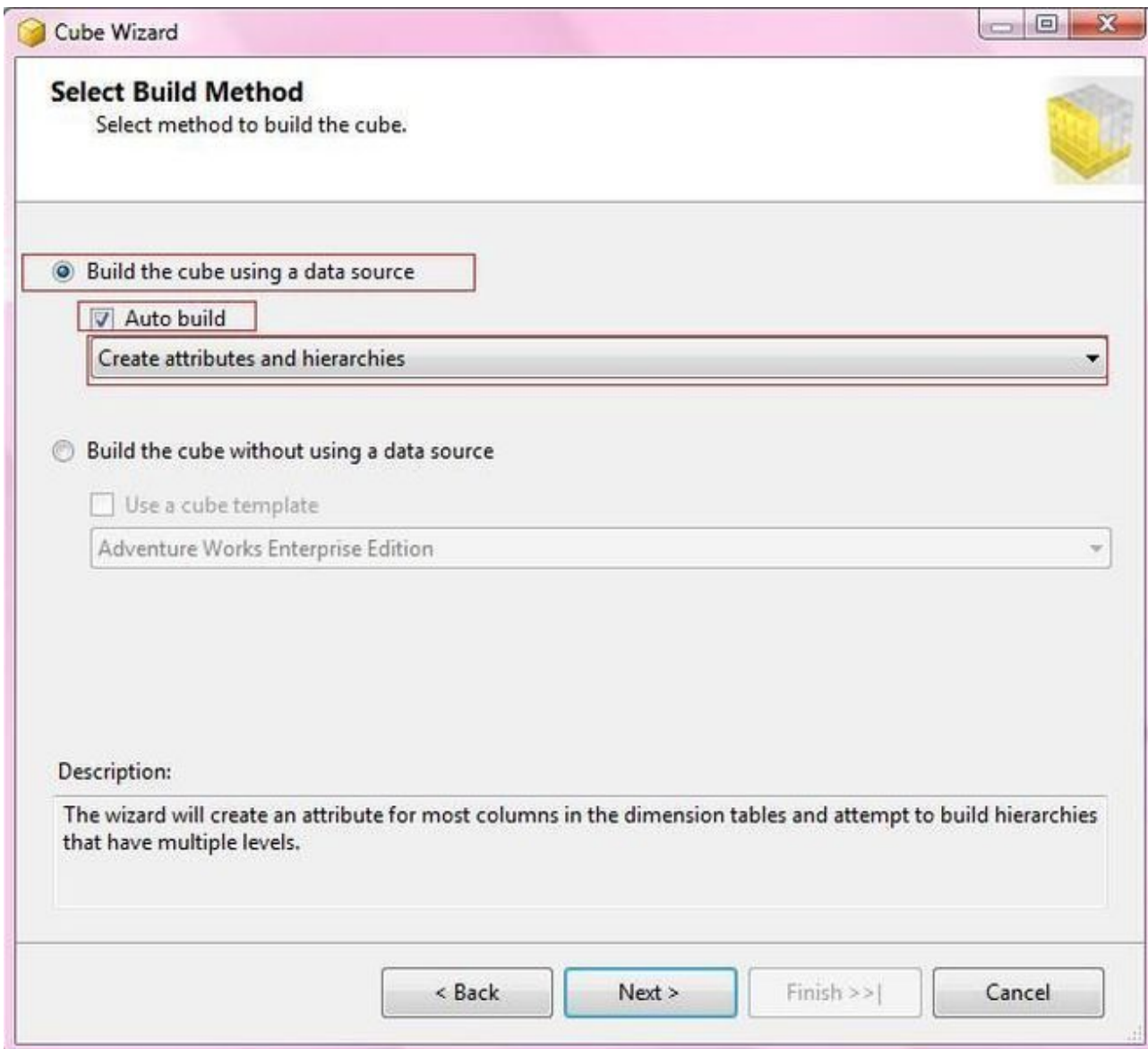


schéma de source de données

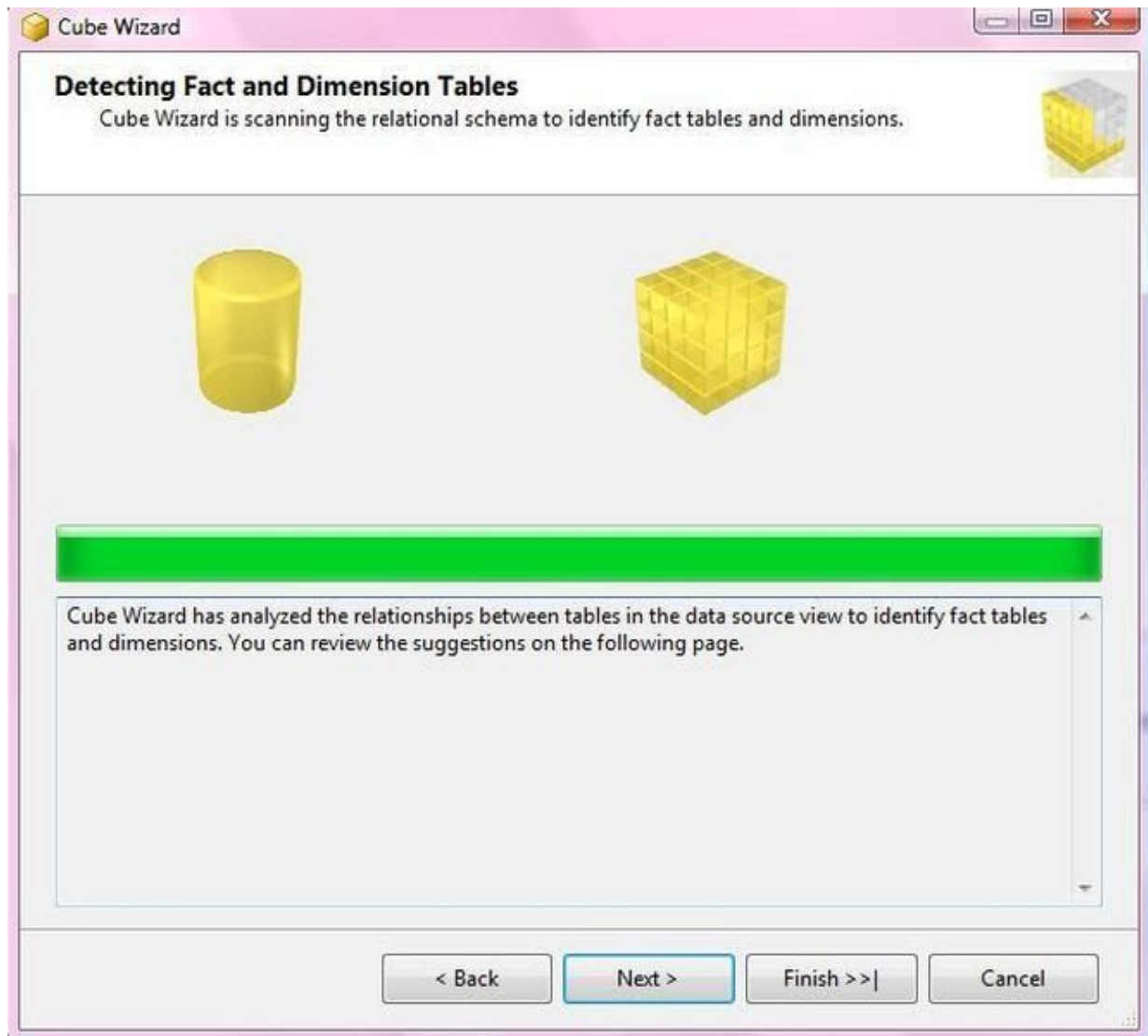
V-A-3 - Création du cube

Maintenant que nous avons la vue sur les données que nous allons utiliser, nous allons créer notre cube de données. Ici nous allons créer notre cube en mode assistant bien que ce ne soit pas préférable. En temps normal prenez l'habitude de créer vos dimensions proprement puis, traitez les, créez vos mesures et traitez votre cube.

Comme le montre la figure suivante, le cube sera construite en utilisant notre source de données définie. De plus, Analysis Services construira automatiquement ses attributs et ses hiérarchies en s'appuyant sur les métadonnées de la base **DATAWAREHOUSE**. Bien évidemment, pour avoir un cube optimal il ne faut pas laisser Analysis Services s'en occuper.

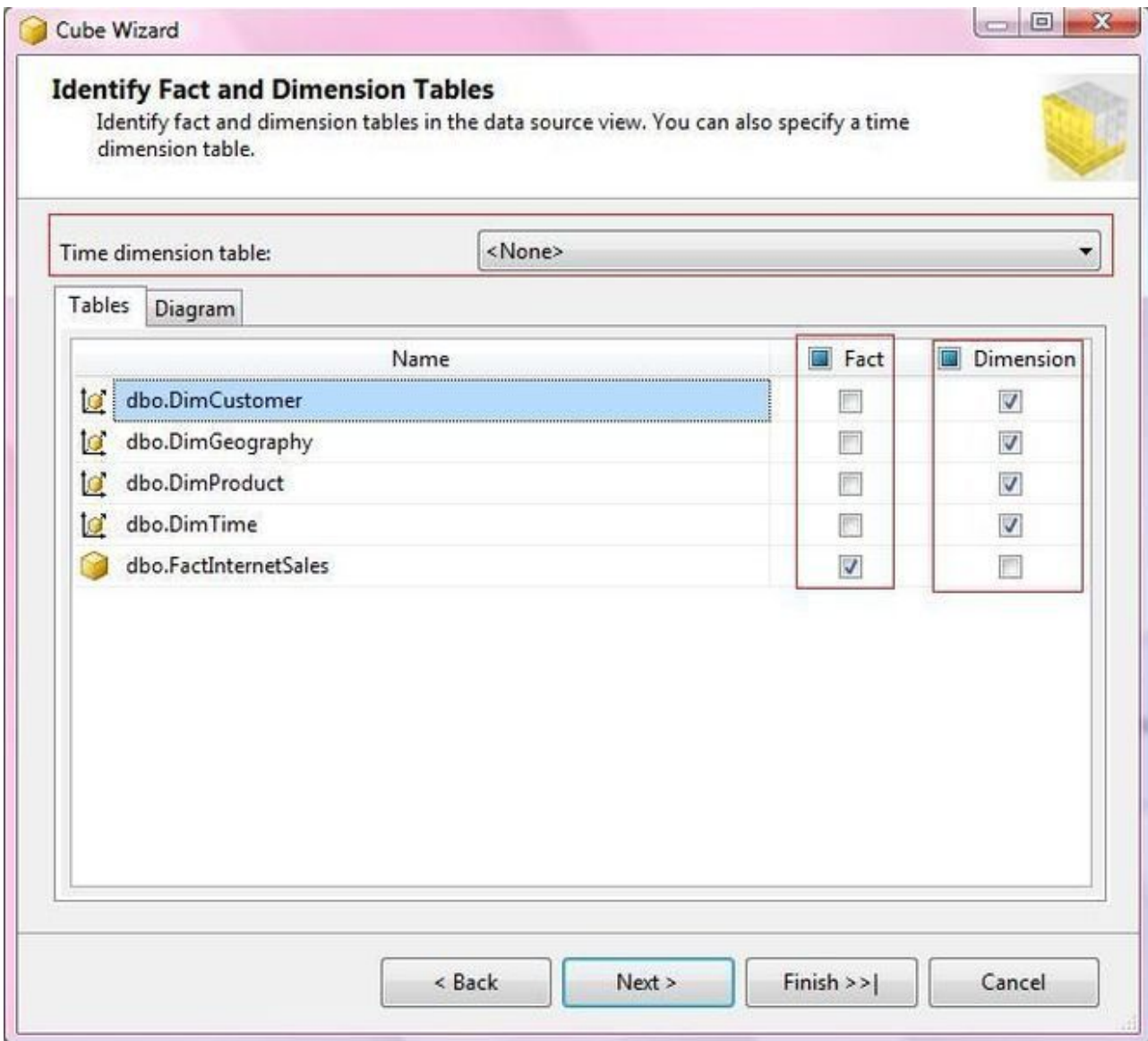


Cela déclenchera donc le moteur d'Analysis Services qui va faire ses calcul comme le montre la figure suivante:



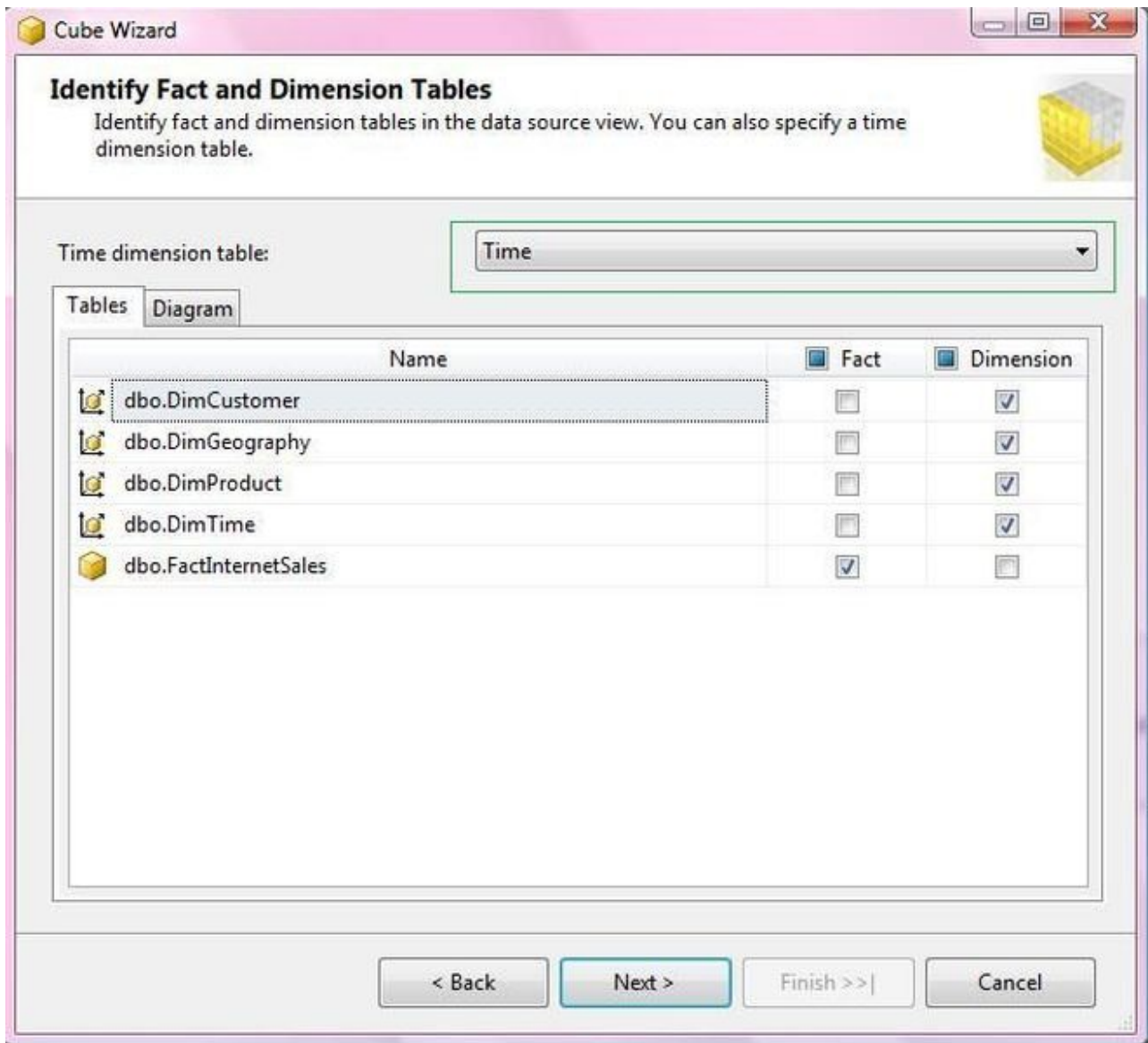
Calcul du cube

Après calcul de Analysis Services, voilà ce qu'il propose:

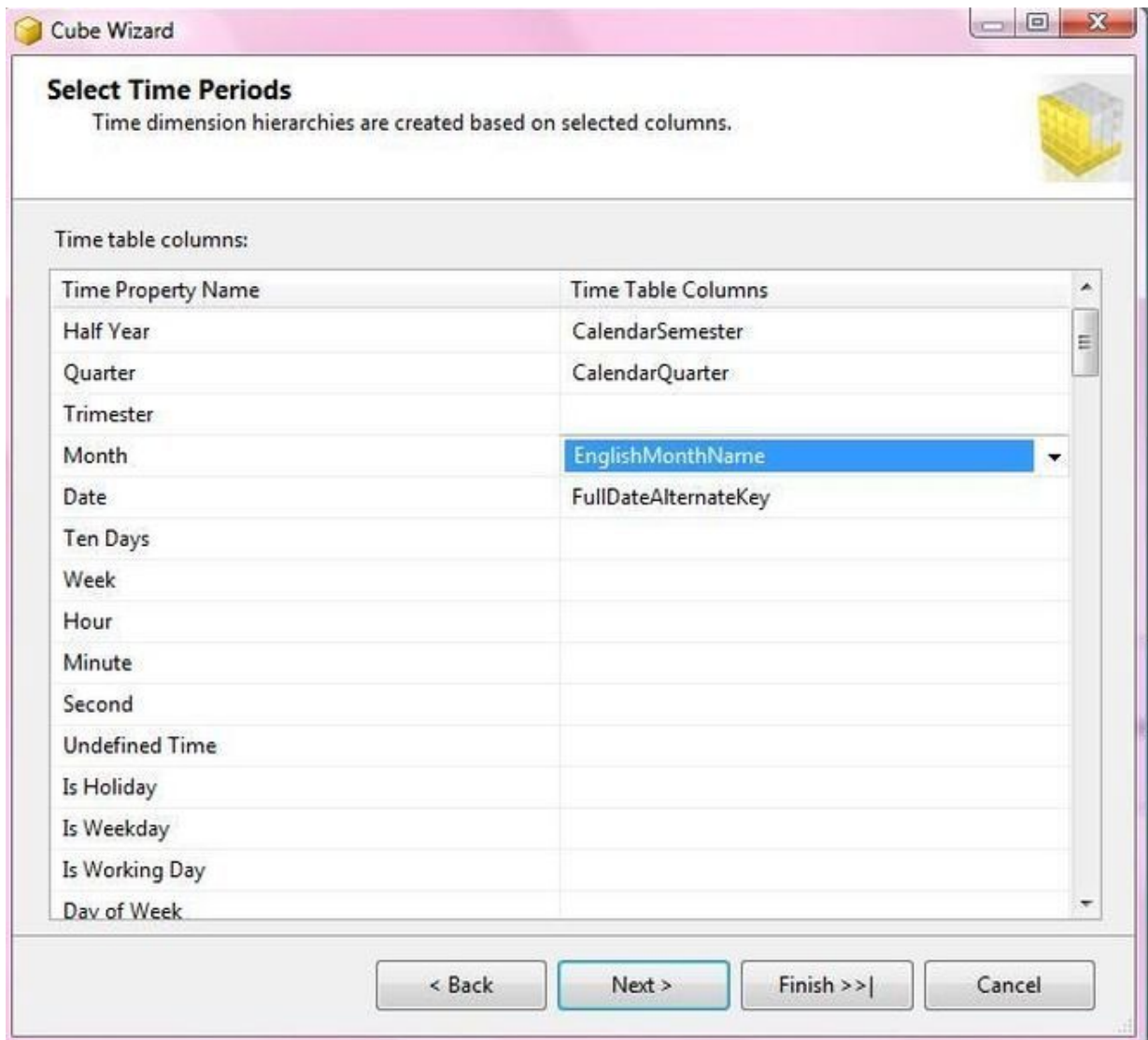


Il propose donc **FactInternetSales** comme table de fait et propose **DimCustomer**, **DimGeography**, **DimTime** et **DimProduct** comme tables de dimensions. Bien sûr pour proposer cela, il s'appuie sur ce qui a été définie sur la base **DATAWAREHOUSE** notamment sur les clés.

Cela est correct pour nous sauf que nous allons lui dire de traiter la dimension **DimTime** comme une dimension temps d'Analysis Services. Pourquoi cela? Eh bien, juste parce que la dimension Temps d'Analysis Services est bien garni et permet de réaliser des requêtes **MDX** avancées sur les périodes. Par exemple les requêtes sur les périodes parallèles. Il faut donc comme sur la figure suivante choisir **Time**. Il s'agit de la dimension **Time** d'Analysis Services.



Maintenant, il faut mapper cette dimension **Time** d'Analysis Services avec notre dimension **DimTime**. Faites cela comme le montre la figure suivante:



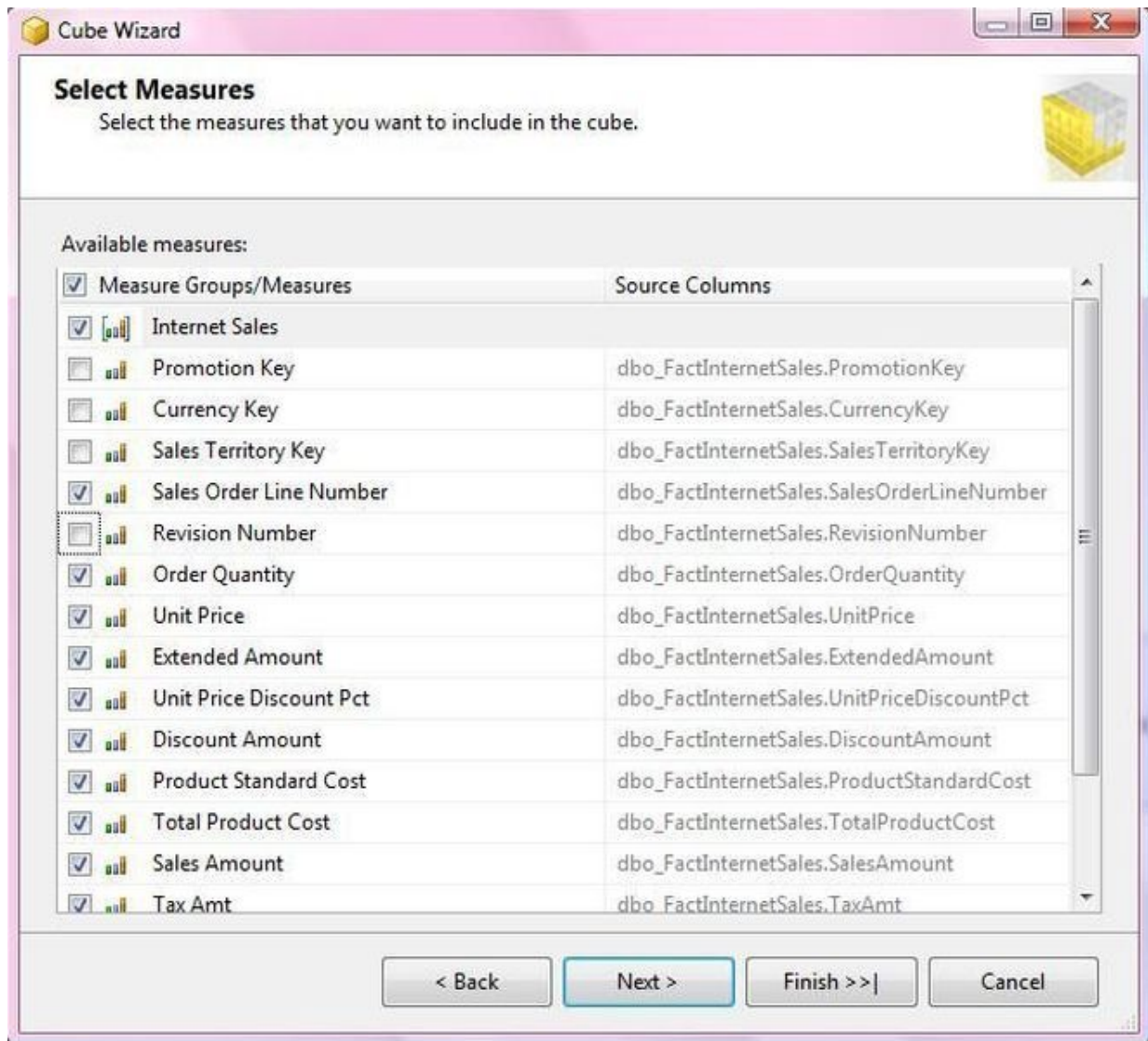
Mapping entre Time et DimTime

On aura pour cela les mappings suivants:

- Half Year (semestre) d'Analysis Services correspondra à CalendarSemester de DATAWAREHOUSE
- Quarter (trimestre) d'Analysis Services correspondra à CalendarSemester de DATAWAREHOUSE
- Month(mois) d'Analysis Services correspondra à EnglishMonthName de DATAWAREHOUSE
- Date(jour) d'Analysis Services correspondra à FullDateAlternateKey de DATAWAREHOUSE

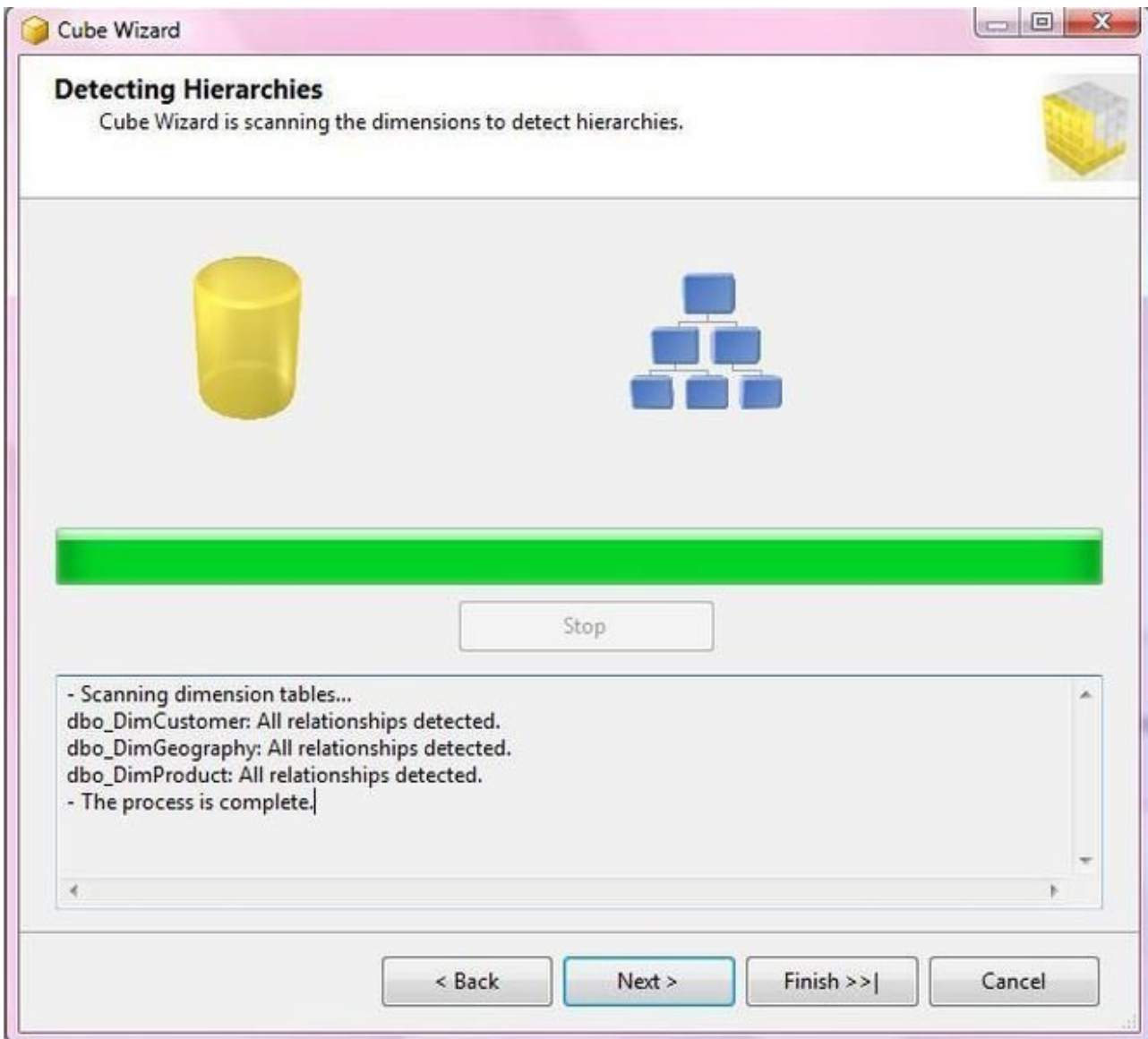
Remarquez que la granularité la plus fine correspond à la date du jour. Et ce même jour (**FullDateAlternateKey**) est la clé de DimTime.

Maintenant nous arrivons au stade où il faut définir les mesures. Nous allons choisir tous les attributs de la table de faits sauf les clés étrangères et **RevisionNumber** comme le montre la figure suivante:

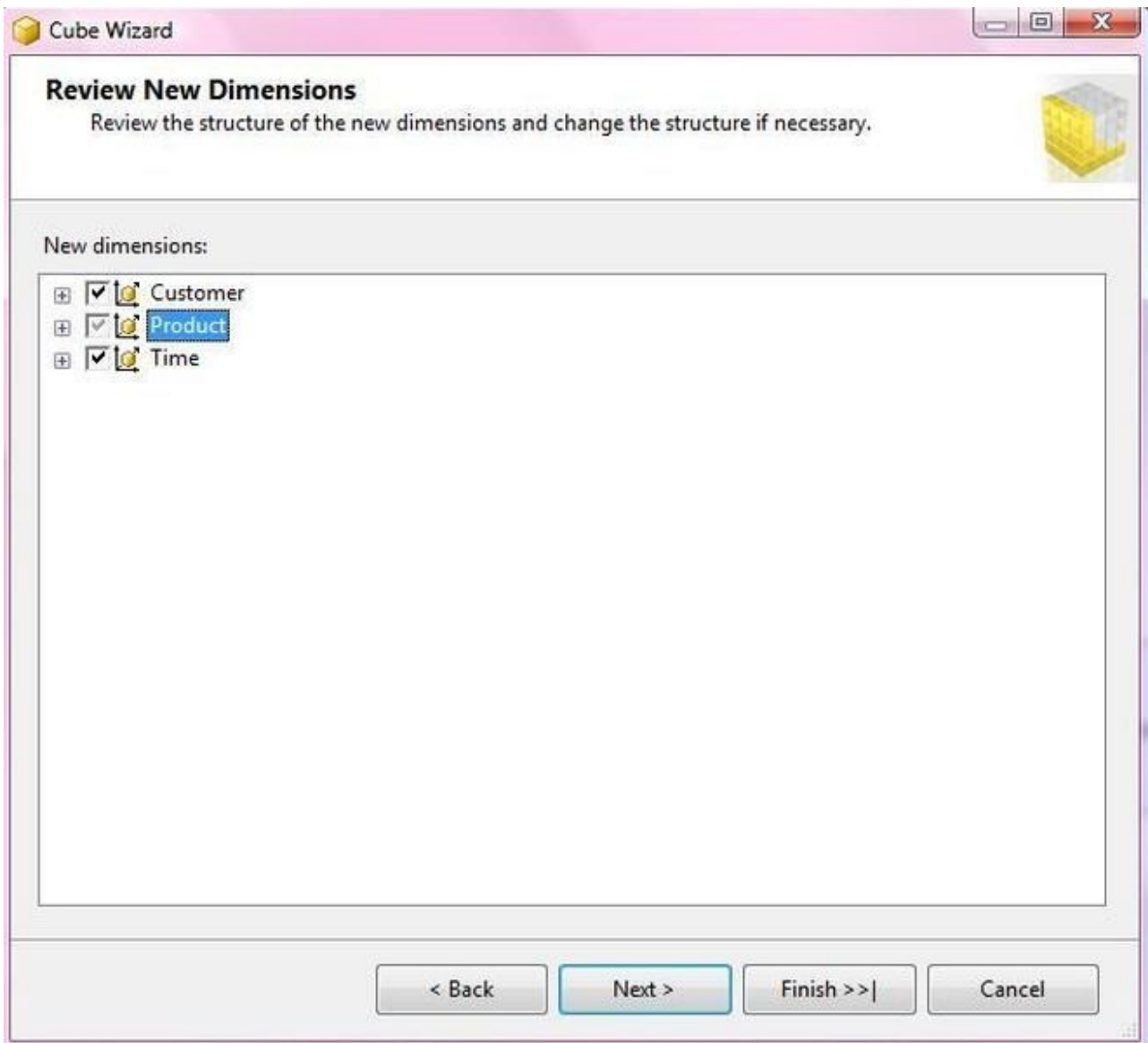


définition des mesures

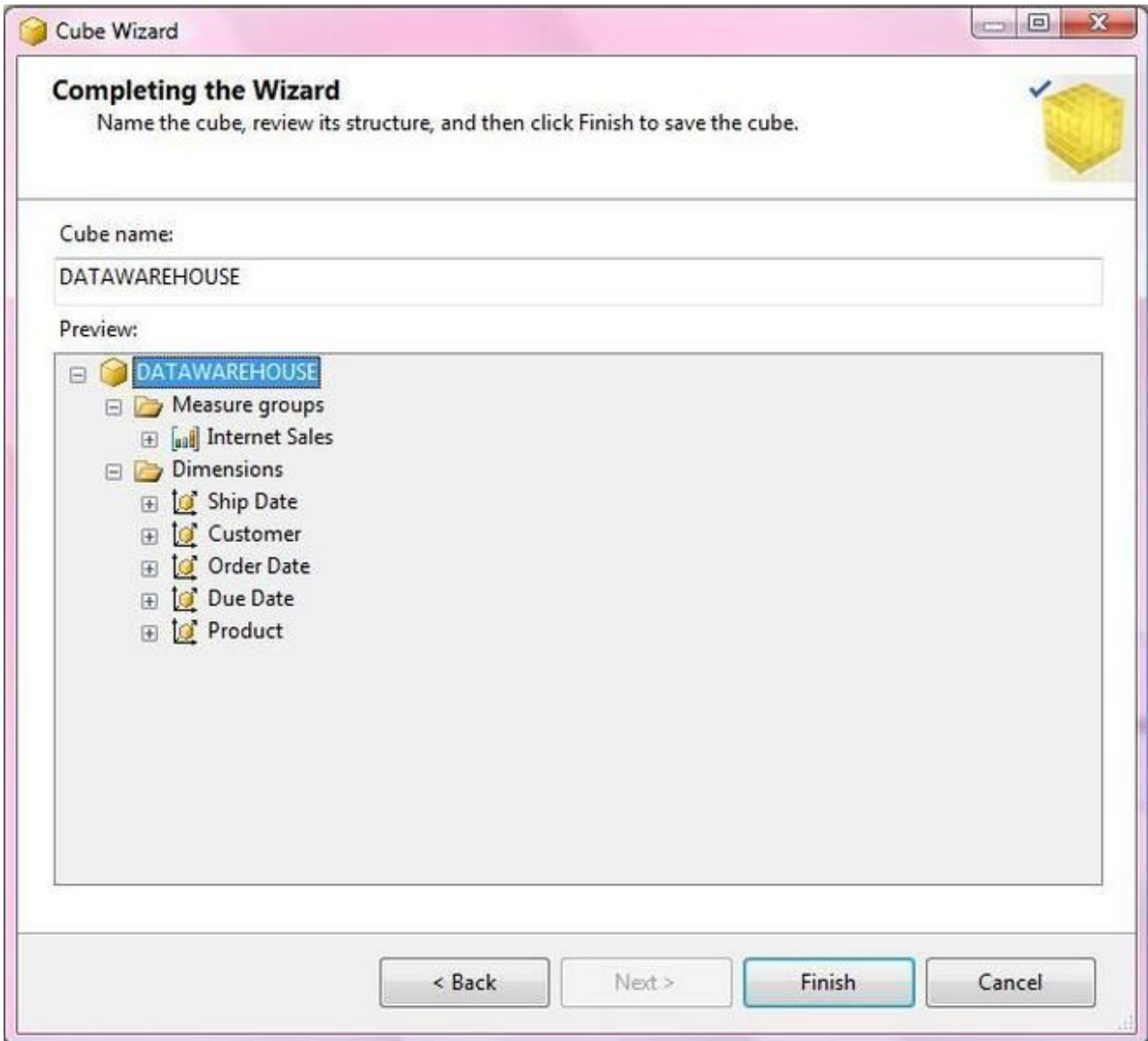
L'opération de scan va maintenant démarrer comme le montre la figure suivante:



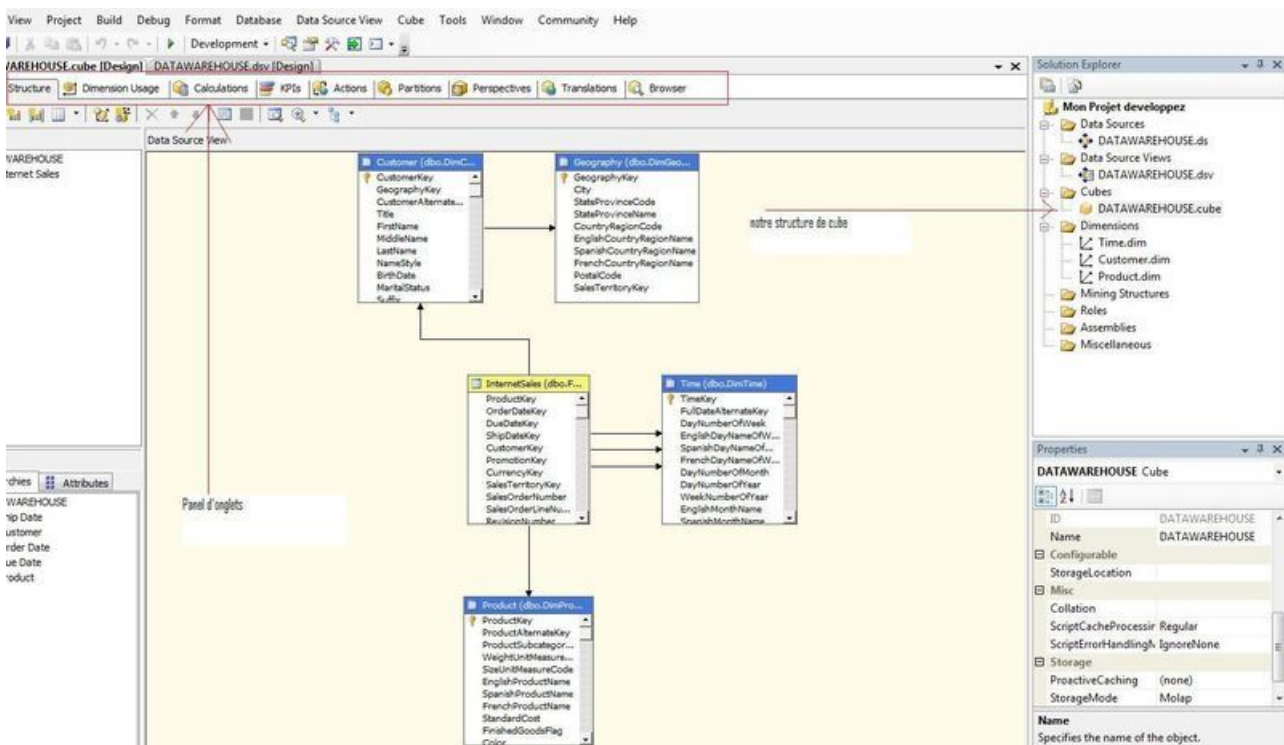
Cette opération de scan va donc détecter les relations d'attributs et de hiérarchie dans le cube. Après l'assistant nous proposera les attributs à prendre dans chaque dimension. Par défaut, tous les attributs sont proposés. De même observez que la dimension Geography n'existe plus car elle est considérée comme une hiérarchie de la dimension Customer.



Et voilà vous arrivez à la fin de la création de votre cube auquel vous lui donner un nom comme le montre la figure suivante:



Voici à quoi ressemble maintenant votre environnement BIDS:

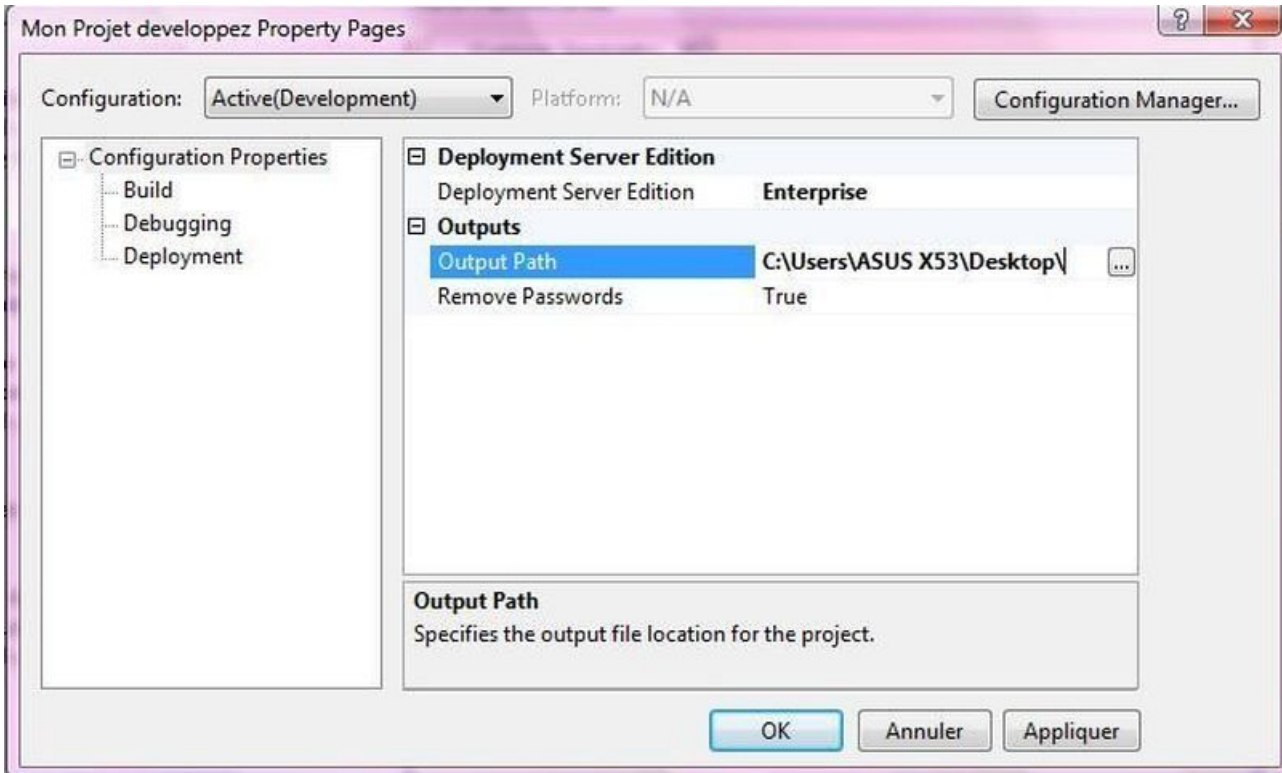


V-A-4 - Réarrangement

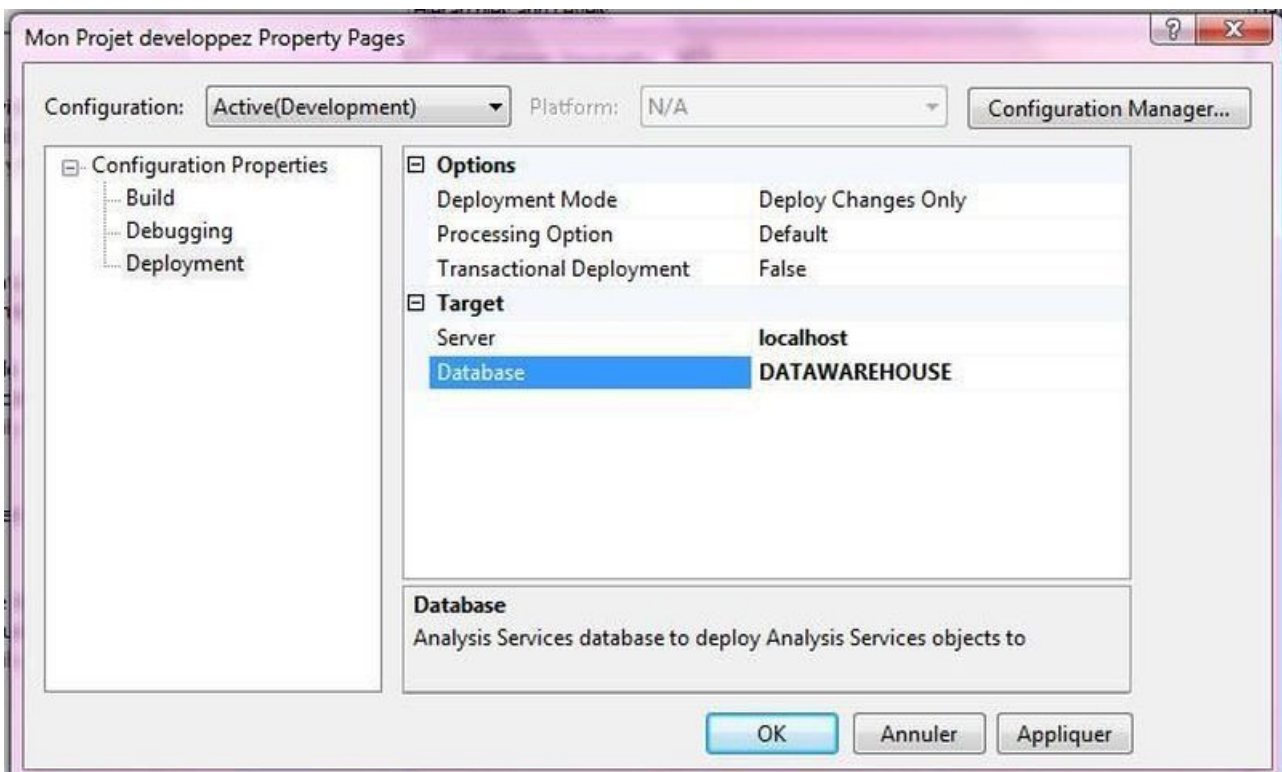
Après la construction du cube, il est recommandé de repasser dessus afin de le retravailler. Rien que cette partie mériterait d'être traitée en plusieurs tutoriaux à part. Elle requiert une attention très particulière. Ici nous n'allons pas en parler.

V-A-5 - Déploiement

Maintenant que tout est prêt, il ne nous reste plus qu'à déployer notre cube sur un serveur d'analyse. Cela permettra de rendre notre cube disponible à tout le monde. Pour cela, faites bouton droit sur le projet et **propriété**. Vous pouvez alors définir les propriétés de déploiement. Pour cela définissez d'abord les propriétés du **build** en choisissant **build**. Vous pouvez donc définir l'édition SQL Server sur lequel déployer votre cube (ici Enterprise Edition) ainsi que le lieu où stocker le script de déploiement. Le script de déploiement est particulièrement utile pour recréer le cube, exactement comme on fait avec les script **.sql**.



Ensuite cliquez sur **déploiement**. Cela permettra de déployer le cube sur le serveur d'analyse. Choisissez le nom du serveur sur lequel le déployer (ici localhost) et le nom de que voudrez lui donner (ici **DATAWAREHOUSE**).



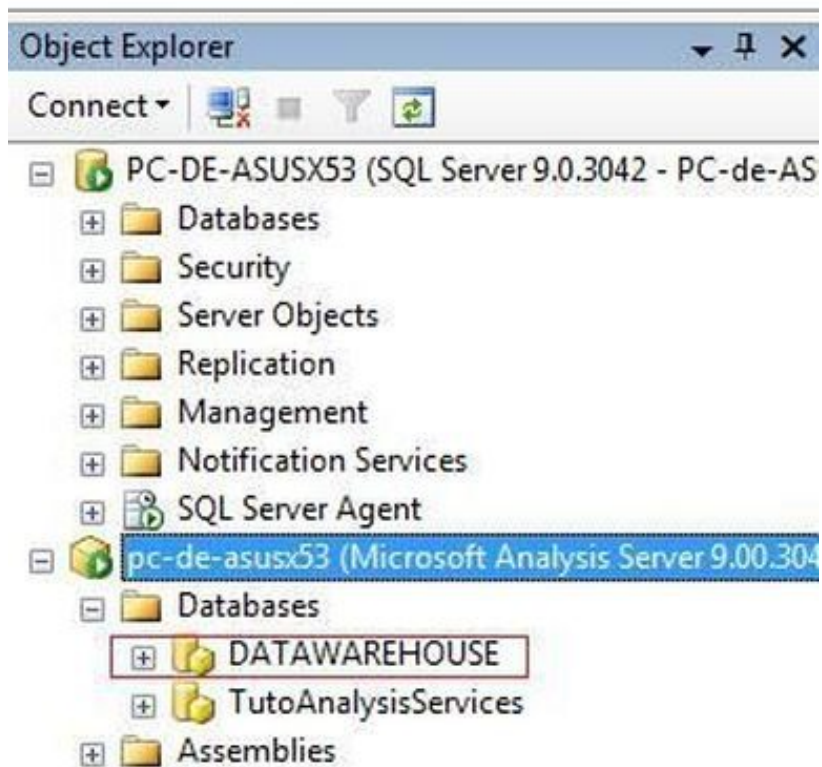
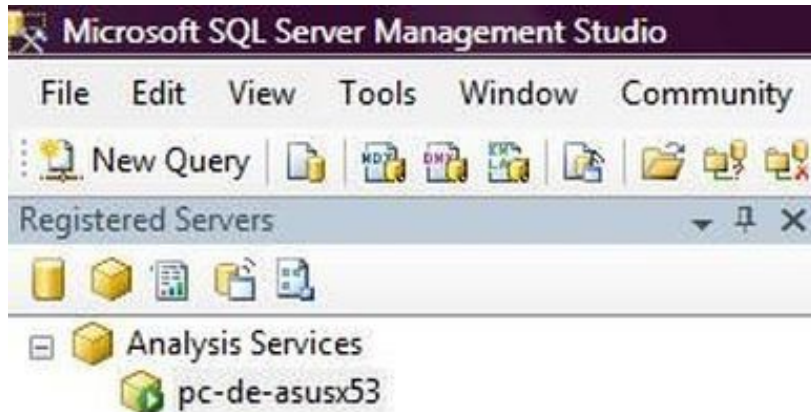
V-B - Management Studio

Maintenant que votre cube est déployé sur une instance Analysis Services, nous sommes prêts à l'exploiter. Pour cela il suffit d'utiliser management studio.

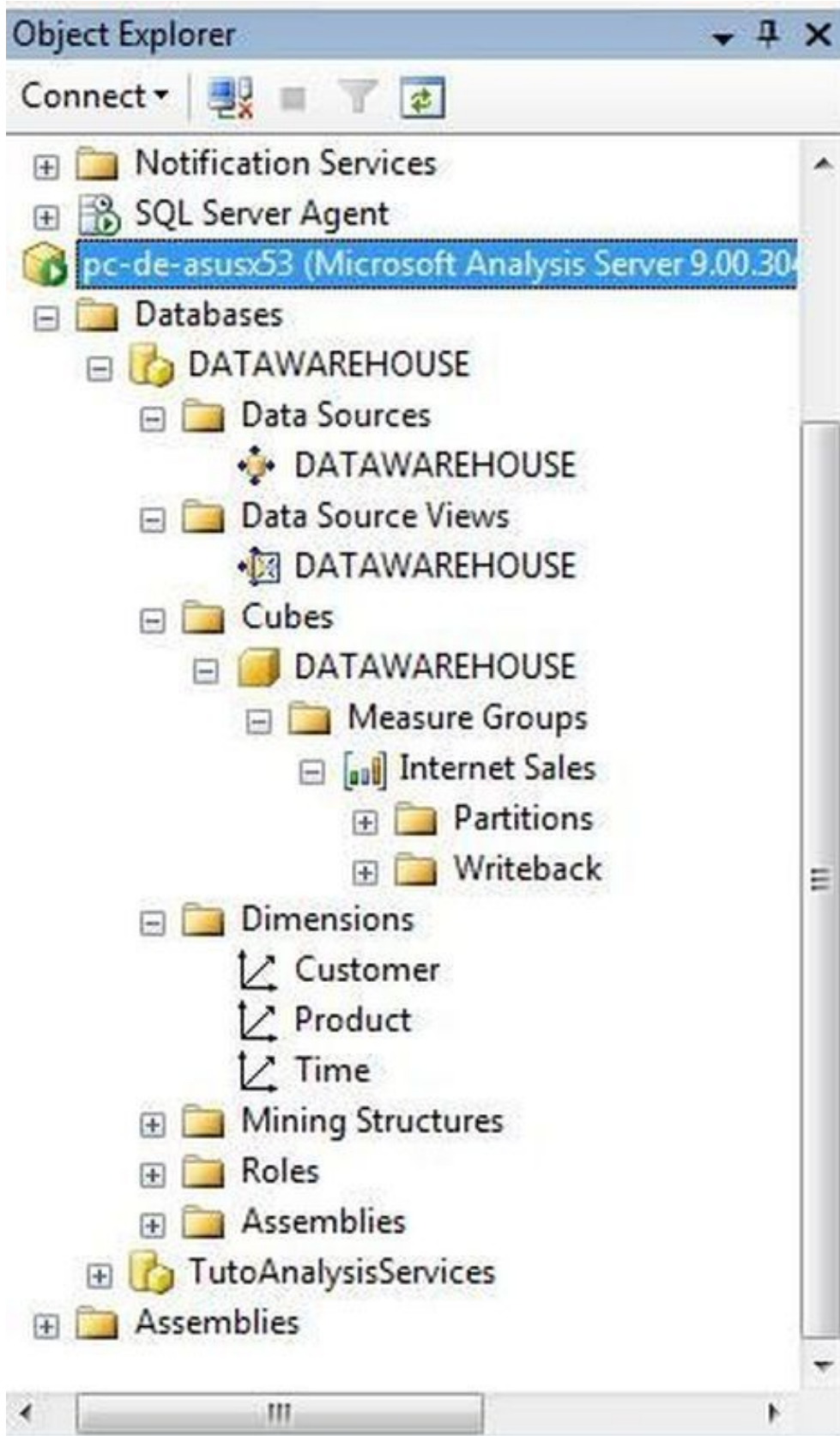
Démarrez SQL Server Management Studio en choisissant Analysis Services et le nom de votre serveur. Votre serveur est le nom de votre machine si vous avez choisi une installation par défaut de SQL server et le nom de votre instance nommée si vous avez une instance nommée.



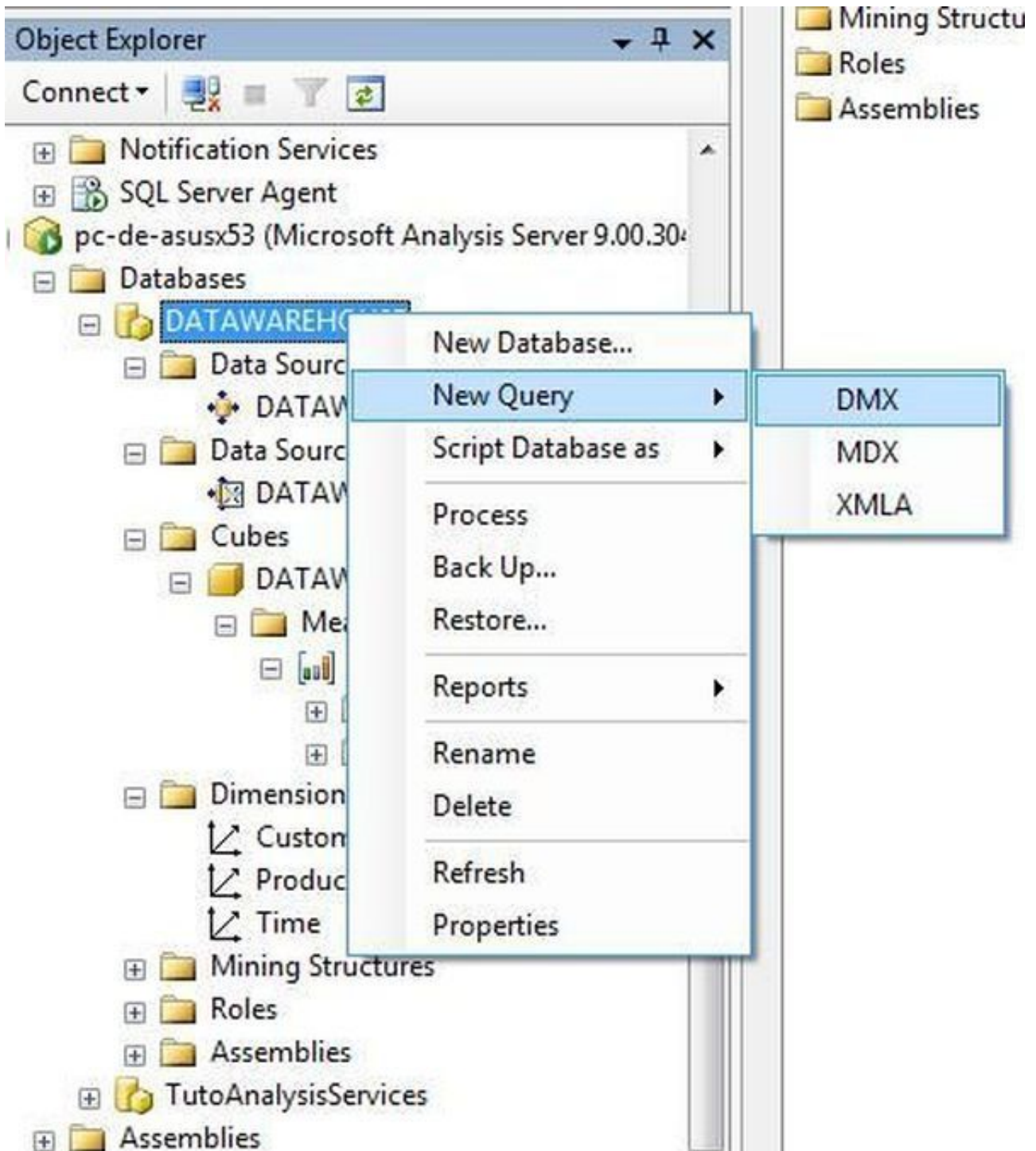
Une fois connecté à Analysis Services, vous pouvez voir votre base de données dimensionnelle **DATAWAREHOUSE** dans l'explorateur de données.



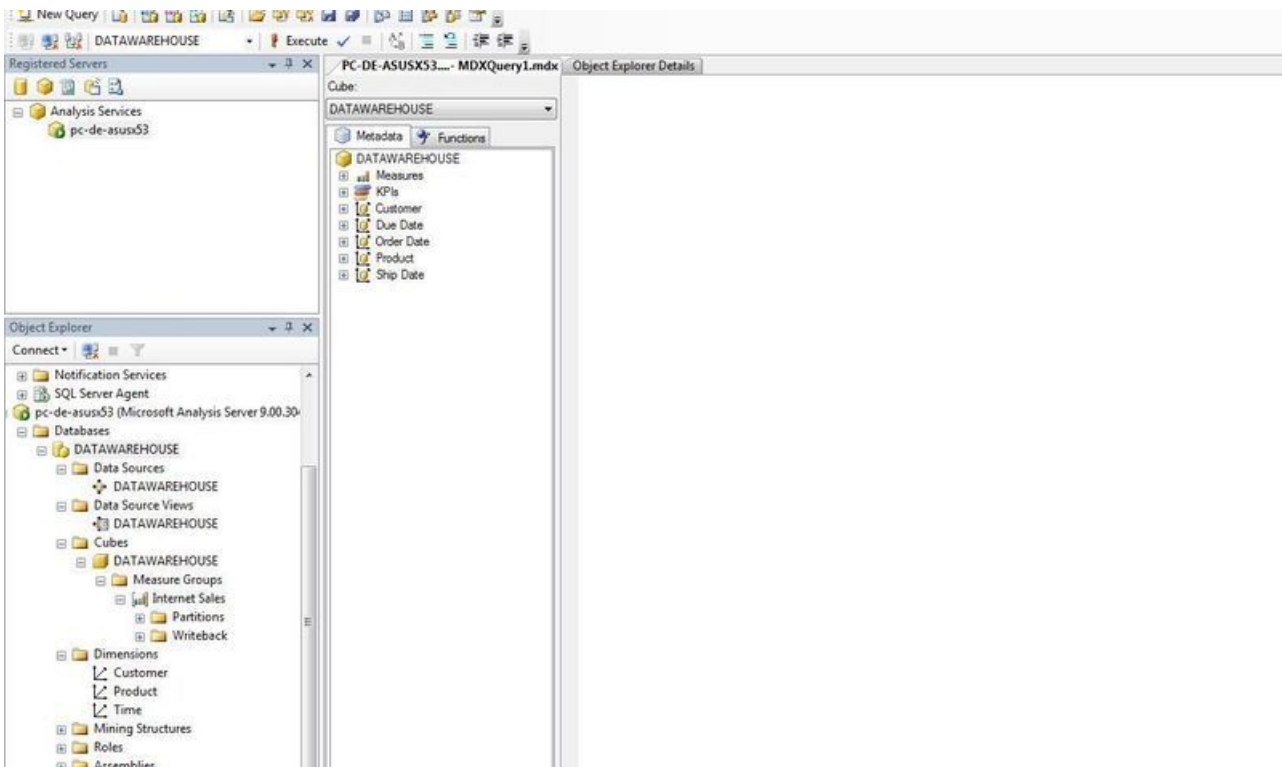
En défilant votre base de données **DATAWAREHOUSE**, vous pouvez y voir votre cube de données, vos dimensions, vos mesures et tous les objets d'analyse définis comme le montre la figure suivante:



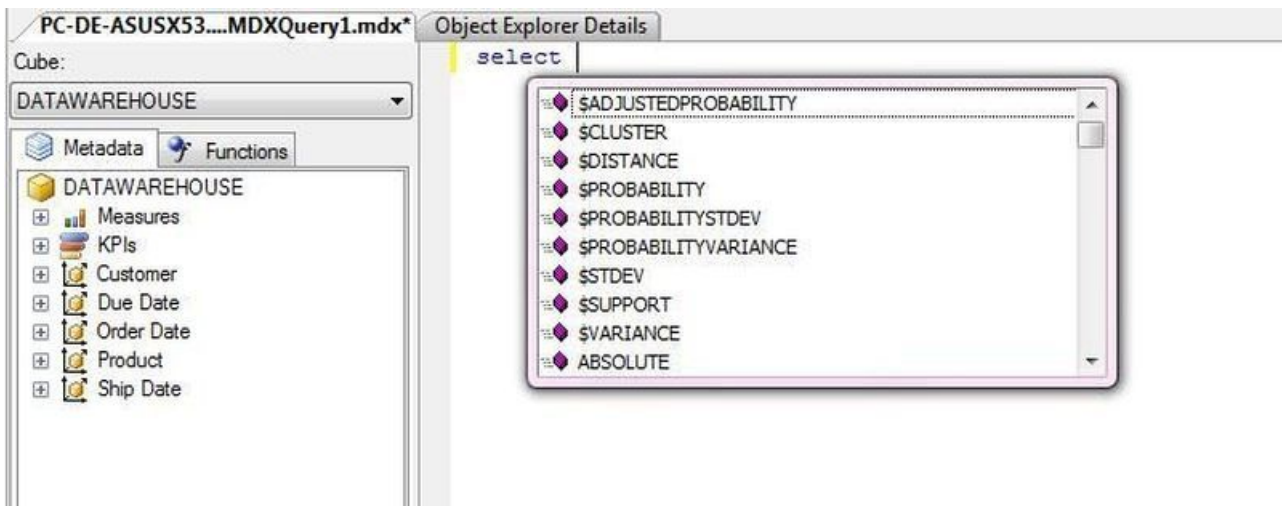
Chosissez alors de faire une requête MDX comme le montre la figure suivante:



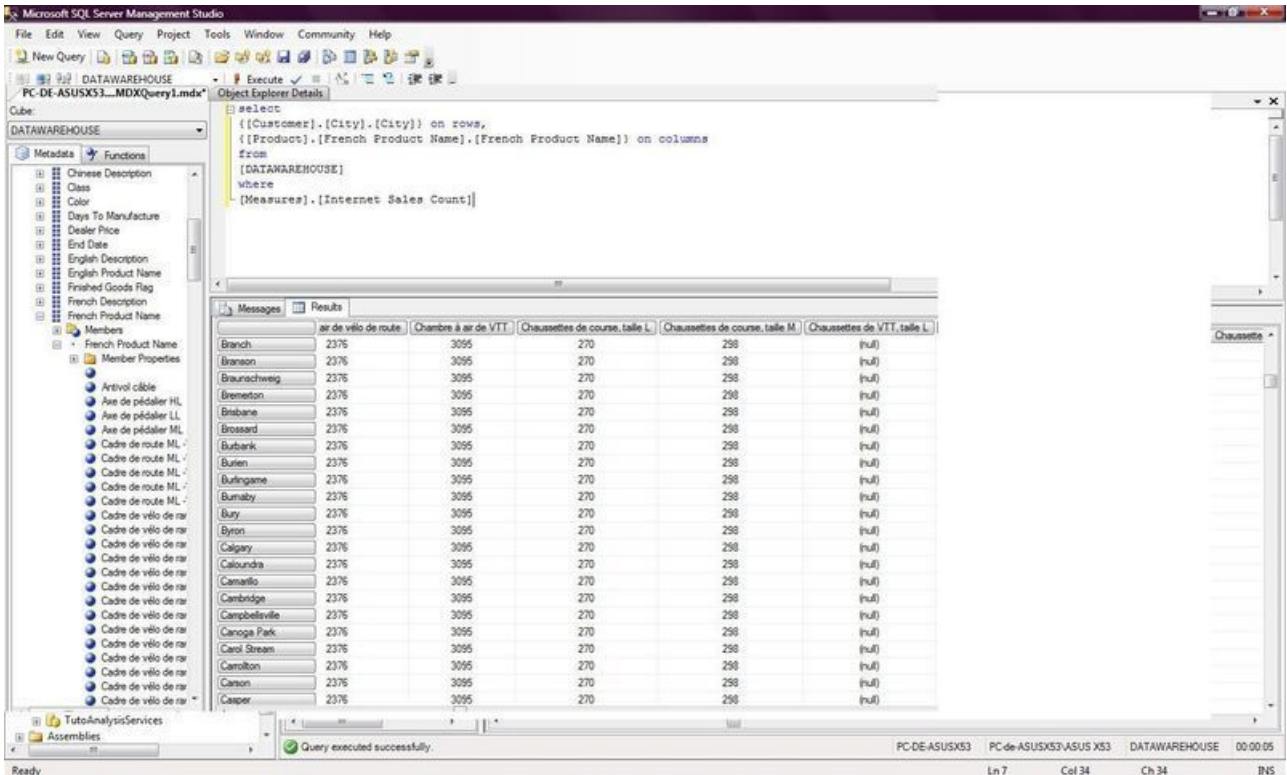
Vous avez alors un panel qui s'ouvre et qui présente les métadonnées et les fonctions disponibles pour votre cube. En d'autres termes, il s'agit d'un panel qui vous sert de **QBE**. C'est à dire qu'au lieu de savoir comment faire une requête **MDX**, on peut le faire assez facilement via ce panel.



Si vous choisissez de taper votre requête MDX, vous voyez apparaître la complétion automatique. Grâce à cet outil vous n'êtes pas obligé de connaître les fonctions MDX. Faites juste " **Ctrl espace**" pour le faire apparaître.

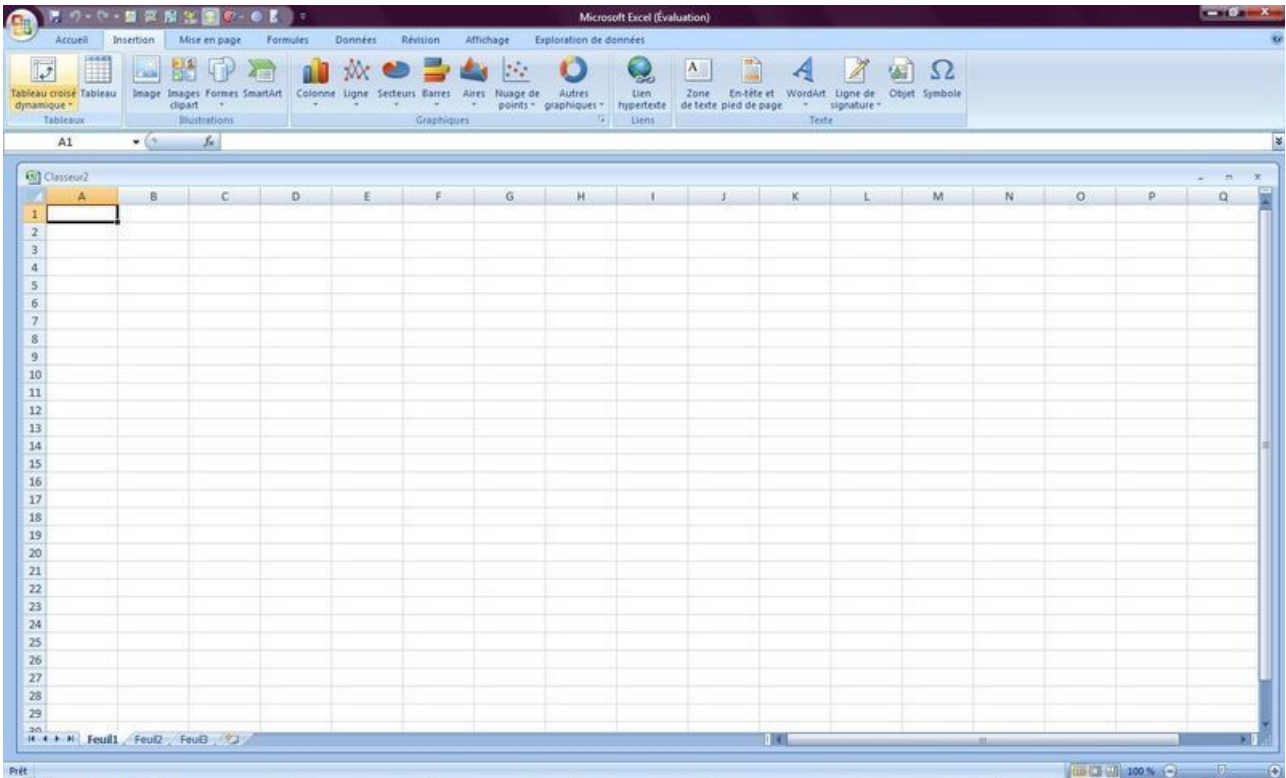


La figure ci-dessous montre un exemple de requête MDX qui montre les ventes par produits et par ville. Cela nous ramènera par exemple à mettre les villes en ligne (on rows) et les produits en colonne (on columns) comme le montre la figure suivante:

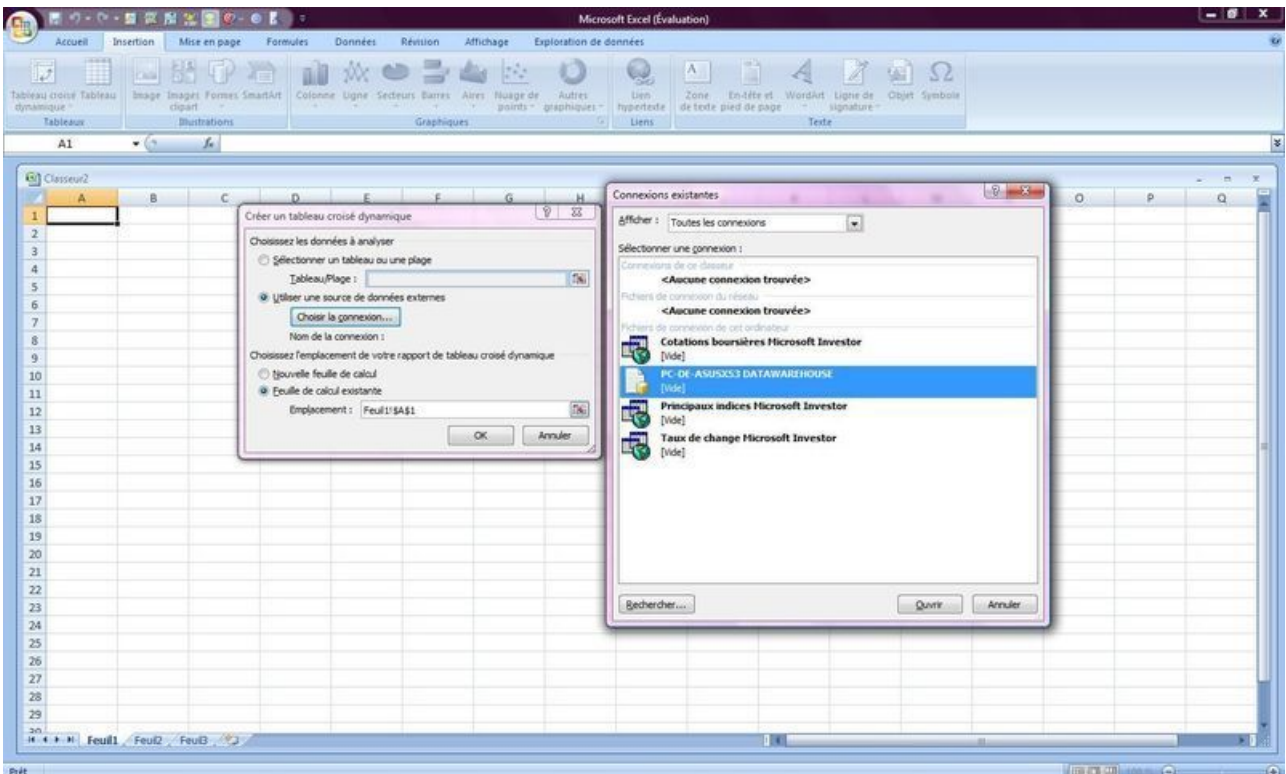


V-C - Excel

La navigation sur les données multidimensionnelles peut aussi se faire sous Excel. Dans mon cas j'utilise Excel 2007 qui commence à avoir pas mal d'add-in pour les outils décisionnels. Pour cela, il faudra se connecter à notre base de données multidimensionnelle **DATAWAREHOUSE**. Commençons par ouvrir Excel puis essayons de faire une requête d'Analyse croisée dynamique comme suit:



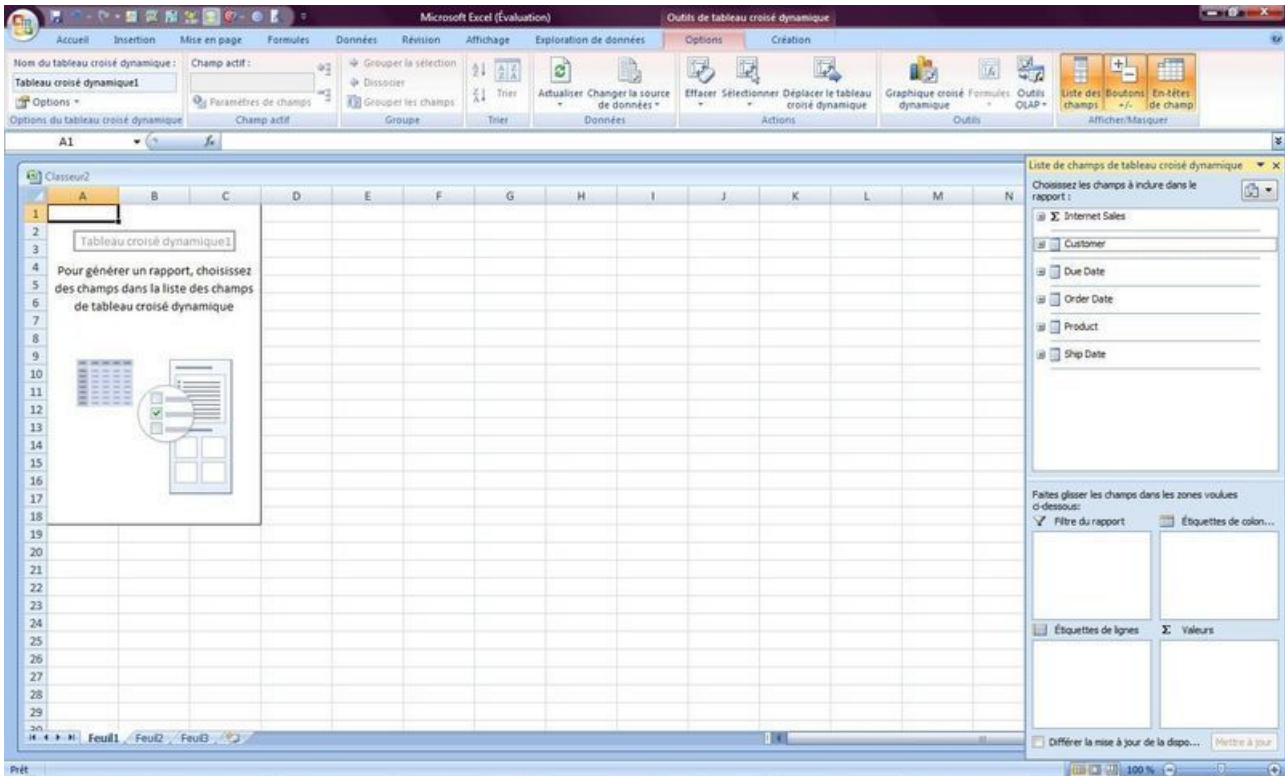
Il faut maintenant nous connecter à notre base **DATAWAREHOUSE** comme le montre la figure suivante:



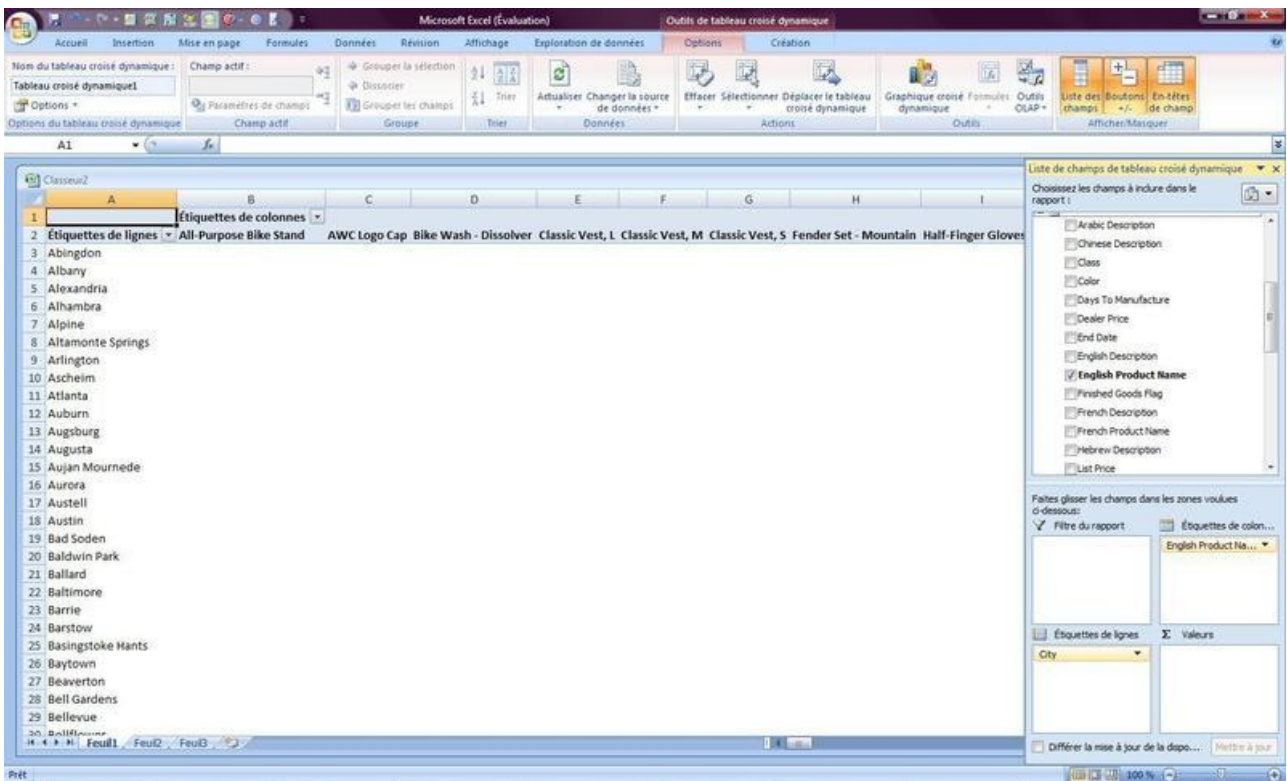
Dans mon cas ma source de données sur la base **DATAWAREHOUSE** avait déjà été définie. Vous pouvez en

définir une nouvelle.

Une fois connecté à votre base d'analyse, vous obtenez l'écran suivant avec toutes les métadonnées à droite. Cela permet d'afficher les données voulues en ligne, en colonne et la mesure à l'intersection.



Ainsi vous obtenez l'image suivante si vous choisissez **vile** en ligne et **produit** en colonne.



VI - Conclusion

Vous venez de voir les différentes étapes pour créer et déployer un cube sous Analysis Services 2005. Ce tutorial ne présente cependant que sommairement les options de création du cube. Pour traiter tous les détails et options de création de cubes, cela nécessiterait plusieurs tutoriaux.