

Atelier 3 :

Slowly Changing Dimensions

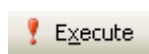
Objectif :

La réalisation d'un package ETL qui permet :

- l'alimentation de la dimension Produit à partir d'une table produit du système transactionnel
- l'insertion des nouvelles lignes dans la table produit du datawarehouse
- la mise à jour des données de la table de datawarehouse
- l'historisation dans le datawarehouse les lignes des produits supprimés

Tâche 1 : Création des tables dans la base de données

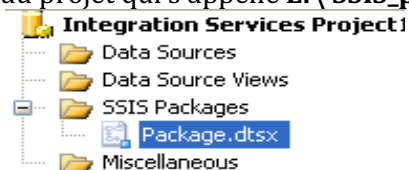
- Ouvrir Microsoft SQL Server Management studio
- Dans le champ server name, mettre un point « . »
- Ouvrir le fichier Script_BD.sql puis cliquer sur exécuter



Résultat : vous venez de créer la base de données source et le datawarehouse ainsi que la table produit de la base source et la dimension produit du datawarehouse

Tâche 2 : Initialiser le projet SSIS

- Démarrer Business Intelligence Studio
- Dans Business Intelligence Studio, créer un nouveau projet qui s'appelle **E:\ SSIS_proj1**



- Renommer le package par défaut à **product.dtsx**

Résultat : vous venez de créer un package ETL qui va permettre l'alimentation de la dimension produit.

Tâche 3 : Définition de la source et de la cible de données

- Dans le package, dans la partie « gestionnaire de connexions », cliquer droite
- Cliquer sur new « OLE DB Connexion »
- Dans la fenêtre de configuration qui s'ouvre cliquer sur « nouveau »
- Entrer les paramètres suivants :
Server name : un point « . »

Sélectionner ou entrer un nom de base de données : Vente_transactionnelle

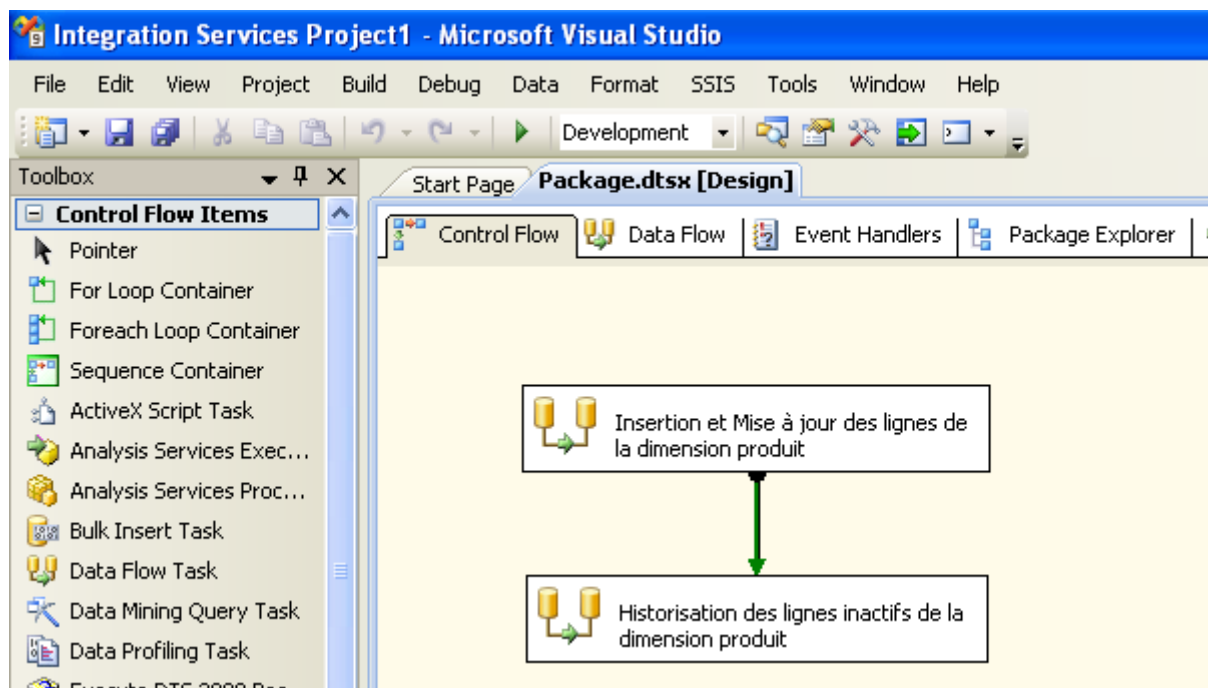
- Cliquer sur OK
- Répéter les mêmes étapes pour créer une connexion vers la base de données « Vente_datawarehouse »

Résultat : vous venez de créer les connexions nécessaires dans le package pour faire l'alimentation

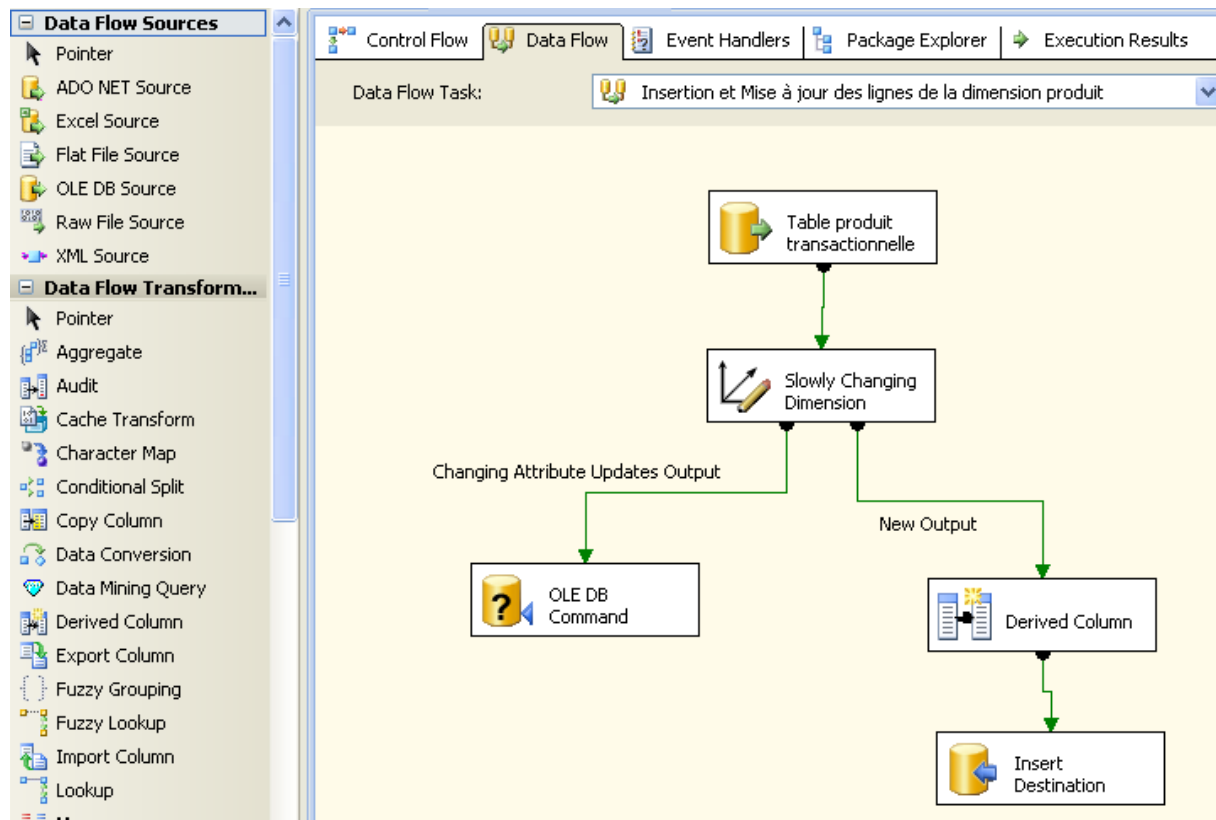
Tâche 4 : Créer un flux de contrôle dans le package

- Dans l'onglet flux du contrôle du package, glisser une tâche « **flux de données** » depuis la palette de conception du flux de contrôle.
- Nommer la nouvelle tâche « **Insertion et Mise à jour des lignes de la dimension produit** ».
- glisser une autre tâche « **flux de données** » depuis la palette de conception du flux de contrôle.
- Nommer la nouvelle tâche « Historisation des lignes inactifs de la dimension produit ».

Résultat : vous venez de créer un flux de contrôle qui contient deux flux de données, le premier sera responsable de l'insertion et la mise à jour de la dimension Produit et le deuxième va permettre l'historisation des lignes inactifs de cette table



Tâche 5 : Conception du flux de données « Insertion et Mise à jour des lignes de la dimension produit »



Créer une Source de données OLE DB qui point sur la table produit transactionnelle

- Double cliquer sur le flux de données « **Insertion et Mise à jour des lignes de la dimension produit** »
- Ajouter un « **OLE DB Source** » depuis la palette.
- Double cliquer sur le composant :
 - dans la liste « gestionnaire de connexion OLE DB » choisir la connexion « Vente transactionnelle »
 - dans la liste mode d'accès aux données choisir « Table ou vue »
 - dans le nom de la table choisir la table « **Produit** »
- Cliquer sur OK

Créer un Composant slowly changing dimension : ce composant sert à comparer les données provenant de la table produit transactionnelle et les données de la dimension produit. Ceci a pour finalité la détection des lignes à insérer et des lignes à mettre à jour

- Ajouter un **slowly changing dimension** depuis la palette
- Lier la source de donnée produit à ce composant en glissant la flèche verte
- Double cliquer sur ce composant pour le configurer puis cliquer sur suivant
- Dans la fenêtre qui s'affiche (voir figure) :
 - dans la liste gestionnaire de connexion choisir « vente_datawarehouse »
 - dans la liste table, choisir la dimension « Produit »
 - faire le mapping entre l'input et les colonnes de la dimension
 - choisir la colonne idproduit come « business key »

Slowly Changing Dimension Wizard

Select a Dimension Table and Keys
Select a dimension table to load and map columns in the transformation input to columns in the dimension table.

Connection manager:
LocalHost.VENTE_DATAWAREHOUSE

Table or view:
[dbo].[PRODUIT]

Input Columns	Dimension Columns	Key Type
	date_insert	
	date_update	
idcategorie	idcategorie	Not a key column
idproduit	idproduit	Business key
	IsCurrent	
libelle_produit	libelle_produit	Not a key column
	package_insert	
prix_unitaire	prix_unitaire	Not a key column

Buttons: Help, < Back, Next >, Finish >>|, Cancel

- Cliquer sur suivant, puis choisir le comportement du composant vis-à-vis les colonnes idcategorie, libelle_produit, prix_unitaire, Qte_stock . choisir pour toute ces colonnes « changing attribute » pour pouvoir les mettre à jour dans le cas où le système trouve une différence entre la source et la destination
- Cliquer sur suivant et cocher l'option

Changing attributes

☒ Change all the matching records, including outdated records, when changes are detected in a changing attribute

- Cliquer sur suivant et décocher l'option

☐ Enable inferred member support

☒ All columns with a change type are null

☐ Use a Boolean column to indicate whether the current record is an inferred member

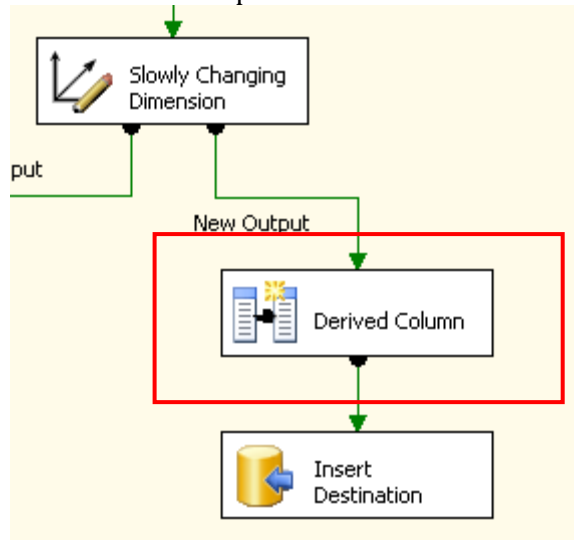
Inferred member indicator:

- Cliquez sur finish

Résultat : vous venez de créer une séquence qui va permettre le chargement de la dimension produit, le composant a créé une sortie pour l'insertion et une autre pour la mise à jour.

Ajouter des colonnes calculées à la dimension produit

- Placer un composant « Colonnes dérivées » dans la sortie « nouvelle sortie »



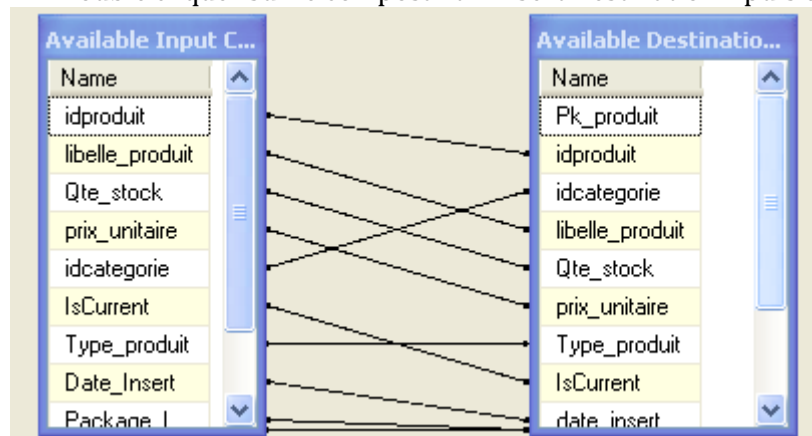
- Double cliquer sur ce composant
 - Créer les colonnes suivantes :
- | | |
|----------------|---|
| IsCurrent | 1 |
| Type_produit | idproduit > 200000 ? "Pack" : "Service" |
| Date_insert | @[System::StartTime] |
| Package_Insert | @[System::PackageName] |
| Date_update | @[System::StartTime] |

Derived Column Name	Derived Column	Expression	Data Type	Length
IsCurrent	<add as new column>	1	four-byte signed int...	
Type_produit	<add as new column>	idproduit > 200000 ? "Pack" : "Service"	Unicode string [DT_...	7
Date_Insert	<add as new column>	@[System::StartTime]	date [DT_DATE]	
Package_Insert	<add as new column>	@[System::PackageName]	Unicode string [DT_...	7
Date_update	<add as new column>	@[System::StartTime]	date [DT_DATE]	

- Cliquer sur OK

Faire le mapping entre les colonnes d'entrée et les colonnes de sortie

- Double cliquer sur le composant « Insert Destination » puis compléter le mapping



Remplir la date de mise à jour pour les lignes qui vont être mises à jour

- Dans le composant « OLE DB Command », dans l'onglet « propriétés du composant », ouvrir `_sqlCommand`

----	----
SqlCommand	UPDATE [dbo].[PRODUIT] SET [idcategorie] = ?,[libelle_pro

- Remplacer la requête existante par la requête suivante :

```
UPDATE [dbo].[PRODUIT] SET [idcategorie] = ?,[libelle_produit] = ?,[prix_unitaire] =  
?,[Qte_stock] = ?,date_update = Getdate() WHERE [idproduit] = ?
```

Nous avons ajouté une mise à jour de la colonne date_update pour tracer la date de mise à jour de la ligne.

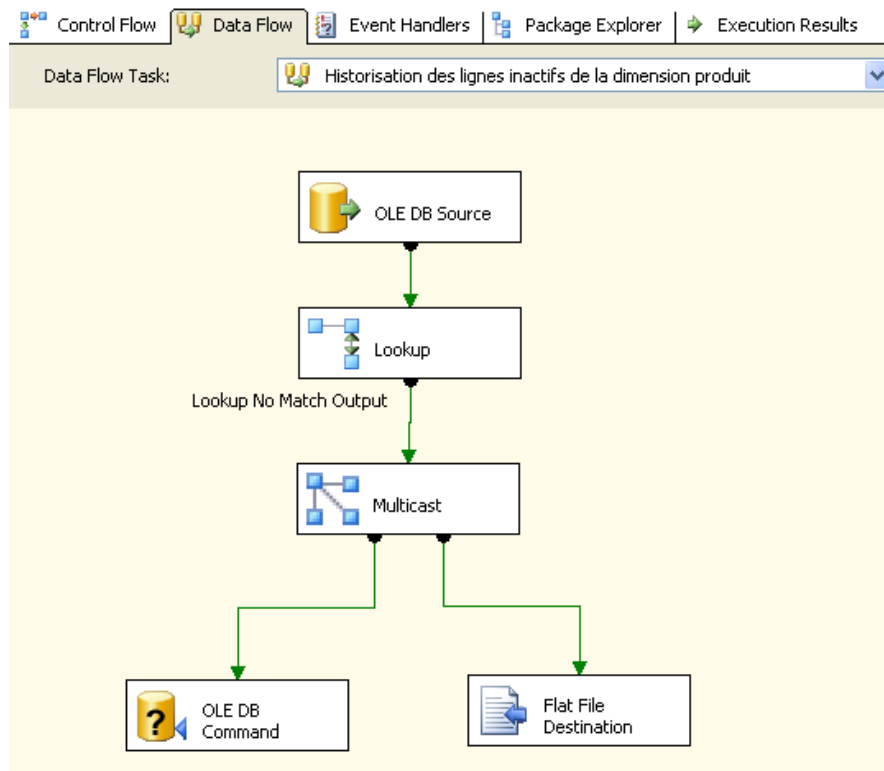
- Enregistrer et fermer

Résultat : vous avez terminé la création du flux de données qui va permettre l'insertion des nouvelles lignes dans la dimension et la mise à jour des lignes qui ont changé.

Test à faire :

- A la fin de cette étape vous pouvez lancer le chargement pour visualiser l'alimentation des données dans la dimension
- Vous pouvez faire des modifications sur les données de la table produit depuis SQL Sever Management Studio et relancer, pour voir le mécanisme de mise à jour

Tâche 6 : Conception du flux de données « Historisation des lignes inactifs de la dimension produit »



Créer un composant OLE DB Source qui va nous ramener les données de la dimension produit

- Ajouter le composant **OLE DB Source**.
- Double cliquer sur ce composant.
- **Choisir la connexion « Vente_transactionnelle », choisir SQL Commande et entrer la commande suivante**

```
SELECT idproduit, idcategorie, libelle_produit, Qte_stock, prix_unitaire FROM  
dbo.PRODUIT where isCurrent = 1
```

Cette commande va nous ramener les lignes actives de datawarehouse, afin de pouvoir chercher dans le système transactionnel et voir si elles sont toujours actives ou non. Une ligne active a la valeur 1 dans la colonne isCurrent, une ligne inactive a la valeur 0 dans la colonne isCurrent

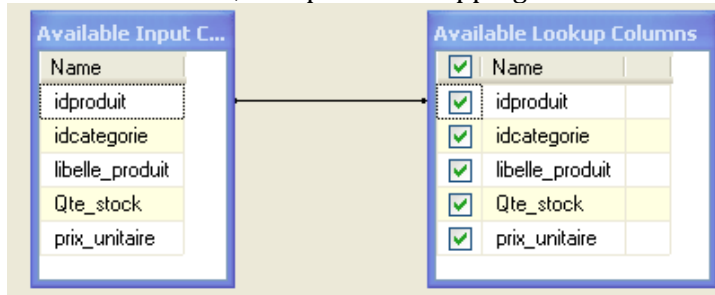
Créer un composant Lookup (recherche) qui va chercher ces lignes dans le système transactionnel

- Ajouter le composant **Recherche**

General
Connection
Columns
Advanced
Error Output

- Dans connexion choisir la connexion « Vente_Transactionnelle » puis cocher utiliser une table ou vue et choisir la table produit

- Dans Columns, compléter le mapping comme suit :



Ce mapping va permettre la comparaison entre les données de la dimension du datawarehouse et de la table transactionnelle en se basant sur la clé « idproduit »

- Cliquer sur OK

Créer un composant multidiffusion qui va permettre d'envoyer deux flux, un pour historiser les lignes en sortie et un autre pour les mettre dans un fichier texte

- Insérer le composant « multidiffusion ».
- Insérer un composant « Destination fichier plat » et un composant « OLE DB Command » puis lier les deux sorties de la multidiffusion vers ces deux composants

Créer un fichier texte qui va contenir les lignes historisées

- Créer un fichier texte sur le bureau.
- Double cliquer sur le composant « Destination fichier plat », faire nouveau puis OK
- Dans « nom du fichier », faire parcourir et pointer sur le fichier que vous avez créé

Historiser les lignes dans le datawarehouse

- Double cliquer sur le composant « Commande OLE DB »
- Dans gestionnaire de connexion choisir « vente_datawarehouse »

Name	Connection Manager	Description
OleDbConnection	LocalHost.VENTE_DATAWARE...	The OLE DB runtime connectio...

- Dans le deuxième onglet « propriétés du composant » dans la commande sql mettre

Update dbo.produit SET isCurrent = 0, date_update = Getdate() where idproduit = ?

SqlCommand	Update dbo.produit SET isCurrent = 0, date_update = Getdate() where idproduit = ?
------------	---

- Cliquer sur OK
- Cliquer sur Enregistrer.

Résultat : vous avez terminé la création du flux de données qui va l'historisation des lignes inactives dans le datawarehouse.

Test à faire :

- Supprimer une ligne depuis la table produit de la base vente_transactionnelle
- lancer le traitement et vérifier la colonne IsCurrent de cette ligne dans le datawarehouse puis chercher dans le fichier texte si elle a été enregistrée