

## Atelier5 : Slowly Changing Dimensions

---

### Les paramètres requis sont manquants ou erronés. **Tâche 1 : Création des tables dans la base de données**

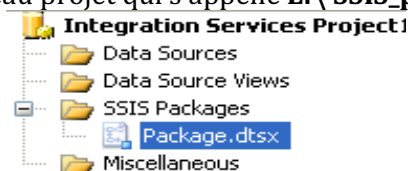
- Ouvrir Microsoft SQL Server Management studio
- Dans le champ server name, mettre un point « . »
- Ouvrir le fichier Script\_BD.sql puis cliquer sur exécuter



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 »

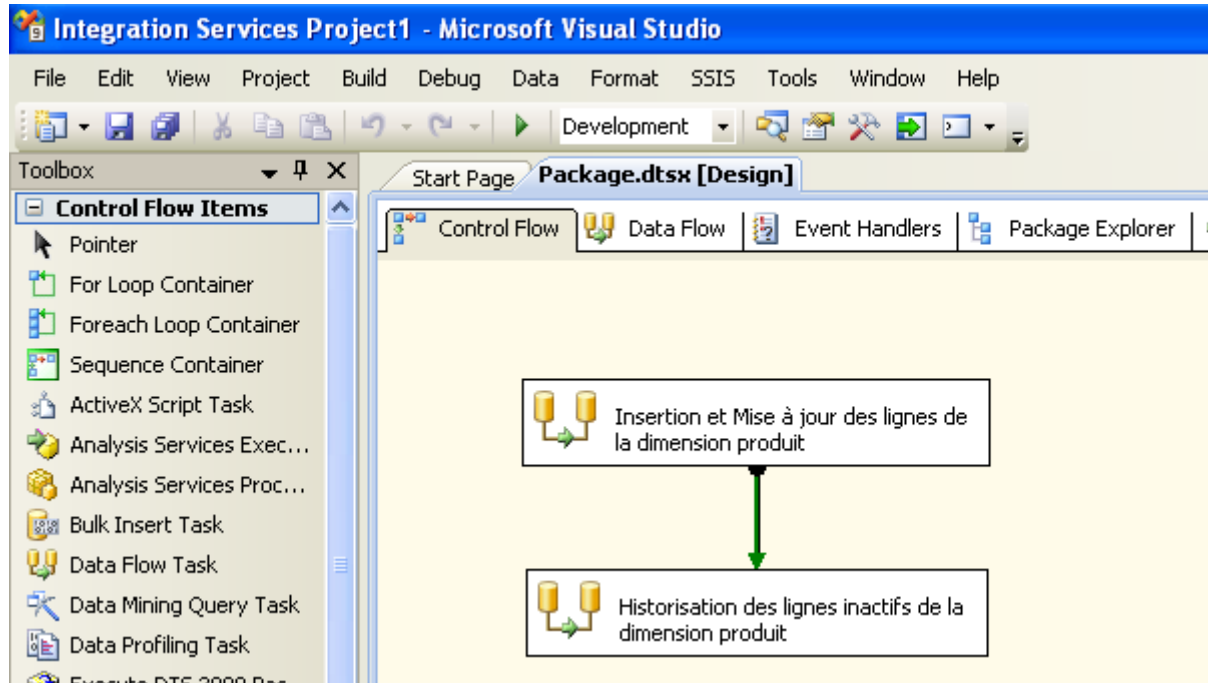
Résultat : vous venez de créer les connexions nécessaires dans le package pour faire l'alimentation

### **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** ».

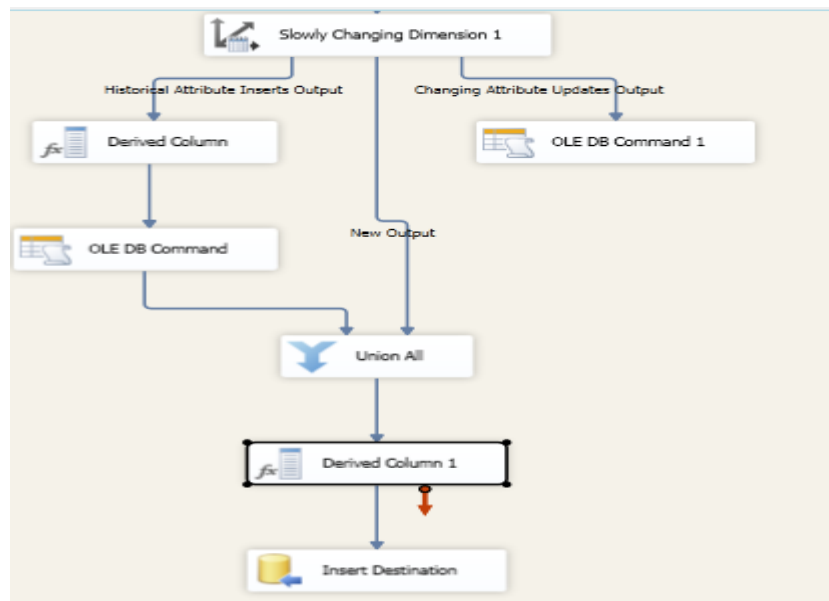
Résultat : vous venez de créer un flux de contrôle qui contient deux flux de données, le premier sera responsable de l'insertion et la mise à jour de la dimension Produit et le deuxième va permettre l'historisation des lignes inactifs de cette table



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

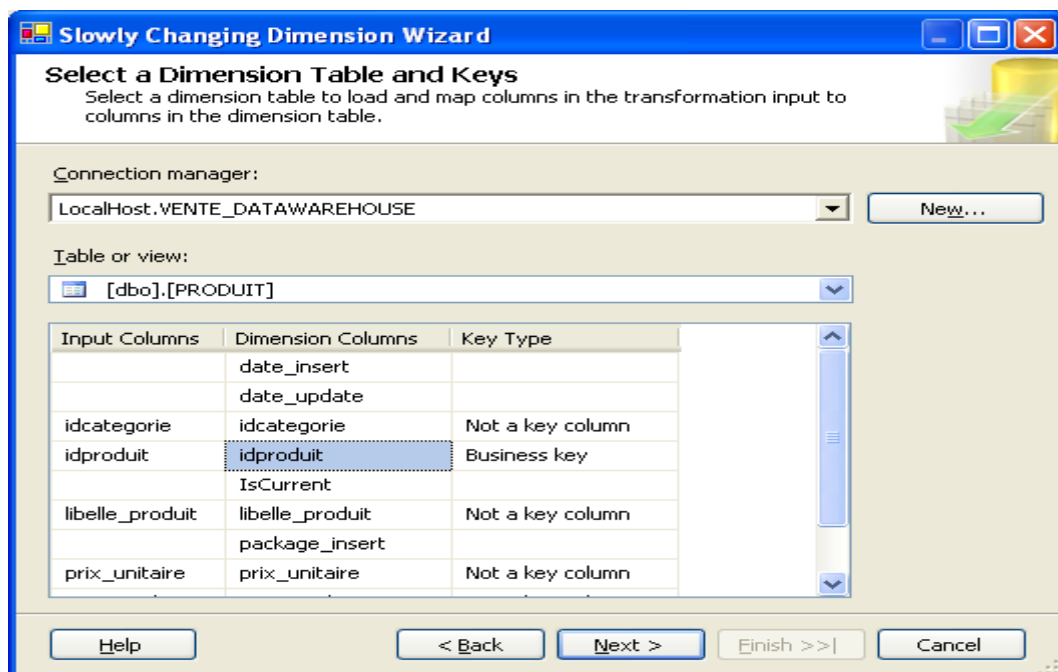
Créer une Source de données OLE DB qui point sur la table produit transactionnelle

- 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 »



- Cliquer sur suivant, puis choisir le comportement du composant vis-à-vis les colonnes idcategorie, libelle\_produit,, Qte\_stock . choisir pour toute ces colonnes « changing attribute » (type SCD 1) pour pouvoir les mettre à jour dans le cas où le système trouve une différence entre la source et la destination et « Historicale attribute » (type SCD 2) pour prix-unitaire

Dimension Columns	Change Type
idcategorie	Changing attribute
libelle_produit	Changing attribute
prix_unitaire	Historical attribute
Qte_stock	Changing attribute

- 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

- Cliquez sur le bouton « Next » pour passer à l'écran suivant de l'Assistant ; dans l'écran Suivant, vous devez spécifier (comme nous avons au moins une colonne du type SCD2) une colonne de date de début (entrée en vigueur) et une colonne de date d'expiration et une variable pour définir la valeur de la date, comme dans la figure

Slowly Changing Dimension Wizard

**Historical Attribute Options**  
You can record historical attributes using a single column or start and end date columns.

☐ Use a single column to show current and expired records  
Column to indicate current record: CurrentFlag  
Value when current: True  
Expiration value: False

☒ Use start and end dates to identify current and expired records  
Start date column: EffectiveDate  
End date column: ExpirationDate  
Variable to set date values: System::StartTime

Help < Back Next > Finish >>| Cancel

- 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

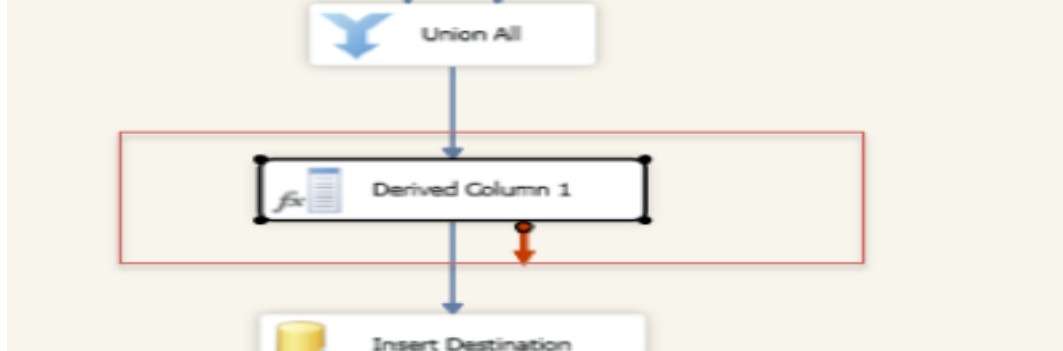
Inferred member indicator:

- 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

- Se placer sur le composant « Colonnes dérivées » dans la sortie « nouvelle sortie »



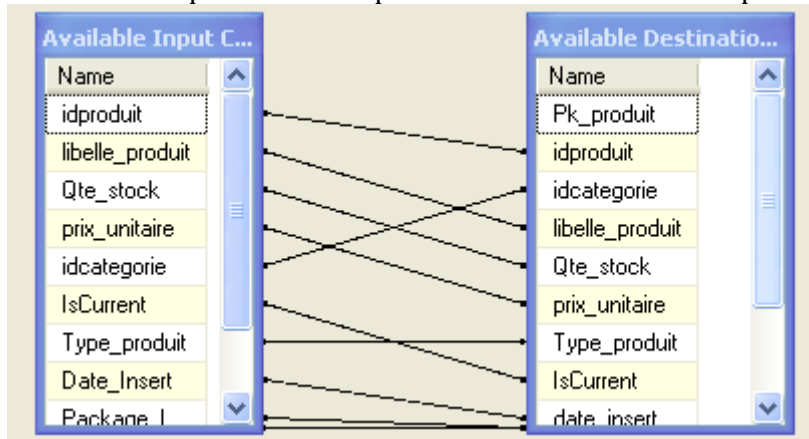
- Double cliquer sur ce composant
  - Créer les colonnes suivantes :
- |                |   |
|----------------|---|
| IsCurrent      | 1                                       |
| Type_produit   | idproduit > 200000 ? "Pack" : "Service" |
| Date_insert    | @[System::StartTime]                    |
|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

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

- 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`

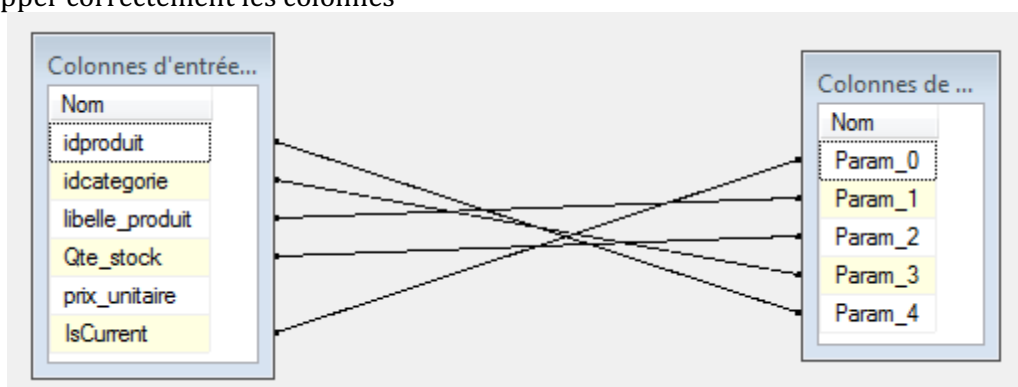
<code>_sqlCommand</code>	<code>UPDATE [dbo].[PRODUIT] SET [idcategorie] = ?,[libelle_pro</code>
--------------------------	--

- Remplacer la requête existante par la requête suivante :

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

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

- Mapper correctement les colonnes



- 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 :

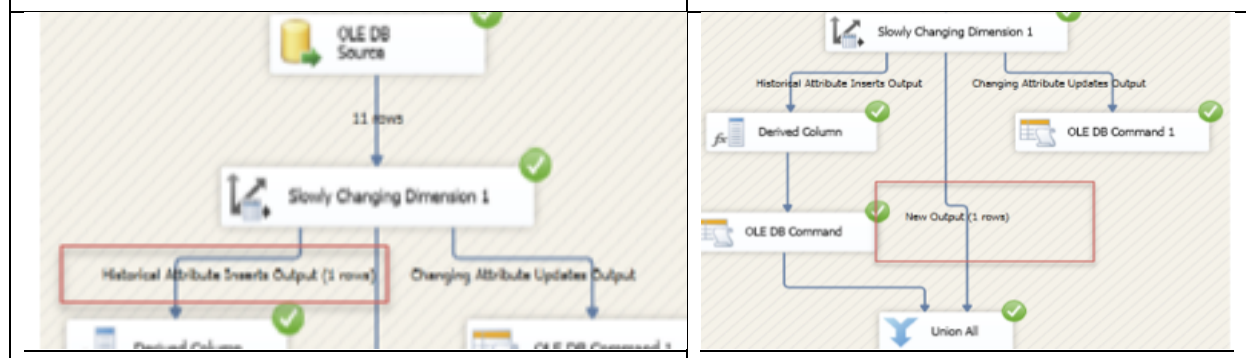
- A la fin de cette étape vous pouvez lancer le chargement pour visualiser l'alimentation des données dans la dimension
- Vous pouvez faire des modifications sur les données de la table produit depuis SQL Sever Management Studio et relancer, pour voir le mécanisme de mise à jour

```
update produit set prix_unitaire=500 where
idproduit=200012
```

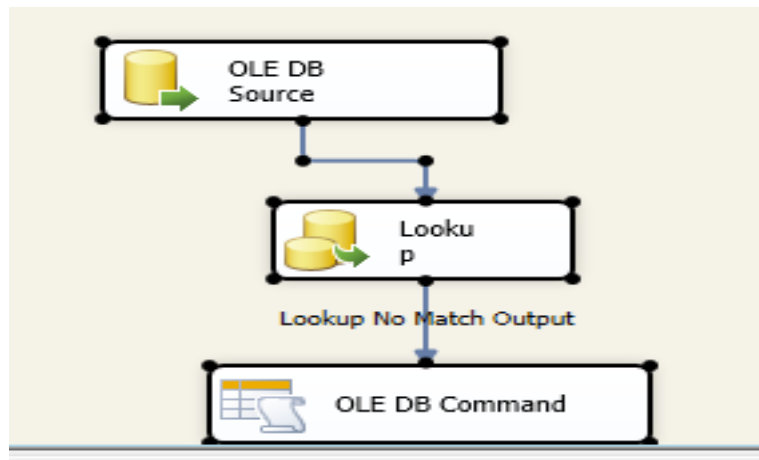
->(1 row(s) affected)

```
insert into produit values
(300012,400001, 'SmartPhone', 22,200);
```

->(1 row(s) affected)



## 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

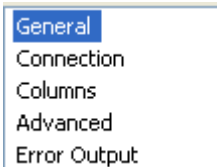
- Ajouter le composant **OLE DB Source**.
- Double cliquer sur ce composant.
- **Choisir la connexion « vente\_datawarehouse », choisir SQL Commande et entrer la commande suivante**

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

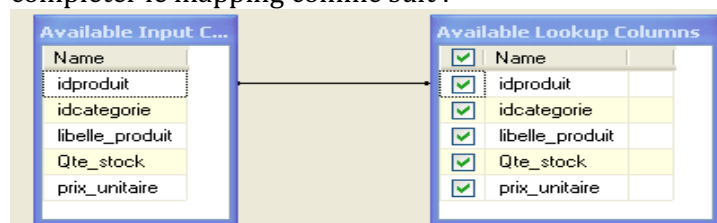
Cette commande va nous ramener les lignes actives de datawarehouse, afin de pouvoir chercher dans le système transactionnel et voir si elles sont toujours actives ou non. Une ligne active a la valeur 1 dans la colonne isCurrent, une ligne inactive a la valeur 0 dans la colonne isCurrent

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

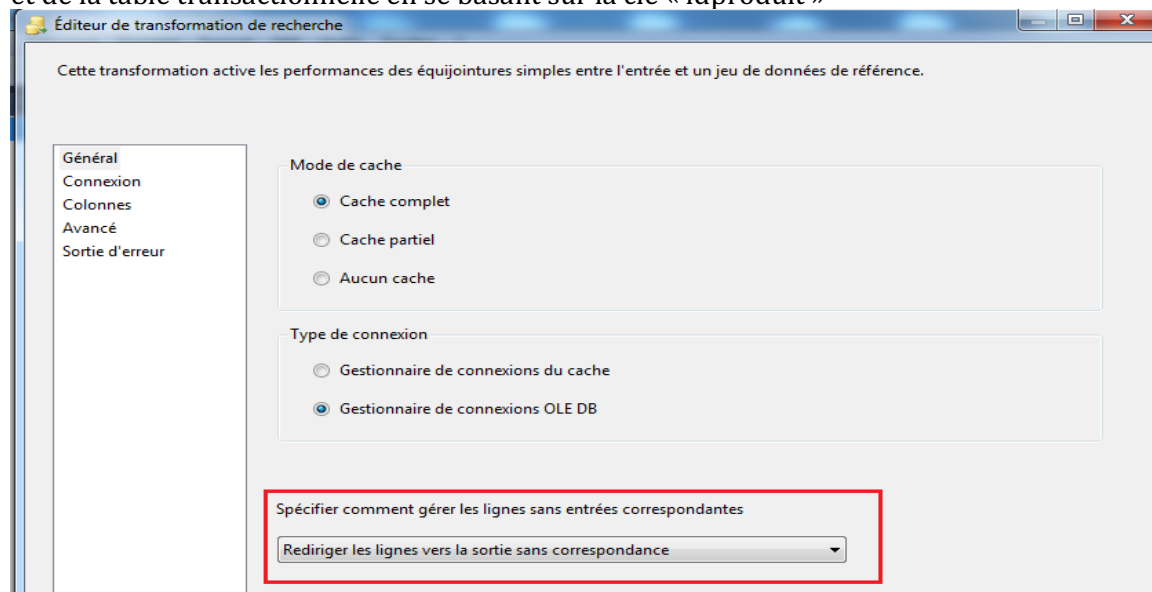
- Ajouter le composant **Recherche**



- 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 »



- Rediriger les lignes vers la sortie sans correspondance
- Cliquer sur OK
- Le lien sera spécifié avec le composant suivant comme « no match »

#### 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()
------------	---

- 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.

