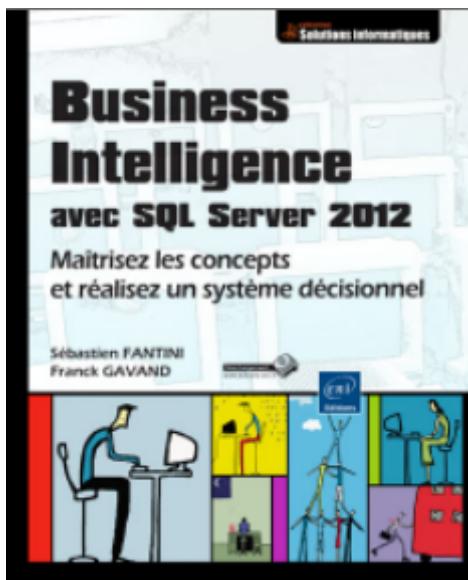


Business Intelligence avec SQL Server 2012



Par Sébastien FANTINI - Franck GAVAND

Date de publication : 25 décembre 2012

Ce chapitre est publié dans le cadre du partenariat entre Developpez et les éditions ENI.  [Commandez le livre entier.](#)

Ce livre sur la Business Intelligence (BI) avec SQL Server 2012, s'adresse à tous les membres d'une équipe décisionnelle : chef de projet, architecte, développeur ETL, développeur de rapports, service Aide à la Maîtrise d'Ouvrage (AMO). Du débutant au technicien expérimenté, le lecteur bénéficiera d'une approche métier du décisionnel.

1 - Découverte de SSIS.....	3
2 - Réaliser son premier flux SSIS.....	7
2-1 - Réaliser le chargement du budget d'un seul site.....	7
2.2 - Charger les données de budget à partir de plusieurs fichiers Excel.....	17
3 - Développer des flux ET L pour le décisionnel.....	22
3-1 - Déroulement de l'exécution d'un processus ETL.....	22
3-2 - Réaliser un flux pour charger le sas de données.....	24
3-3 - Réaliser un flux pour charger une dimension.....	29
3-3-1 - Cas d'une dimension standard.....	29
3-3-2 - Cas d'une dimension en SCD.....	32
3-4 - Réaliser un flux pour charger une table de faits.....	40
4 - L'audit des flux ETL.....	45
4-1 - Les objectifs de l'audit de flux ETL.....	45
4-2 - Conception d'un système d'audit de flux.....	46
4-3 - Exemple de flux avec audit.....	53
5 - Gestion des paramètres de flux et mise en production.....	59
5-1 - Paramétrage des flux.....	59
5-2 - Création du catalogue Integration Services.....	62
5-3 - Déploiement du projet SSIS sur le serveur de développement.....	65
5-4 - Les environnements.....	67
5-5 - Mise en production du projet SSIS.....	72
5-6 - Planifier un flux SSIS.....	74

1 - Découverte de SSIS

Au cours des chapitres précédents, vous avez appris à modéliser un entrepôt de données. L'idée était de faire abstraction des sources de données disponibles dans votre société. Au cours de ce chapitre, vous allez apprendre et comprendre comment va se réaliser la remontée des données du système source vers un entrepôt de données. La principale difficulté est que celui-ci dispose d'une modélisation dimensionnel le conforme, très éloignée de la structure de vos données actuelles.

Dans la gamme SQL Server, l'outil qui va permettre de réaliser le chargement de ces données est SQL Server Integration Services (SSIS).

SSIS a deux aspects :

- Un aspect classique avec une logique de flux de tâches, organisées par des règles de précédence. Cet aspect est appelé Flux de contrôles .
- Un aspect plus spécifique au décisionnel, avec une logique purement E-T-L. Cet aspect est appelé Flux de données .

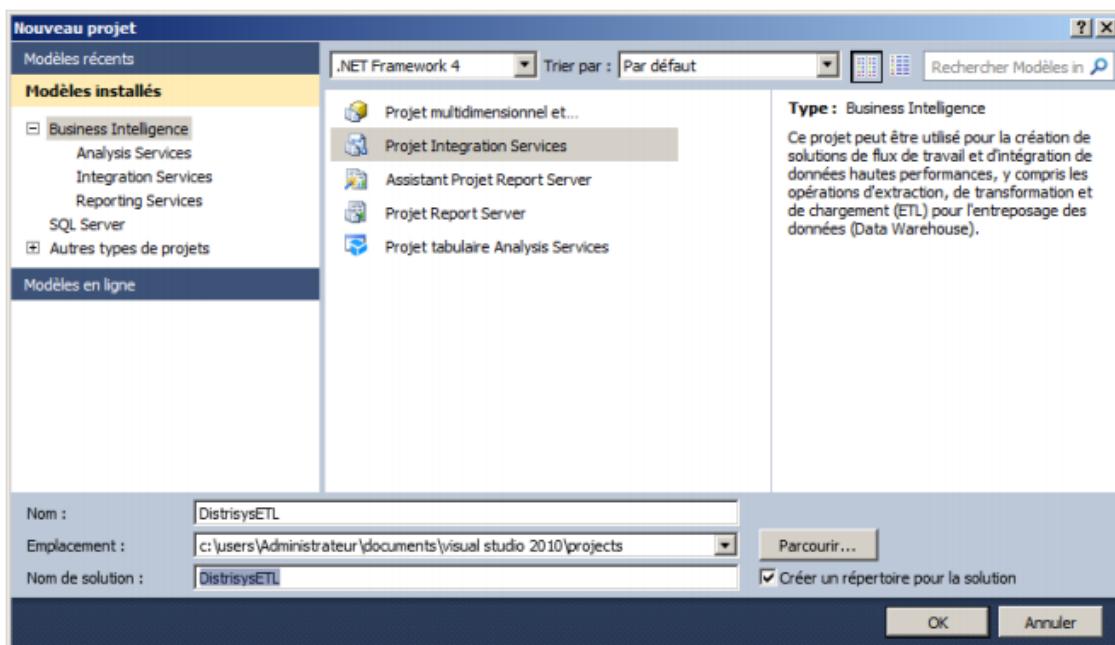
On peut utiliser SSIS sans pour autant faire de l'ETL. Par exemple, vous pouvez vous servir de SSIS pour exécuter des tâches de maintenance de bases de données, pour lancer une suite de batch un peu complexe ou pour réaliser de la réPLICATION de données.

Toutefois, SSIS est aussi un ETL. Le monde de l'ETL a ses codes et ses règles issues de ces quinze dernières années. L'objectif du chapitre, au-delà de la compréhension de ce qu'est l'outil SSIS, est de vous faire découvrir certaines de ces pratiques bien spécifiques au monde du décisionnel. Des pratiques auxquelles SSIS est assez bien adapté.

Un peu comme pour toute la gamme SQL Server, le développement des flux se fera sous SQL Server Data Tools (SSDT). On utilisera en revanche SQL Server Management Studio pour l'administration et l'exploitation. Découvrons ensemble dès à présent l'interface de développement :

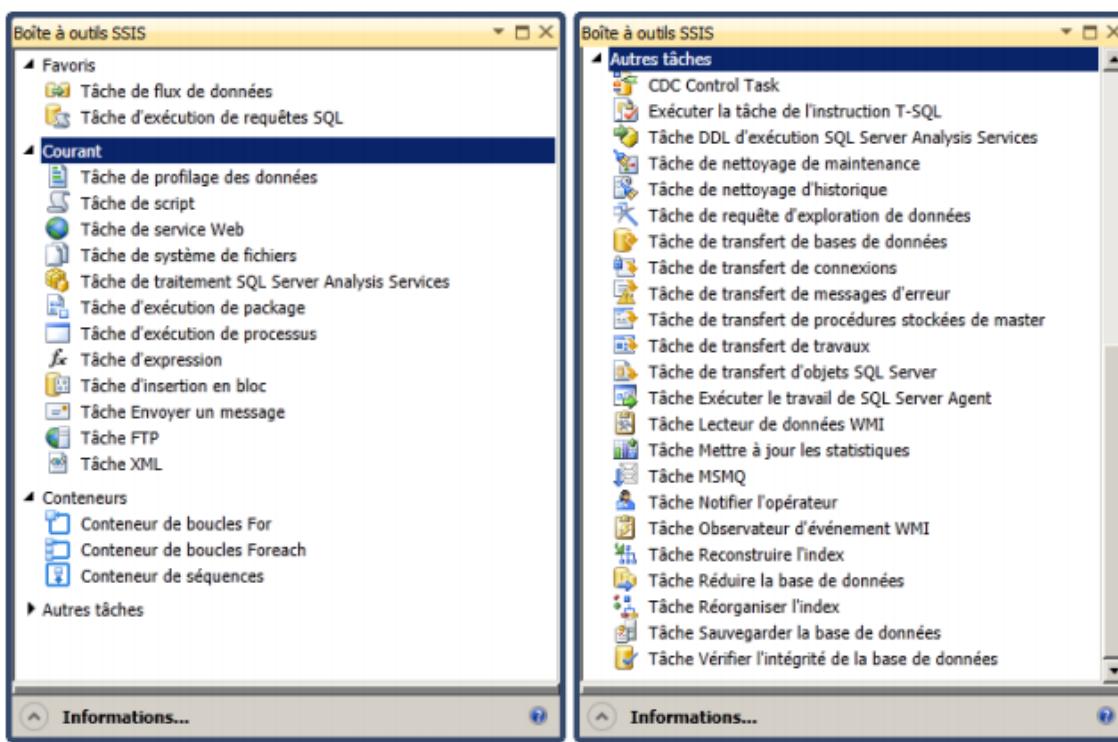
- Ouvrez SSDT.
- Cliquez dans la barre de menu sur **Fichier - Nouveau - Projet**.

Dans la fenêtre **Nouveau projet**, sélectionnez **Projet Integration Services** , puis saisissez le nom et l'emplacement du projet comme ci-dessous :



Le projet s'ouvre par défaut sur l'onglet **Flux de contrôle** d'un package vide. Un package est un fichier au format XML à l'extension **.dtsx**.

Sur le côté gauche, ouvrez la boîte à outils pour découvrir les objets du flux de contrôle disponibles.



Boîte à outils SSIS

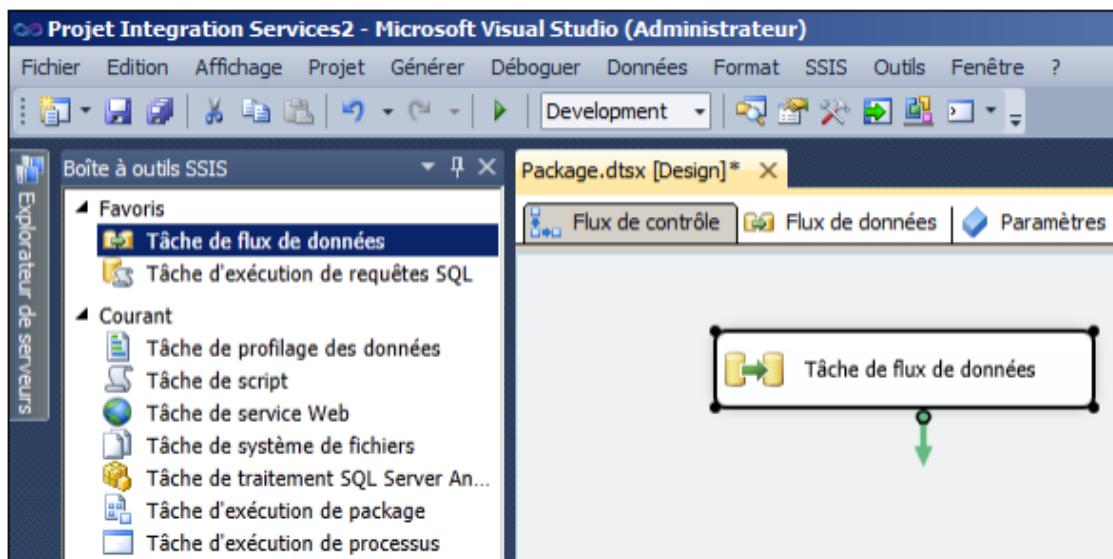
Les tâches disponibles donnent une assez bonne idée du rôle que l'on pourrait faire jouer à SSIS et de ses possibilités : connexion à un service web, exécution de requête SQL, exécution d'application, écriture et exécution de scripts, connexion à un serveur FTP, tâche de traitement de SSAS, tâche de sauvegarde de la base de données...

Dans un flux décisionnel, les tâches de flux de contrôle vont avoir des fonctions de support et d'orchestration, mais ce ne sont pas ces tâches qui vont faire à proprement parlé le chargement des données.

Attention, dans le monde du décisionnel, un entrepôt de données ne se charge pas avec de simples requêtes SQL. Vous verrez que les exigences de traçabilité et de maintenance de tels flux sont trop élevées pour que des requêtes SQL remplissent ce rôle correctement.

Le chargement de données va se réaliser avec la **tâche de flux de données**. Découvrons cet aspect du produit :

Glissez et posez la tâche de flux de données dans la zone de travail centrale.

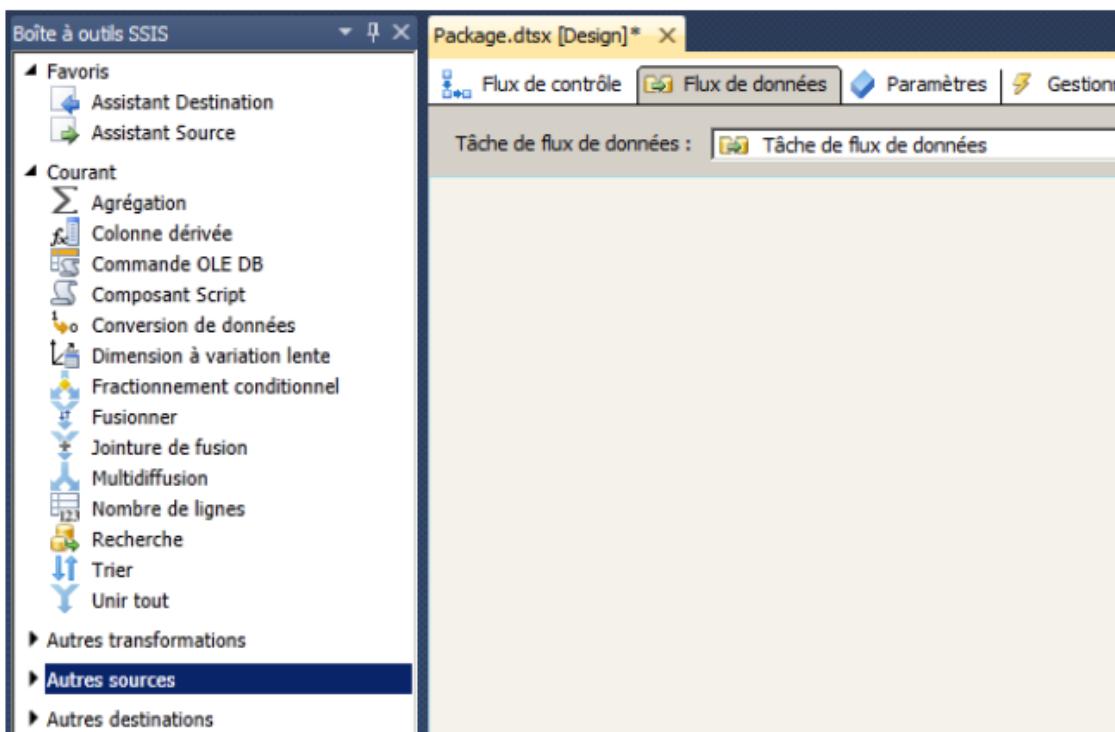


Ajout d'une tâche de flux de données

Puis double cliquez sur la tâche de flux de données pour accéder à l'onglet Flux de données.

Vous noterez que la barre d'outils propose maintenant de nouvelles tâches organisées autour de trois thématiques :

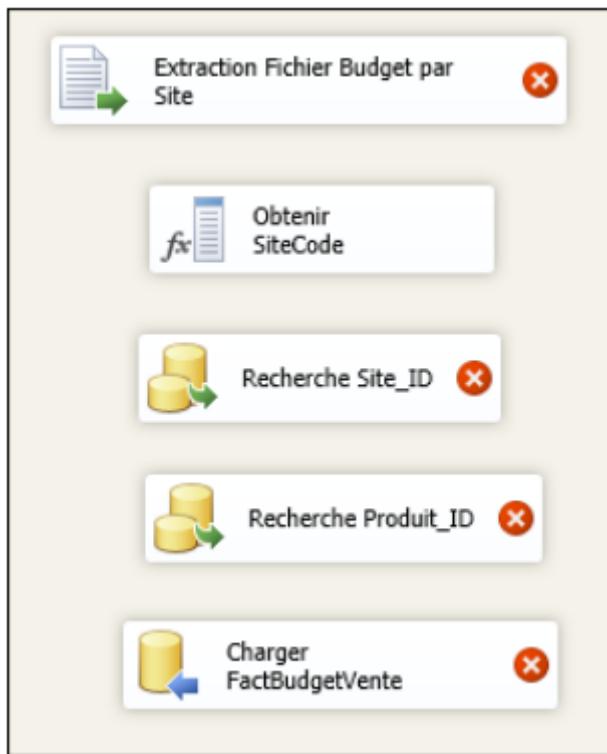
- Les tâches Sources
- Les tâches de transformation
- Les tâches Destinations



La boîte à outils de l'interface de flux de données de SSIS

En faisant glisser la tâche de flux de données, vous avez basculé l'interface en mode véritablement **ETL**. L'acronyme ETL signifie que le flux va être organisé en trois grandes phases :

- La phase **E** signifie qu'une tâche va se connecter à une source, pour en **Extraire** des lignes de données.
- La phase **T** signifie que ces lignes vont passer par des tâches de **Transformation** pour subir des tests, des validations ou des modifications.
- La phase **L** signifie que ces lignes, une fois traitées et transformées, vont être chargées (**Load** en anglais) dans la base de données destination.



Représentation schématique du déroulement d'un flux ETL

L'ensemble de ces phases va se dérouler uniquement en mémoire, d'où des gains de performance qui peuvent être substantiels par rapport au SQL, si on exploite correctement l'outil.

La barre d'outils à gauche organise les tâches disponibles dans SSIS par ces trois grandes phases ETL.

Dans la partie suivante, nous réaliserons un premier flux ETL pour comprendre le fonctionnement de SSIS.

2 - Réaliser son premier flux SSIS

2-1 - Réaliser le chargement du budget d'un seul site

Pour continuer à découvrir l'outil SSIS, nous allons réaliser le flux qui va permettre de charger les budgets de vente dans l'entrepôt de données.

Chez Distrisys, les budgets des ventes sont saisis par chaque site directement dans Excel, puis déposés dans un répertoire accessible par l'équipe informatique.

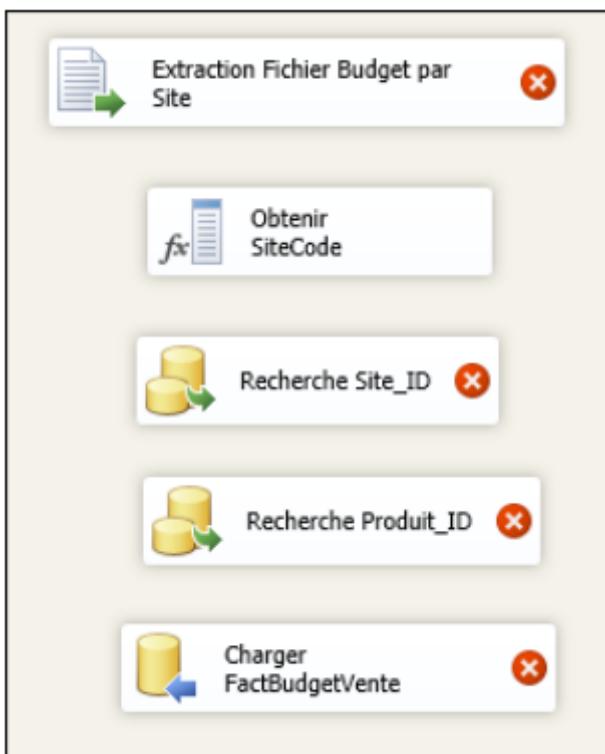
Ces fichiers budgets de ventes sont disponibles en téléchargement sur le site des Éditions ENI.

Téléchargez les fichiers et déposez-les dans un répertoire sur votre disque dur. Vous devriez alors disposer de cinq fichiers au format .csv et portant chacun le nom du code du site auquel leurs prévisions sont destinées.

ans SSIS, au niveau de l'onglet **Flux de données**, glissez cinq nouvelles tâches :

- E : **Source de fichier Plat**(classé dans **Autres sources**) à renommer **Extraction Fichier Budget par Site**.
- T : **Colonne dérivée** afin d'obtenir le **SiteCode** à renommer **Obtenir SiteCode**.
- T : **Recherche** afin d'obtenir le **Site_FK** à renommer **Recherche Site_ID**.

- T : Recherche afin d'obtenir le Produit_FK à renommer **Recherche Produit_ID**.
- L : **Destination OLE DB**(classé dans **Autres destinations**) à renommer **Charger FactBudgetVente**.

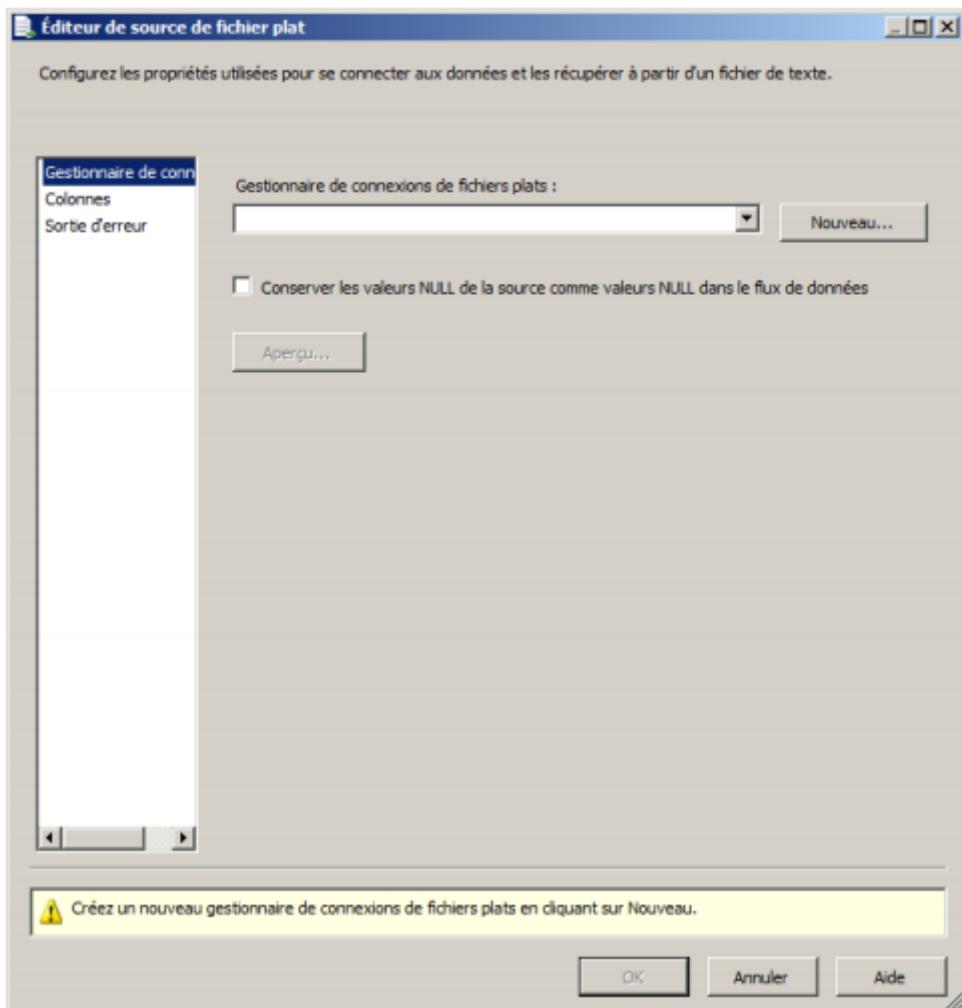


Les cinq nouvelles tâches du flux de données

- Pour renommer une tâche, cliquez dessus avec le bouton droit et sélectionnez Renommer
- Les croix rouges au niveau de chaque tâche signalent des erreurs. C'est normal à ce stade, car nous devons les configurer

Nous allons ensuite configurer chacune de ces tâches une à une :

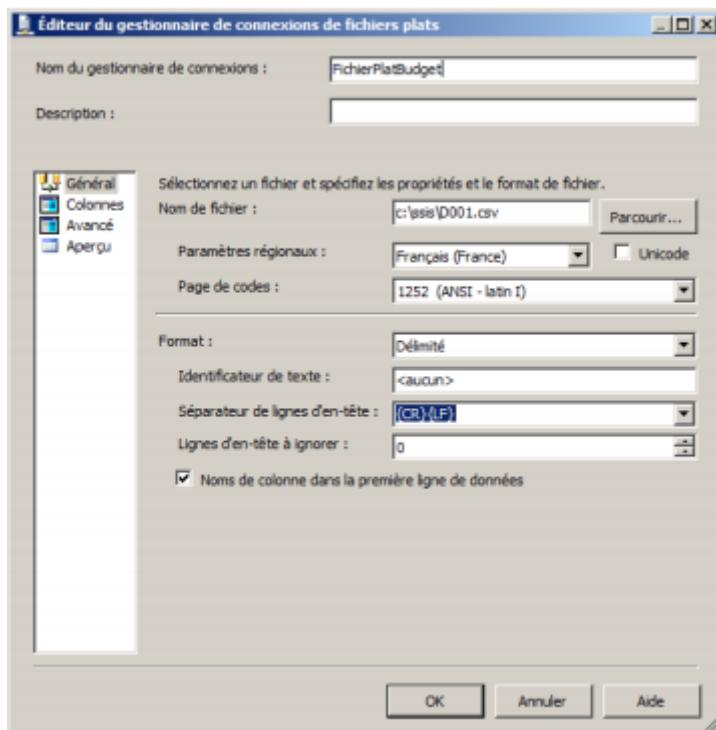
- Double cliquez sur la tâche Extraction Fichier Budget par Site afin d'entrer dans le configurateur de la tâche.
- Au niveau du Gestionnaire de connexions de fichiers plats, cliquez sur le bouton Nouveau.



Éditeur de source de fichier plat

-L'**Éditeur du gestionnaire de connexions de fichiers plats** s'ouvre. Nommez la connexion **FichierPlatBudget**.

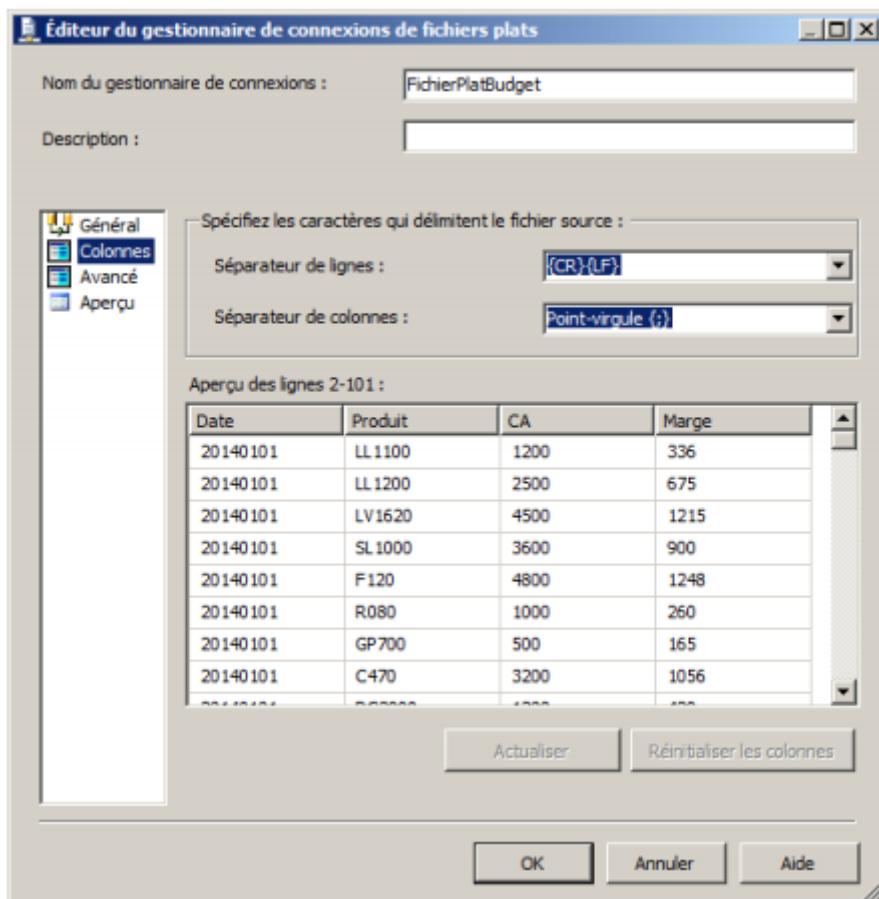
Au niveau de **Nom de fichier**, à l'aide du bouton **Parcourir**, sélectionnez le fichier source contenant une ligne d'entité, cliquez sur l'option **Noms de colonne dans la première ligne de données**. Puis continuez la configuration comme l'indique la copie d'écran ci-dessous :



Écran de configuration de la connexion à un fichier plat

Une alerte apparaît. N'y faites pas attention, car elle disparaîtra lorsque vous aurez entièrement réalisé toute la procédure qui suit.

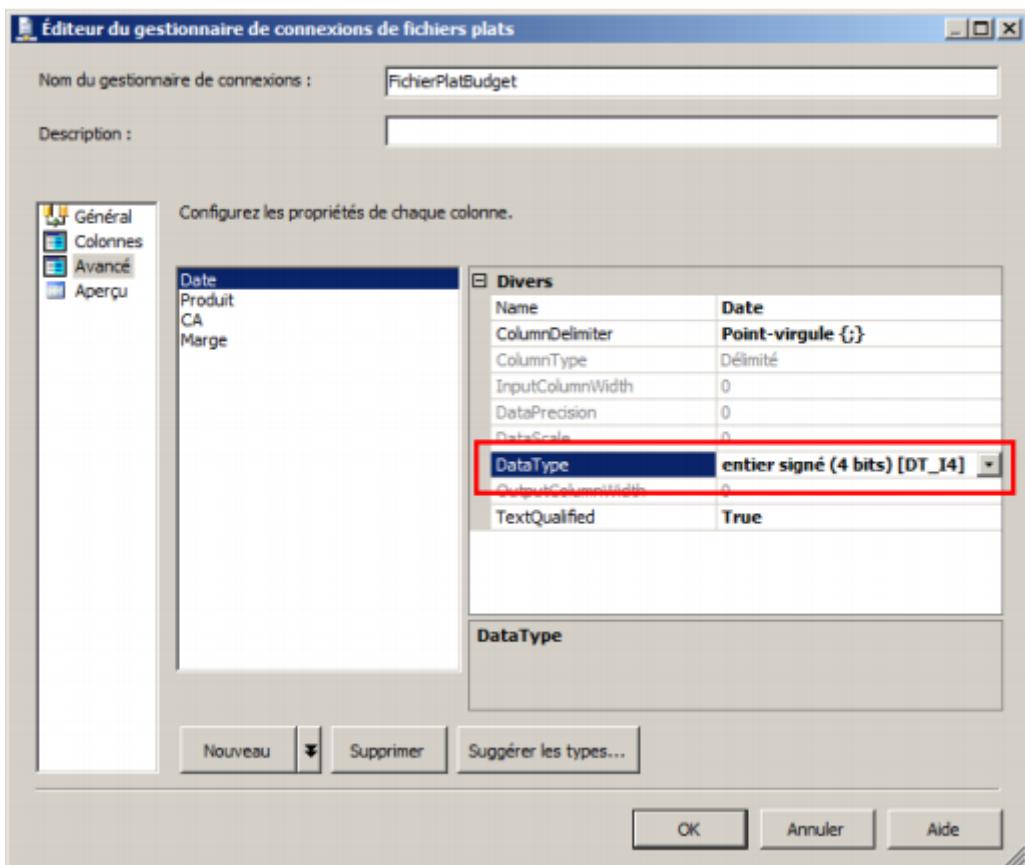
Toujours dans l'éditeur, cliquez sur l'onglet Colonnes. Au niveau du champ **Séparateur de lignes**, sélectionnez **{CR}{LF}**. Au niveau du champ **Séparateur de colonnes**, sélectionnez **Point-virgule {;}**.



Onglet Colonnes du gestionnaire de connexions

Toujours dans l'éditeur, cliquez sur l'onglet **Avancé**. Pour chaque colonne, configurez la propriété **DataType** correspondant au type de champ. Configurez les colonnes de la manière suivante :

- Date en entier signé (4bits) [DT_I4]
- Produit en chaîne [DT_STR]
- CA en entier signé (4bits) [DT_I4]
- Marge en entier signé (4bits) [DT_I4]



Configuration avancée des types des colonnes

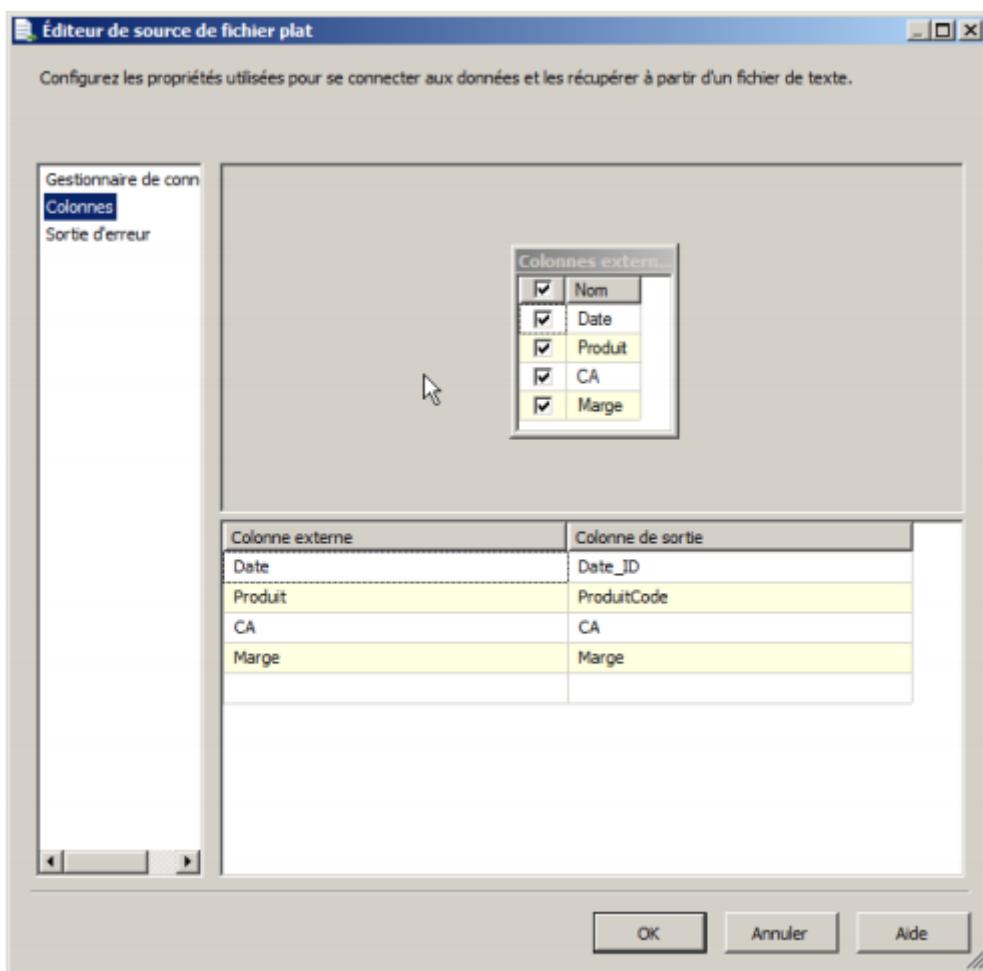
SSIS utilise des types de champs universels ne correspondant pas exactement à ceux de SQL Server. La configuration de ces types de champs est extrêmement importante dans SSIS, celui-ci y étant très sensible.

Le tableau suivant vous donne un aperçu non exhaustif des correspondances les plus courantes :

Type SQL Server	Type SSIS
Int	[DT_I4]
Numeric(9,2)	[DT_NUMERIC]
Varchar	[DT_STR] (page de code 1252)
SmallDateTime	[DT_DATE]

Terminez la configuration de la connexion fichiers plats en cliquant sur **OK**.

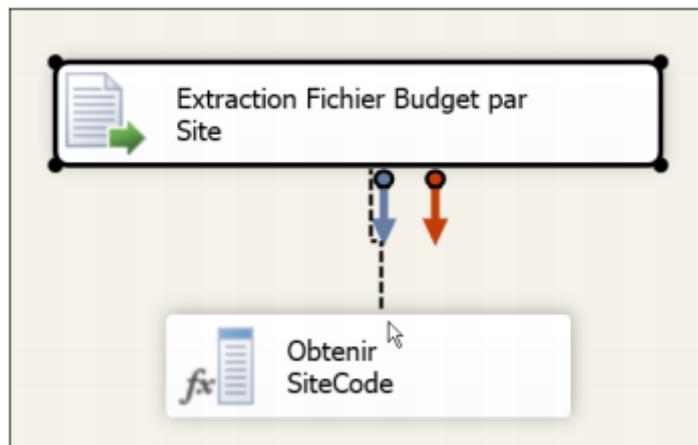
Vous revenez alors à la configuration de la tâche **Extraction Fichier Budget par Site**. Cliquez sur l'onglet **Colonnes**, sélectionnez et renommez les colonnes de sortie **Date** en **Date_ID** et **Produit** en **ProduitCode**.



Cliquez ensuite sur **OK**.

Vous venez de configurer la première tâche : Extraction Fichier Budget par Site. L'alerte d'avertissement rouge devrait disparaître

- Tirez le bout de la flèche bleue vers la tâche suivante **Obtenir SiteCode**.



Liaison entre deux tâches

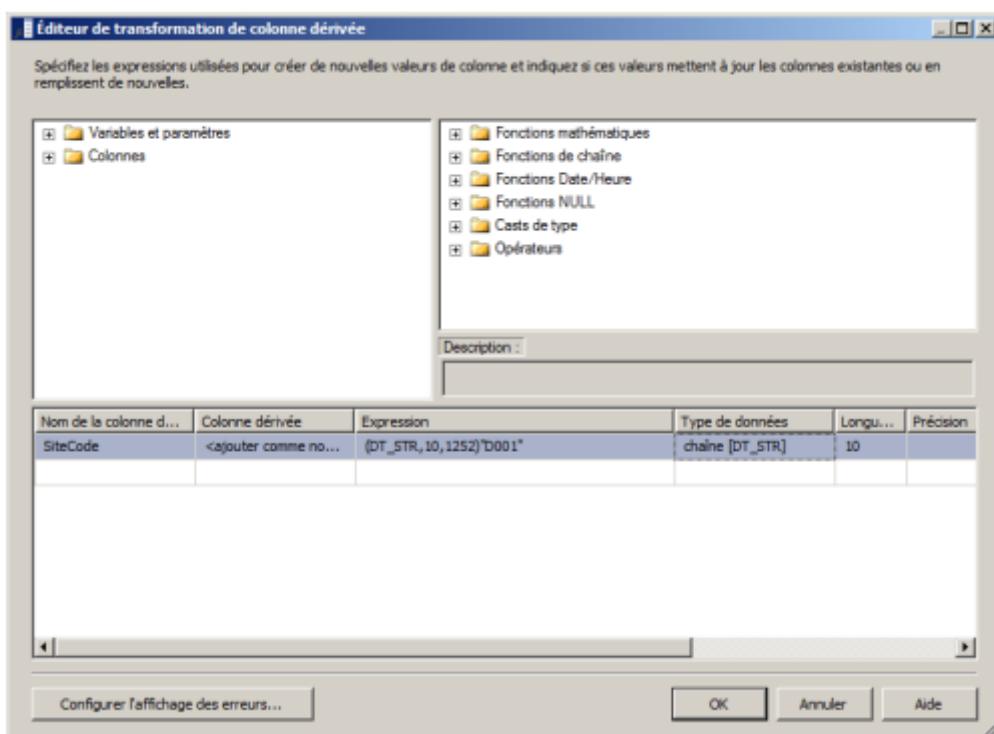
La flèche bleue (dans les flux de données, verte dans les flux de contrôle) est la flèche des succès et la flèche rouge, celle des échecs. En glissant la flèche bleue vers une autre tâche, vous indiquez où déverser les lignes de données en succès.

- Double cliquez sur la tâche **Obtenir SiteCode** pour ouvrir l'éditeur de configuration de la tâche. La colonne dérivée est une des tâches les plus courantes du Flux de données. Cette tâche permet de modifier une colonne à l'aide d'une expression ou d'ajouter une nouvelle colonne au flux. La zone en haut à gauche vous permet de sélectionner une colonne du flux courant, une variable ou un paramètre. Ces notions seront abordées plus loin dans le chapitre.

La zone en haut à droite liste l'ensemble des fonctions disponibles au niveau de cette tâche. Vous y trouverez les fonctions les plus courantes : conversion de données, fonctions mathématiques, opérateurs d'opérations et de conditions, fonctions de chaîne de caractères, de date et du traitement de la valeur Null.

Nous allons créer une règle qui génère une nouvelle colonne **SiteCode** et lui affecte manuellement la valeur "**D001**" (correspondant au SiteCode du siège social de Distrisys).

Configurez la tâche comme ci-dessous :



Vous avez dû remarquer que le type de données d'une chaîne de caractères est par défaut un type **[DT_WSTR]**, équivalent au type SQL Server nvarchar . Or, dans la table DimSite, SiteCode est de type varchar, équivalent au type **[DT_STR]** de SSIS.

Pour convertir SiteCode en **[DT_STR]**, glissez à partir des fonctions de conversion Cast de types la fonction (**DT_STR, Length, code_page**) dans Expression. En remplaçant **Length** par une valeur, vous spécifiez la longueur de la chaîne de caractères.

En remplaçant **Code_page** par 1252 , vous spécifiez la page de code 1252 (ANSI-Latin I).

Au final, vous devriez avoir dans la colonne **Expression** :

`(DT_STR,10,1252)"D001"`

En sortie de l'éditeur de tâche, l'alerte d'avertissement rouge disparaît. Tirez le bout de la flèche bleue vers la tâche suivante **Recherche Site ID**

Maintenant, nous allons nous atteler à configurer la prochaine tâche : la tâche de **Recherche**.

La tâche de Recherche (ou Lookup en anglais) est une tâche essentielle et très caractéristique des processus ETL. Cette tâche va établir une correspondance entre un ou plusieurs champs du flux courant avec des champs d'une table de référence. En sortie, nous pourrons en déduire un ou plusieurs champs de cette même table de référence.

Pour configurer les dernières tâches de Recherche et de chargement de données, nous avons besoin de créer une connexion à DistrisysDW.

Pour cela, faites un clic droit dans la zone du **Gestionnaire de connexions**, situé en bas de l'écran. Sélectionnez Nouvelle **connexion OLE DB**.

Cliquez sur **Nouveau**. Configurez la connexion pour vous connecter à votre entrepôt de données, comme le montre la copie d'écran ci-dessous. Au niveau du champ **Nom du serveur** spécifiez le nom de votre instance SQL Server et dans Sélectionner ou entrer un nom de base de données, sélectionnez dans le menu déroulant **DistrisysDW** :

Cliquez sur **OK**.

- Renommez cette nouvelle connexion DistrisysDW.

Cette connexion est définie pour le package courant. Mais nous aurons besoin dans d'autres packages du projet d'une même connexion sur la base DistrisysDW. SSIS permet de définir des connexions au niveau du projet, afin qu'elles soient disponibles dans tous les packages du projet.

Cliquez avec le bouton droit sur la connexion **DistrisysDW**, et sélectionnez **Convertir en connexion de projet**.

La connexion apparaît maintenant dans l'Explorateur de solutions et elle est disponible pour tous les packages du projet.

Dans le Gestionnaire de connexions, le nom est maintenant préfixé par (**projet**).

Cette phase préparatoire terminée, nous allons pouvoir poursuivre la réalisation de notre flux.

Double cliquez maintenant sur la tâche **Recherche Site ID** pour afficher l'**éditeur de Transformation de Recherche**.

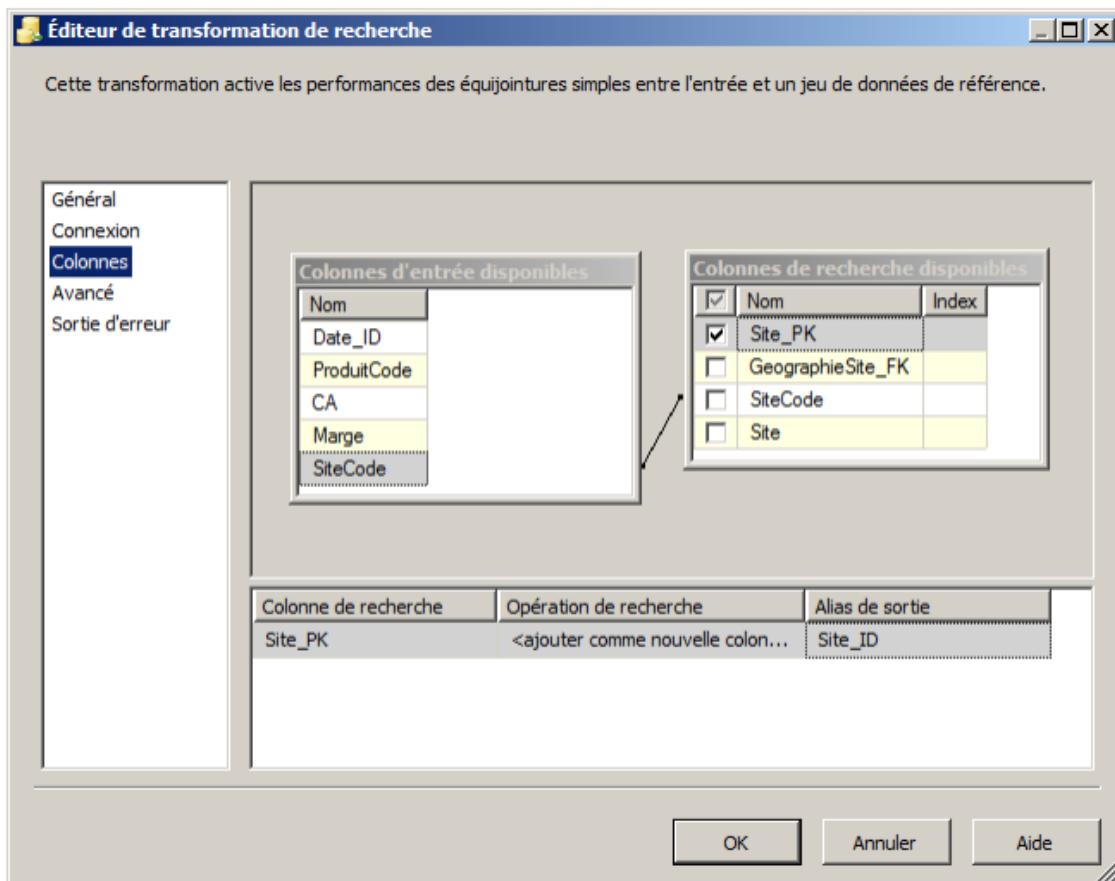
Dans l'exemple suivant, nous allons établir la correspondance entre le **SiteCode** du flux courant avec le **DimSite** de la table DimProduit. En sortie, nous pourrons récupérer l'identifiant technique **Site_PK**.

Pour cela, cliquez sur l'onglet **Connexion**.

Sélectionnez **DistrisysDW** comme **Gestionnaire de connexions OLE DB**.

Ensuite, spécifiez pour le champ **Utiliser une table ou une vue**, la table **DimSite**.

Puis cliquez sur l'onglet **Colonnes**. Mappez **SiteCode** des Colonnes d'entrées disponibles avec le champ **SiteCode** des **Colonnes de recherche disponibles** (table de référence). Pour cela, faites un clic droit sur **SiteCode** de **Colonnes** d'entrée disponibles et sélectionnez **modifier les mappages**.



Ensuite, cliquez sur le champ **Site_PK** de la table de référence pour ajouter cette colonne au flux de données. Renommez ce champ **Site_ID** au niveau de la colonne **Alias de sortie**. Puis cliquez sur **OK** pour sortir et valider les modifications effectuées dans l'éditeur de tâche.

L'alerte d'avertissement rouge disparaît. Tirez alors le bout de la flèche bleue vers la tâche suivante Recherche Produit ID . Sélectionnez la sortie avec correspondance .

Comme réalisé précédemment, configurez la tâche Recherche Produit ID . Pour cela, sélectionnez DimProduit en table de référence. Puis faites le lien entre ProduitCode du flux d'entrée avec ProduitCode de DimProduit et cliquez sur Produit_PK. Renommez la colonne en sortie Produit_ID.

Tirez le bout de la flèche bleue vers la tâche suivante **Charger FactBudgetVente**. Sélectionnez la sortie **avec correspondance**.

Éditez maintenant la tâche Charger **FactBudgetVente**.

Cette tâche a pour objectif de réaliser l'insertion des lignes dans la table FactBudgetVente de l'entrepôt de données.

Puis continuez la configuration de la tâche comme ci-dessous :

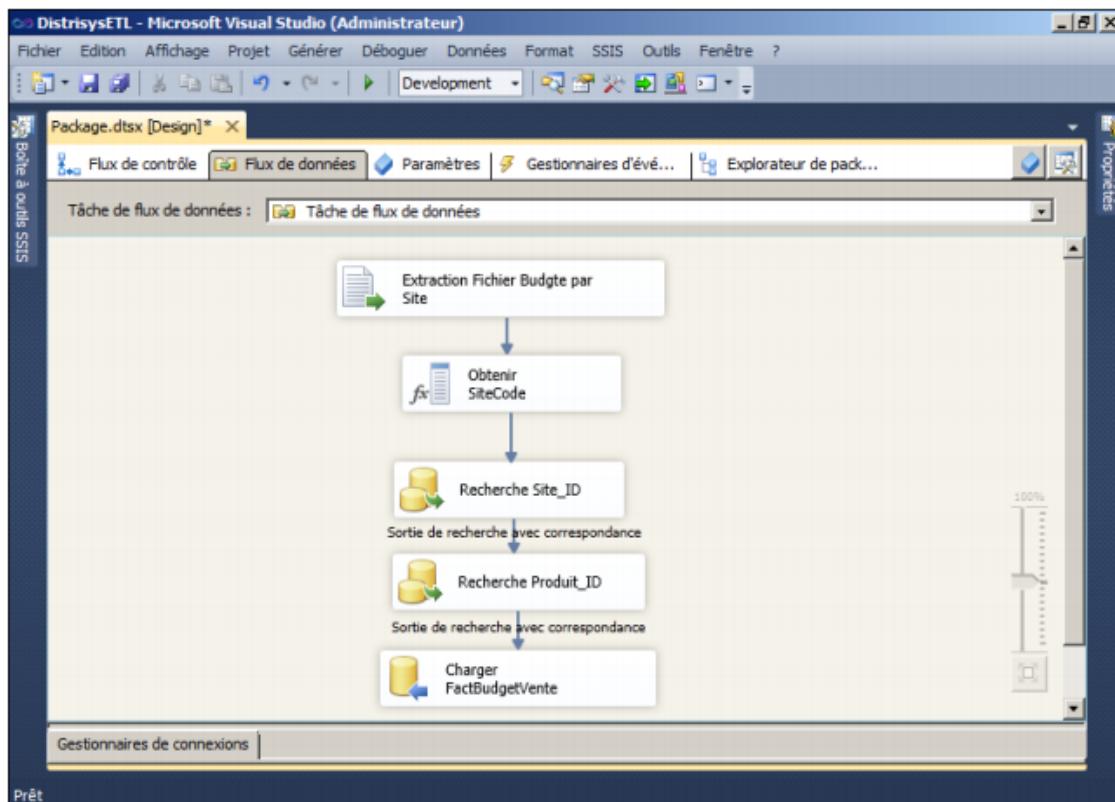
Un message d'erreur apparaît. N'y faites pas attention, car il disparaîtra lorsque vous aurez entièrement réalisé toute la procédure qui suit.

En ce qui concerne le chargement d'une table de faits, le mode d'accès aux données doit toujours être en **chargement rapide**. De même, pour obtenir de meilleures performances, l'option Vérifier les contraintes doit être décochée. Normalement, lors du chargement d'une table de faits, les tâches de type Recherche devraient vous assurer de l'existence des clés techniques pour chaque identifiant de liaison aux tables de dimension.

Pour finir, dans l'onglet **Mappages**, réalisez les correspondances suivantes :

- Date_ID avec DateBudget_FK.
- Produit_ID avec Produit_FK.
- Site_ID avec Site_FK.

Il n'y a plus d'alerte rouge au niveau des tâches du flux. Au final, vous devriez obtenir le flux de données ci-dessous :



Dans l'**Explorateur de solutions**, renommez votre package **DW_BudgetVente.dtsx**.

Pour exécuter le flux, faites un clic droit n'importe où dans la zone de travail et cliquez sur **Exécuter la tâche**.

Si tout se passe bien, une marque verte apparaît dans le coin supérieur droit de la tâche et le nombre de lignes transférées à chaque étape s'affiche. Dans notre cas, 120 lignes apparaissent.

Pour sortir du mode exécution, cliquez sur le bouton Arrêter le mode débogage dans la barre d'outils de débogage.

Un simple traitement du cube vous permet de rendre ces données disponibles.

Dans SSMS, connectez-vous à Analysis Services. Traitez le cube et réalisez un tableau croisé dynamique avec les données nouvellement insérées.

Vous venez de réaliser votre premier flux ETL avec SSIS. Dans la prochaine partie, nous allons charger l'ensemble des budgets pour illustrer l'utilisation de l'onglet **Flux de contrôle**.

2.2 - Charger les données de budget à partir de plusieurs fichiers Excel

Dans cette partie, nous allons compléter le flux précédent afin d'illustrer l'utilisation de l'onglet et des tâches de flux de contrôle, et ainsi bien différencier l'utilisation de ces deux onglets.

Tout d'abord, afin de pouvoir lancer le flux plusieurs fois lors du développement, nous allons ajouter une tâche de flux de contrôle qui efface, avant chaque exécution, toutes les lignes de l'année 2014 de la table FactBudgetVente.

- Pour cela, dans SSIS, allez sur l'onglet **Flux de contrôle** et glissez la tâche d'**Exécution de requêtes SQL**. Renommez la tâche **Efface les données de budget de 2014**.

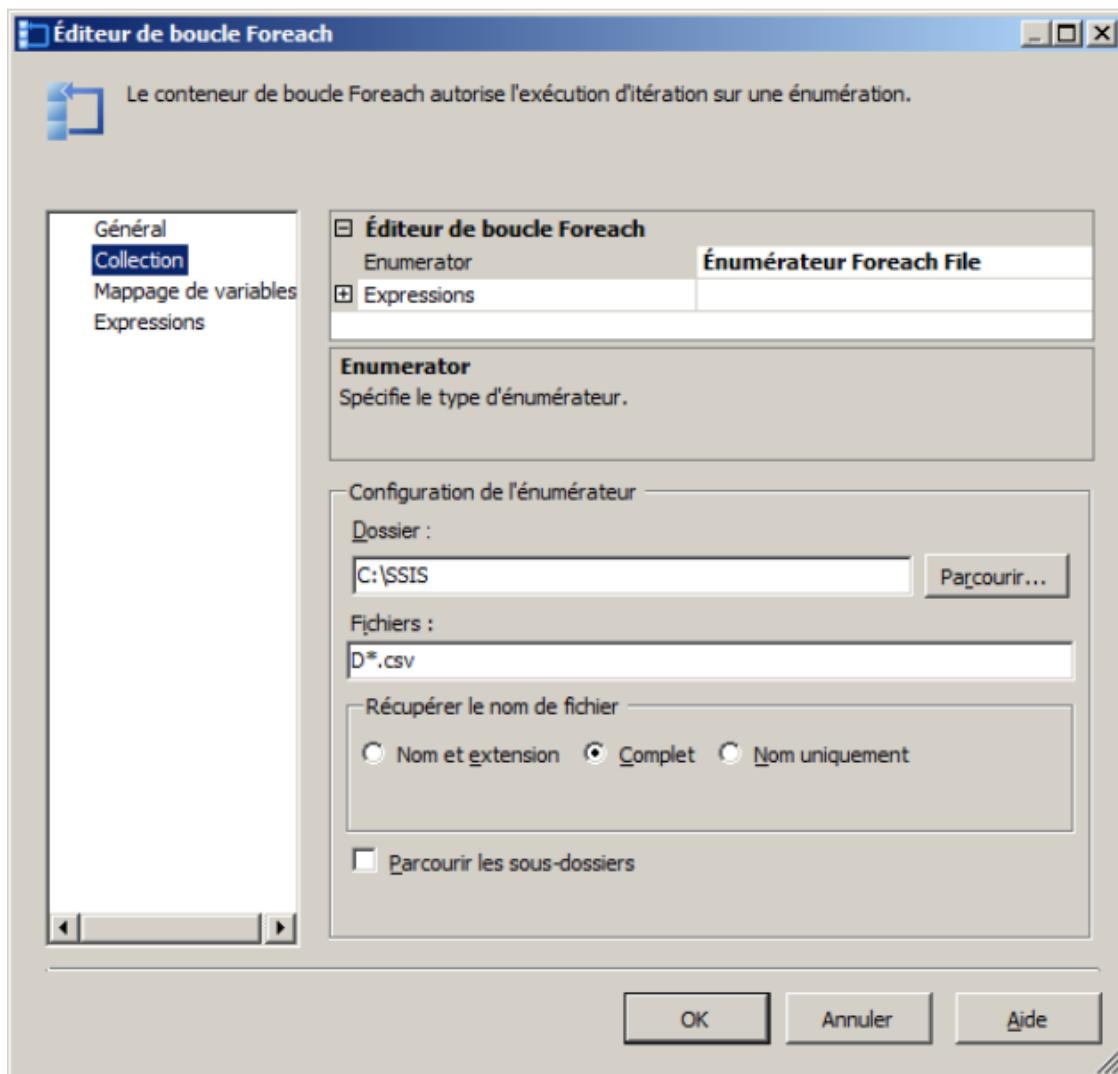
- Éditez la tâche. Au niveau de la propriété **Connection**, sélectionnez **DistrisysDW**. Au niveau de la propriété **SQL Statement**, tapez la requête suivante :

```
DELETE FROM FactBudgetVente WHERE DateBudget_FK
```

Tirez la flèche verte vers la tâche de **flux de données** que vous renommerez : Chargement du budget d'un Site.

- Exécutez le flux en faisant un clic droit sur le package au niveau de l'Explorateur de solutions. Actuellement, le flux ne charge que le budget du site D001 (Siège social). La finalité serait que le flux parcourt le répertoire, où sont déposés les fichiers de budget, puis de charger ces fichiers les uns après les autres. Pour réaliser cela, nous allons utiliser la tâche **Conteneur de boucle Foreach**.

- Glissez la tâche **Conteneur de boucle Foreach**, puis renommez-la : **Lire les fichiers Site dans un répertoire**.
- Supprimez le lien entre les tâches **Efface les données de budget de 2014** et **Chargement du budget d'un Site** .
- Glissez la tâche **Chargement du budget d'un Site** dans le périmètre du For Each.
- Entrez dans le configIBUTEUR de la tâche de conteneur de boucle For Each.
- Configurez les propriétés de l'onglet **Collection** :
 - Énumérateur : **Enumérateur Foreach File**.
 - Dossier : spécifiez le répertoire où sont stockés vos fichiers CSV.
 - Fichiers : **D*.csv** , afin de récupérer uniquement les fichiers de site au format .csv.
 - Récupérez le nom de fichier : **Complet**, afin de récupérer le chemin complet d'accès aux fichiers.

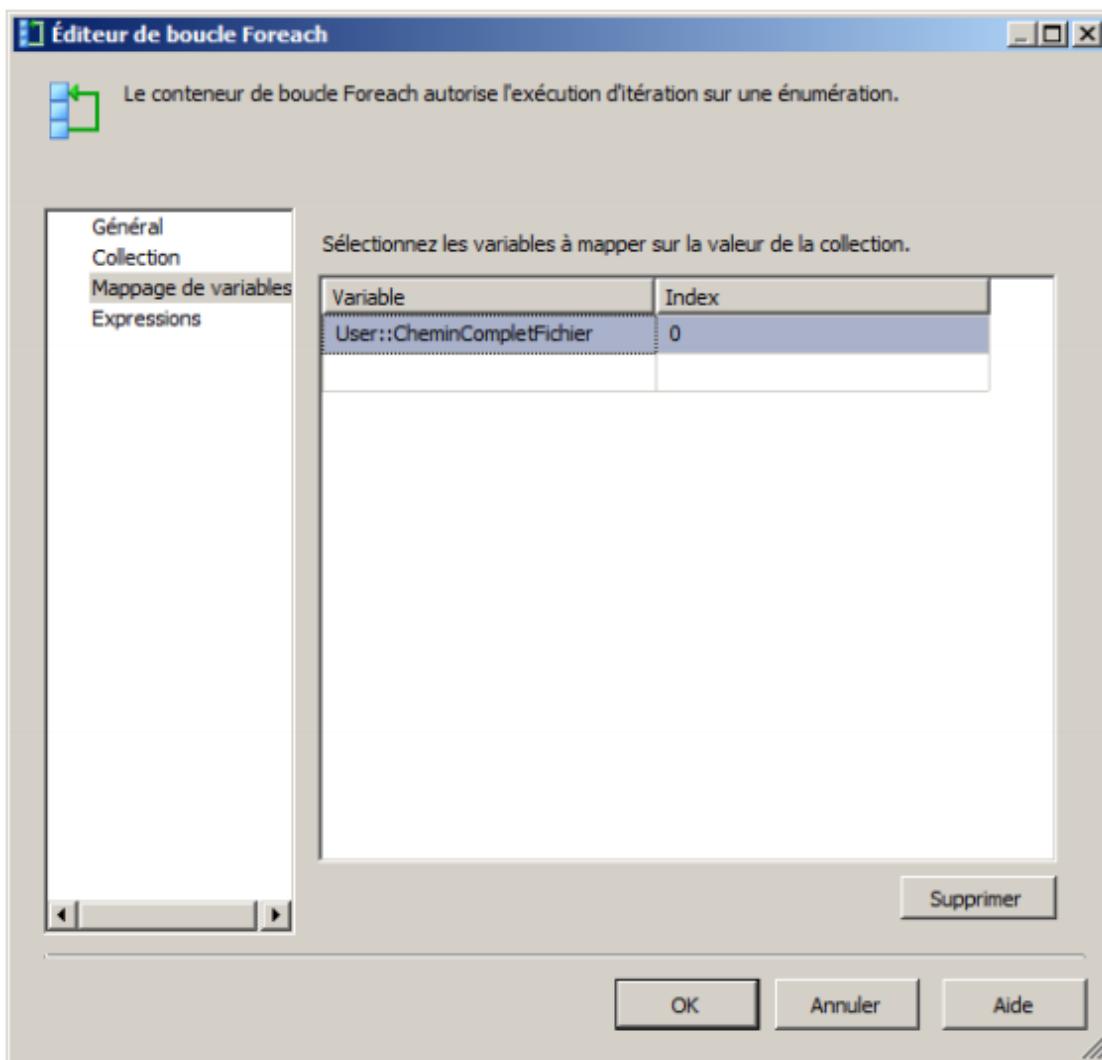


Configuration de la boucle Foreach dans le cas du parcours de fichiers dans un répertoire

Dans l'onglet Mappage de variables, créez une nouvelle variable CheminCompletFichier avec les propriétés suivantes :

- Conteneur : **DWBudgetVente**
- Nom : **CheminCompletFichier**
- Espace de noms : **User**
- Type de valeur : **string**
- Valeur : c:\SSIS\D001.csv (spécifiez le chemin complet de votre fichier csv afin d'initialiser le contenu de la variable).

- Puis mappez cette variable sur l'index 0, comme ci-dessous :



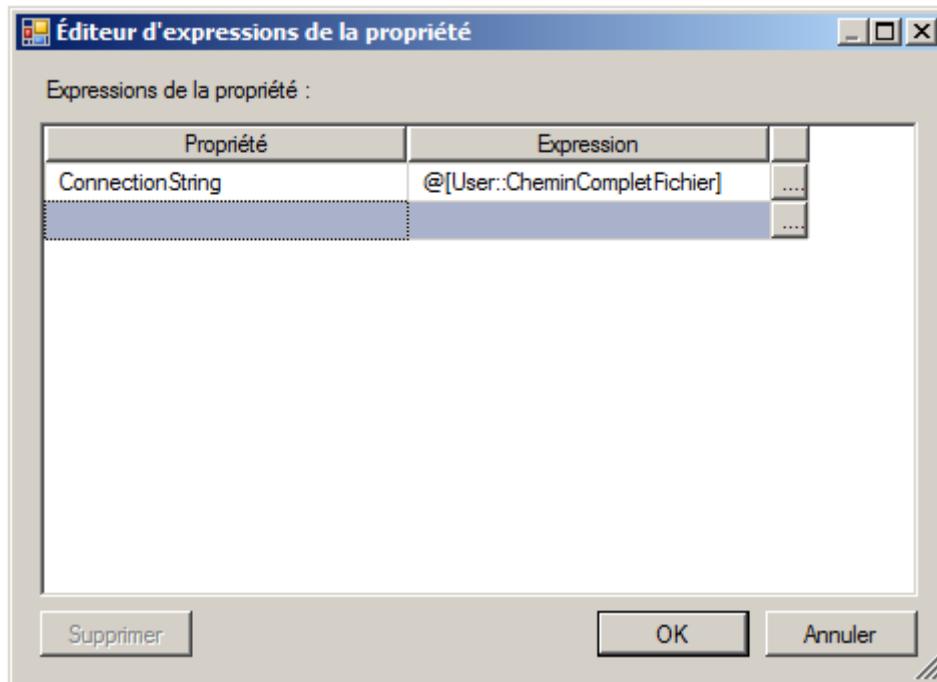
À chaque itération de la boucle, la variable **User::CheminCompletFichier** va prendre la valeur du chemin d'accès au fichier de budget (nom du fichier avec extension incluse).

Nous allons donc dynamiser la valeur du chemin d'accès de la connexion FichierPlatBudget à l'aide de cette nouvelle variable.

Pour cela, dans le Gestionnaire de connexions en bas de la zone de travail de SSIS, faites un clic droit sur FichierPlatBudget afin d'afficher les propriétés de la connexion.

Dans les propriétés, cherchez la propriété Expressions puis cliquez sur le bouton pour afficher l'éditeur.

Dans l'**Éditeur d'expressions de la propriété**, affectez la variable **User::CheminCompletFichier** à la propriété **ConnectionString**, comme ci-dessous :



Ainsi la propriété **ConnectionString** de la connexion **FichierPlatBudget** va prendre la valeur de la variable User::CheminCompletFichier à chaque itération de la boucle.

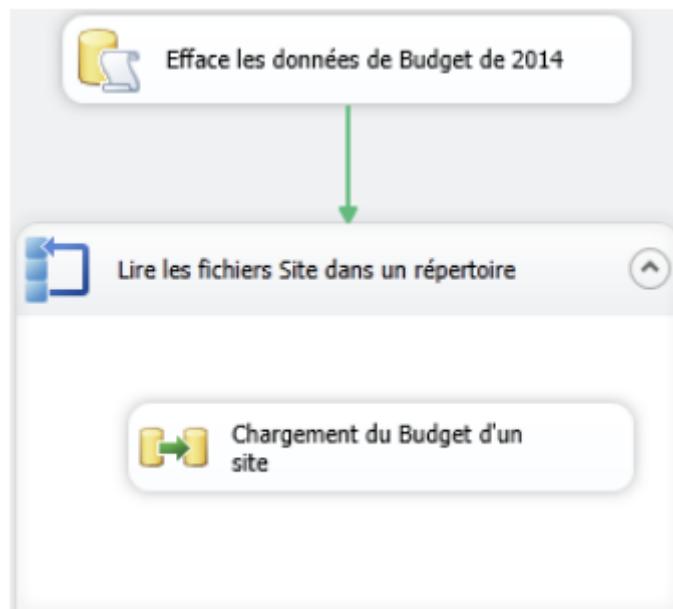
- Pour afficher la liste des variables et pour pouvoir les modifier ou en créer d'autres, cliquez dans la barre de menu sur **SSIS>>variables**.
- Dans le Gestionnaire de connexions, notez maintenant le signe à côté du nom qui signale l'utilisation d'une expression.

Pour finir, nous devons récupérer le SiteCode contenu dans le nom du fichier, et donc dans le nom de la variable.

Basculez dans l'onglet **Flux de données** et modifiez la tâche **Obtenir SiteCode**, pour affecter à la colonne **SiteCode** l'expression suivante :

```
(DT_STR,10,1252)SUBSTRING(RIGHT(@[User::CheminCompletFichier],8),1,4)
```

Le flux de contrôle au final devrait ressembler à ceci :



Exécutez le flux pour vérifier que tout fonctionne correctement. La tâche de flux de chargement du budget va s'exécuter autant de fois qu'il y aura de fichiers csv de budget.

Cet exemple concret de l'utilisation de SSIS vous a permis de comprendre la différence d'utilisation des deux facettes de SSIS. En utilisation décisionnelle :

- Le flux de contrôle permet de piloter l'exécution d'un flux de données et doit, autant que possible, ne pas avoir d'influence directe sur les données.
- Le flux de données réalise l'extraction, le traitement et le chargement. Il n'a d'influence que sur les données elles-mêmes.

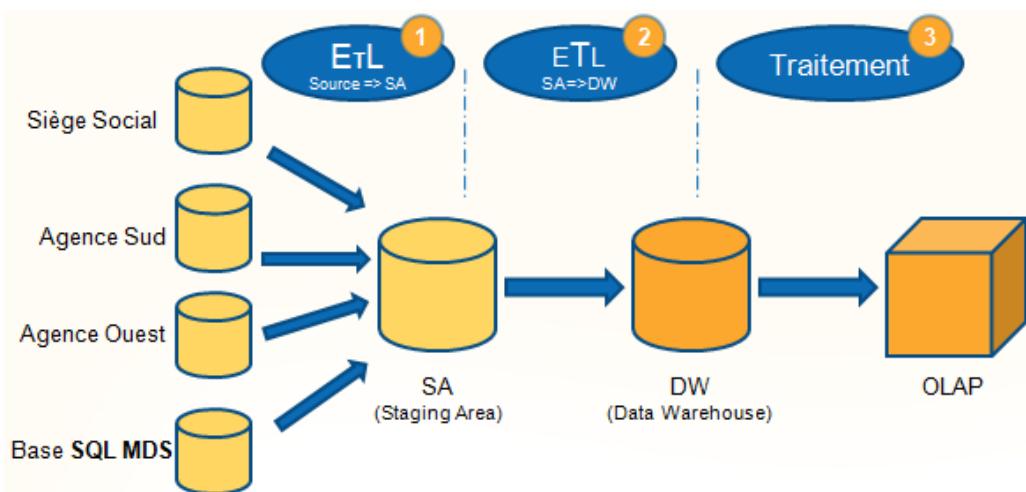
3 - Développer des flux ETL pour le décisionnel

3-1 - Déroulement de l'exécution d'un processus ETL

Dans cette partie, nous allons vous présenter les différents types de flux, que vous serez en mesure de rencontrer pour alimenter votre entrepôt de données.

Dans les faits, les données ne vont pas transiter directement des systèmes sources vers l'entrepôt de données. Les données vont transiter par au moins un palier : le sas de données. Dans notre cas, nous appellerons cette base DistrisysSA (SA en anglais signifiant Staging Area).

Le chargement va se faire suivant ce schéma de principe :



Architecture de chargement de données

La base SQL MDS fait référence aux bases de données de référentiel du produit SQL Server Master Data Services. Nous présenterons cet outil dans le chapitre suivant Gérer les données de référence avec Master Data Services.

Le SA a plusieurs rôles :

- Rapatrier les informations émanant de sources multiples, en garantissant qu'il n'y ait pas de pertes de données lors de ce processus.

La base SQL MDS fait référence aux bases de données de référentiel du produit SQL Server Master Data Services. Nous présenterons cet outil dans le chapitre suivant Gérer les données de référence avec Master Data Services.

Le SA a plusieurs rôles :

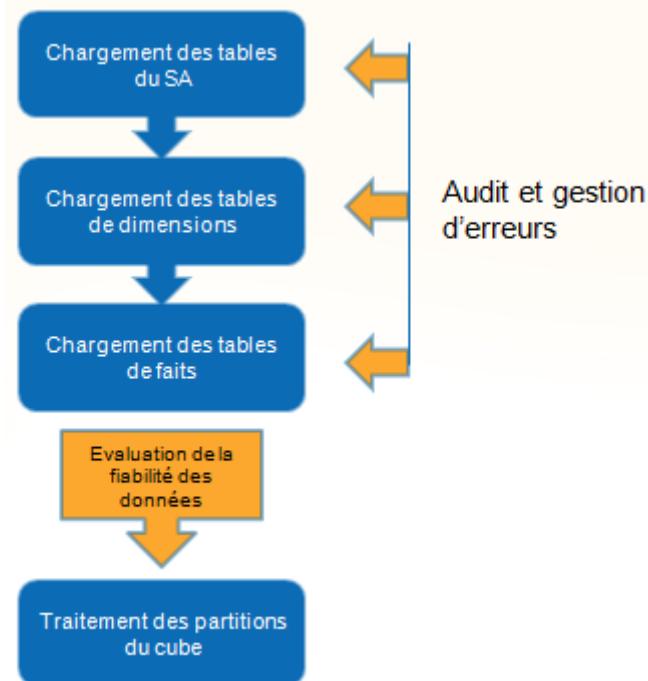
- Rapatrier les informations émanant de sources multiples, en garantissant qu'il n'y ait pas de pertes de données lors de ce processus
- Faire une zone mémoire tampon d'un état brut de la source à un instant passé et ainsi, faciliter la mise en œuvre d'un processus de reprise de données, que nous verrons dans ce chapitre à la section L'audit des flux ETL.

La mise en place d'un SA est une étape indispensable à la bonne mise en œuvre de vos flux ETL.

Nous répartirons les rôles de la manière suivante :

- Les flux entre les systèmes sources et le SA seront des flux de copie de données (EL). Nous éviterons donc, dans le SA, toute contrainte d'intégrité, et dans les flux, toute règle de gestion et autre requête avec jointure interne, qui peut provoquer une déperdition de données sources. Les tables du SA ne sont pas soumises à une modélisation. Le SA est simplement à but pratique afin de simplifier la seconde étape.
- Les flux entre le SA et le DW seront de véritables flux ETL. Nous utiliserons alors pleinement l'onglet **Flux de données** de SSIS ainsi que les tâches de transformation. C'est à cette étape-ci, que nous réaliserons un audit précis de nos flux.

Le déroulement du flux décisionnel va donc se dérouler ainsi :



Déroulement de l'exécution des flux décisionnels

En décisionnel, il existe donc trois sortes de flux différents :

- Les flux de copie des données sources vers le SA.
- Les flux de gestion et de mise à jour des dimensions du DW.
- Les flux de chargement de s tables de faits du DW.

Dans les parties qui vont suivre, nous allons illustrer la réalisation de chacun de ces flux.

3-2 - Réaliser un flux pour charger le sas de données

Lors de cette partie, nous allons étudier un flux permettant de copier des données de facturation vers une base de données DistrisysSA.

Dans notre étude de cas, Distrisys dispose d'un système de gestion par site. Pour réaliser une copie complète des données, notre flux devra donc se connecter successivement à chacun de ces sites.

Afin de réaliser, de suivre et d'exécuter le flux présenté dans ce chapitre, téléchargez sur le site des Éditions ENI les éléments suivants :

Les fichiers de sauvegarde de base de données :

- DistrisysERP_SiegeSocial.bak
- DistrisysERP_AgenceSud.bak
- DistrisysERP_AgenceOuest.bak
- DistrisysSA.bak

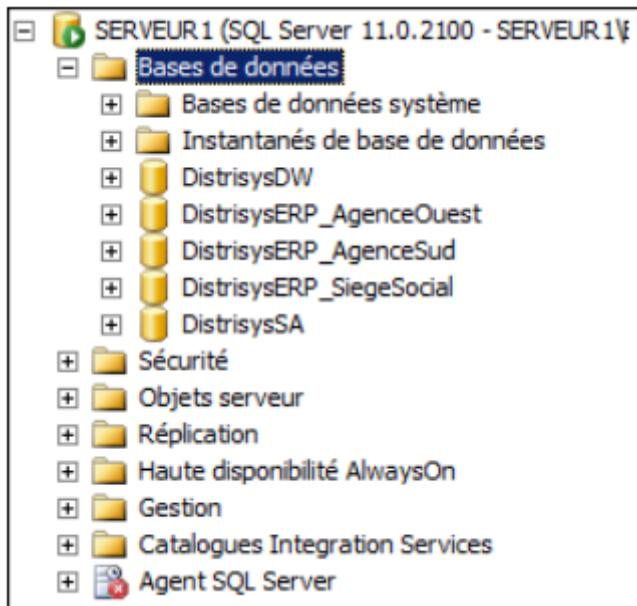
La solution SSIS :

- Répertoire DistrisysETL

Restaurez les trois bases de données.

Vous pouvez vous reporter au chapitre Installation et découverte des outils SQL Server -Restauration d'une base de données pour les procédures de restauration d'une base SQL Server.

Dans SSMS, vous devriez alors avoir les cinq bases de données suivantes :



La base de données DistrisysSA contient les trois tables suivantes :

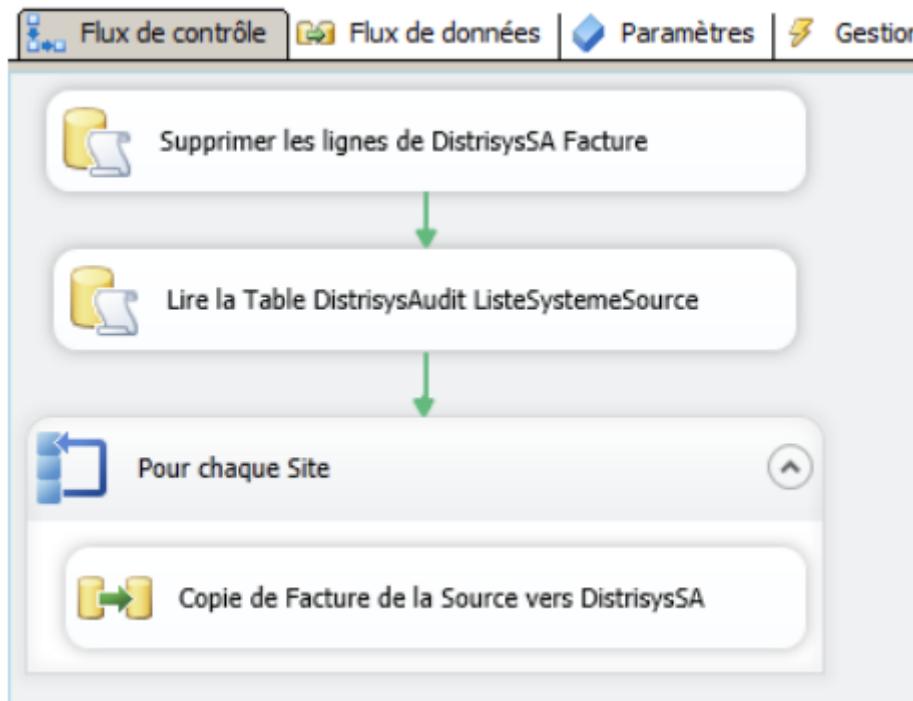
- **Facture** : le contenu des tables sources, concernant les données de facturation, sera copié dans cette table.
- **Produit** : le contenu des tables sources, contenant les données concernant les produits, sera copié dans cette table.
- **ListeSystemeSource** : cette table liste les sites auxquels nous souhaitons nous connecter, ainsi que les chaînes de connexion de chacune des bases de données de ces sites.

Par défaut, les chaînes de connexion existantes font référence à un serveur et une instance SQL Server. Pensez à remplacer ces valeurs par le nom de votre instance SQL Server.

Nous allons maintenant ouvrir le package contenant le flux à étudier.

Dans SSIS, ouvrez la solution DistrisysETL précédemment téléchargée et ouvrez le package **SA_Facture.dtsx**.

L'onglet **Flux de contrôle** du package SA_Facture.dtsx se présente ainsi :



Flux de contrôle du flux SA_Facture.dtsx

Ce package dispose des variables suivantes :

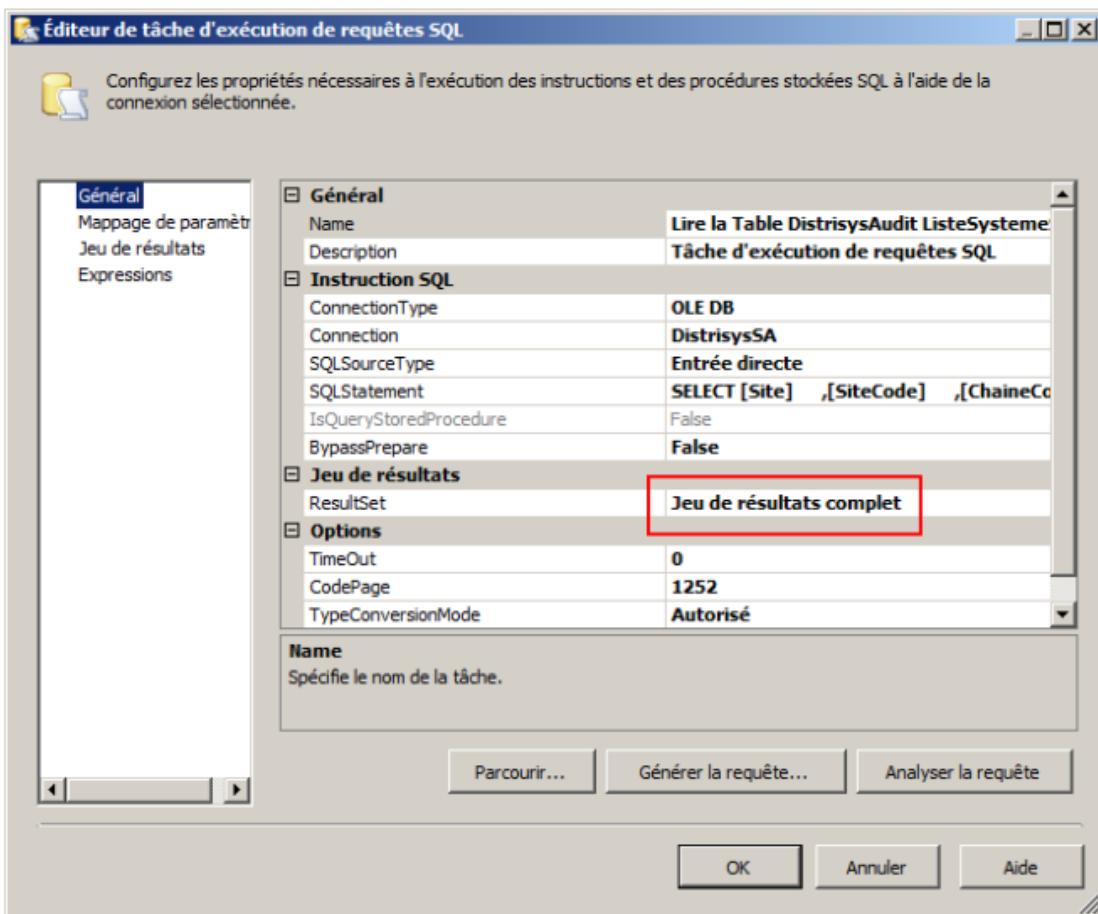
Variables				
Nom	Étendue	Type de donn...	Valeur	
ChaineConnexionSource	SA_Facture	String	Data Source=serveur1;Initial Catalog=SA_Facture;Provider=SQLNCLI10.1;Integrated Security=SSPI;	
ListeSystemeSource	SA_Facture	Object	System.Object	
SiteCode	SA_Facture	String	D001	
SiteNom	SA_Facture	String	Siège Social	

Liste des variables du flux SA_Facture.dtsx

Le flux fonctionne ainsi :

- La première tâche réinitialise la table Facture de la base de données DistrisysSA.
- La seconde tâche, **Lire la Table DistrisysAuditListeSystemeSource**, affecte le contenu de la table **ListeSystemeSource** dans la variable de type Objet du même nom

Pour faire cela, la tâche Exécution SQL est configurée ainsi :



La requête exécutée est la suivante :

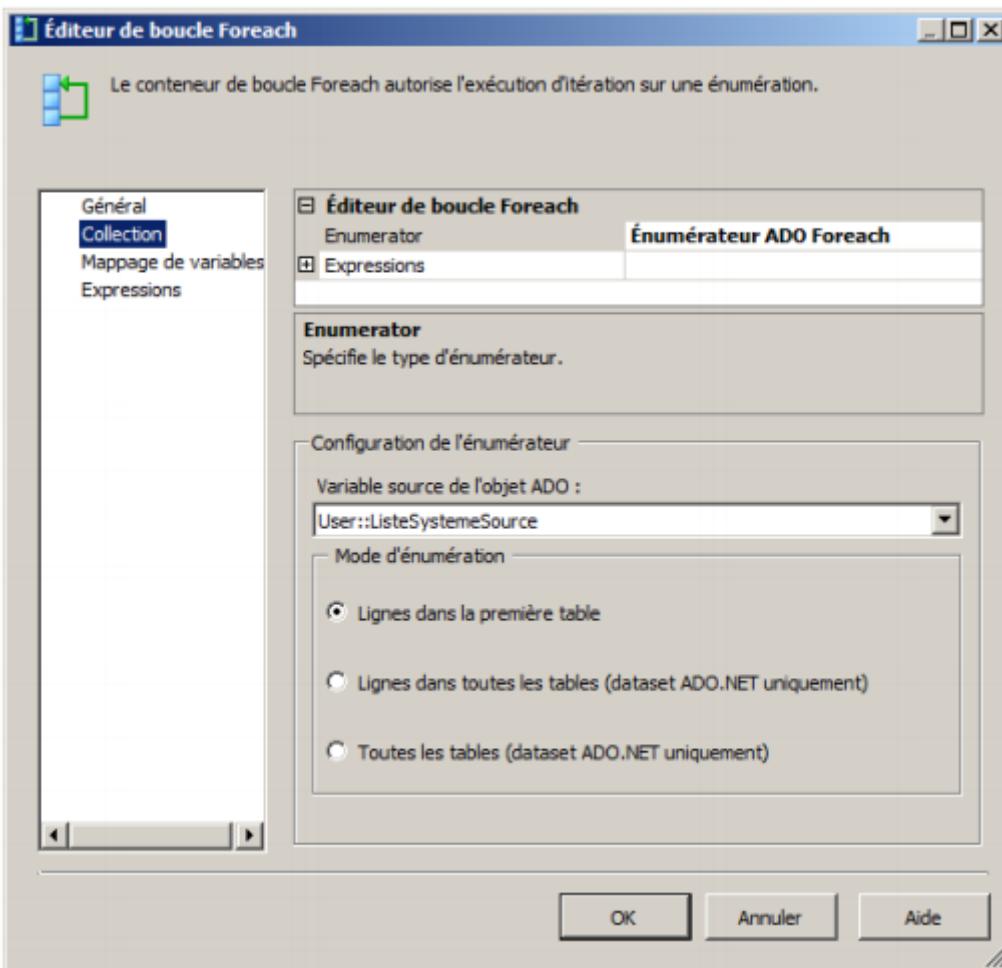
```
SELECT [Site],[SiteCode],[ConnexionSource] FROM
    [ListeSystemeSource] Where Valide='O'
```

La section **Jeu de résultats** est configurée ainsi :



3. La troisième tâche **Pour chaque Site**, de type **Conteneur de boucle Foreach**, parcourt la variable récupérée précédemment, pour lancer à chaque itération le flux de données Copie de Facture de la source.

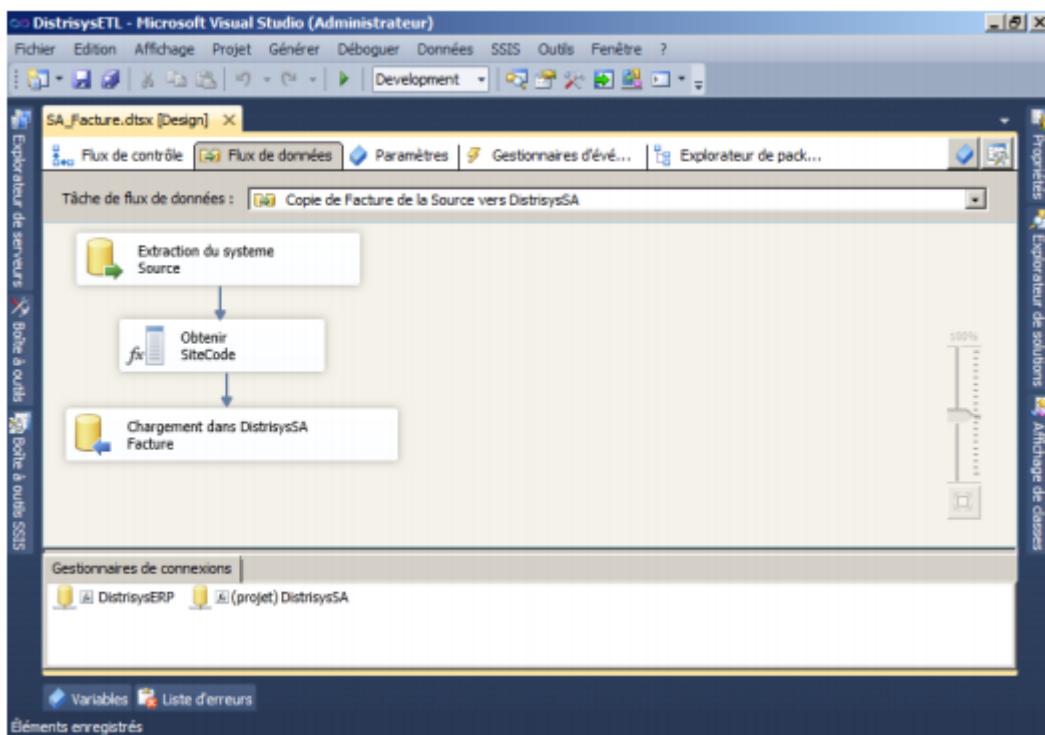
La tâche **Pour chaque Site** est configurée ainsi :



À chaque itération, les variables **User::SiteNom**, **User::SiteCode** et **User::ChaineConnexionSource** viennent récupérer les valeurs des champs de la ligne courante, parcourues par la boucle. Pour cela, la section **Mappage de variables** est configurée ainsi :

La colonne **Index** correspond à l'ordre des colonnes dans la requête de la tâche **Lire la Table DistrisysAuditListeSystemeSource** définie précédemment.

4. Enfin, la tâche de flux de données effectue la copie des données de facturation du système source courant vers la table Facture de DistrisySSA.



Le flux de données est très simple, on évite autant que possible toute modification de données. La colonne dérivée **Obtenir SiteCode** ajoute simplement au flux le contenu de la variable **User::SiteCode** identifiant le site courant.

La connexion à la source de données se fait par la configuration de la connexion DistrisysERP. À chaque itération, DistrisysERP récupère la valeur de la chaîne de connexion de la variable **User::ChaineConnexionSource**. Cette configuration est du même ordre que celui réalisé dans le flux de chargement du budget.

Pour que le flux fonctionne, modifiez les chaînes de connexions spécifiées dans la table ListeSystemeSource, ainsi que celles de DistrisysSA, pour les adapter à votre environnement serveur.

Exécutez le flux pour suivre et observer le comportement du package. Ce flux n'est qu'une illustration de flux de récupération de données. Ce type de flux peut prendre des formes assez diverses. À l'opposé, les flux de chargement des dimensions et des tables de faits sont des flux très stéréotypés. Leurs formes sont assez transposables d'une table à une autre, et d'un système à l'autre.

3-3 - Réaliser un flux pour charger une dimension

3-3-1 - Cas d'une dimension standard

Nous allons à présent étudier un flux de chargement et de mise à jour de la table de dimension Produit. Sauf exception, tous les flux de chargement et de mise à jour des dimensions sont réalisés sur le modèle standard qui va suivre ou sur celui proposé dans la partie suivante. Nous considérerons que le flux qui charge la table Produit, à partir des systèmes sources a été réalisé. Nous disposons alors des données Produit courantes dans la table Produit de la base de données DistrisysSA.

La société Distrisys prévoit de mettre en place un référentiel Produit. Cela sera effectué au chapitre suivant Gérer les données de référence avec Master Data Services. Si ce travail de référentiel avait été préalablement réalisé, cette table Produit de la base DistrisysSA pourrait être avantageusement remplacée par une vue de l'entité Produit du Master Data.

Le flux que nous allons étudier va récupérer les données de cette table Produit de DistrisysSA.

Puis il va venir comparer ces données avec le contenu de DimProduit de DistrisysDW.

Dans la plupart des cas, les flux de dimensions doivent vérifier :

- S'il y a un nouveau membre à ajouter dans la dimension.
- S'il y a eu une modification dans une des propriétés d'un élément de la dimension. Si c'est le cas, le flux effectue la mise à jour.
- Dans le cas de modification ou d'ajout d'un nouveau membre, le flux doit émettre une alerte.

Nous verrons cela dans ce chapitre à la section L'audit des flux ETL.

Commençons la procédure.

- Ouvrez le package **DW_Dimproduit.dtsx**.
- Le flux de contrôle ne contient qu'une tâche de flux de données. Cliquez sur l'onglet **Flux de données**.

Flux ETL de mise à jour de la dimension Produit

- Celle de la table Produit de DistrisysSA.
- Celle de la table DimProduit de DistrisysDW.

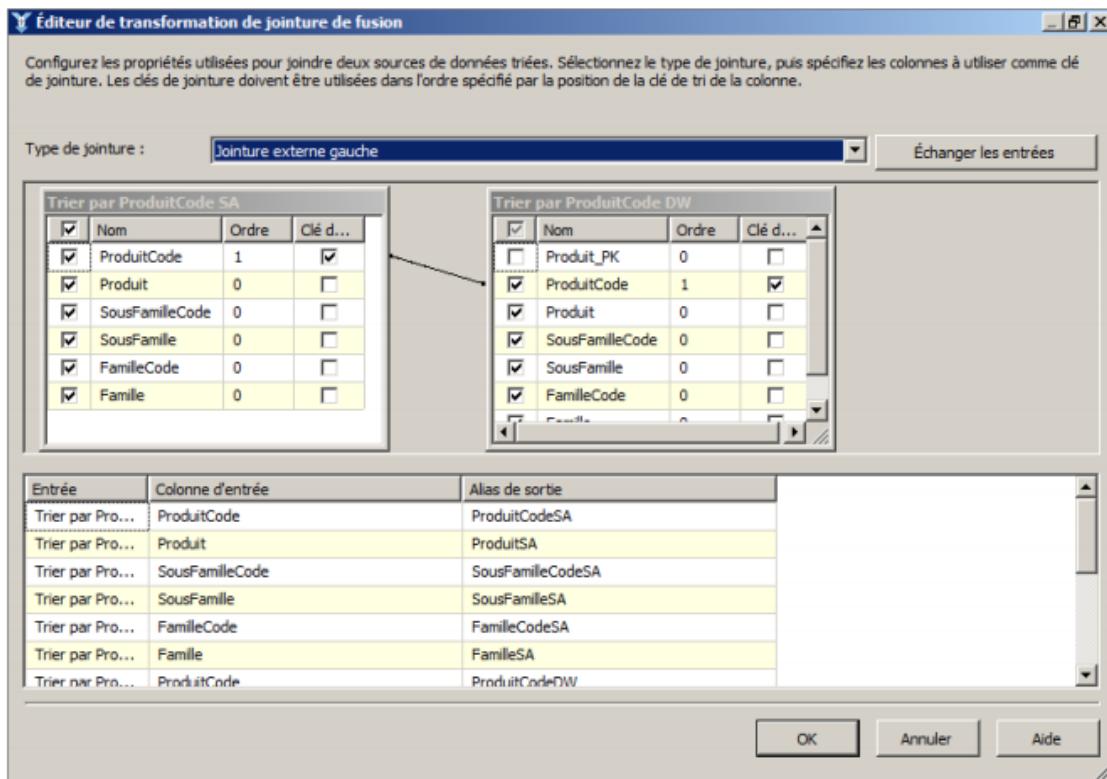
La tâche de colonne dérivée **Epurer** les chaînes de caractères n'est là que pour nettoyer les chaînes de caractères de tout caractère d'espacement à droite et à gauche, pouvant altérer la transformation qui va suivre.

L'idée du flux est de venir comparer ces deux sources de données, en réalisant une jointure externe gauche en faveur des données de DistrisysSA, à la manière d'un LEFT OUTER JOIN en SQL.

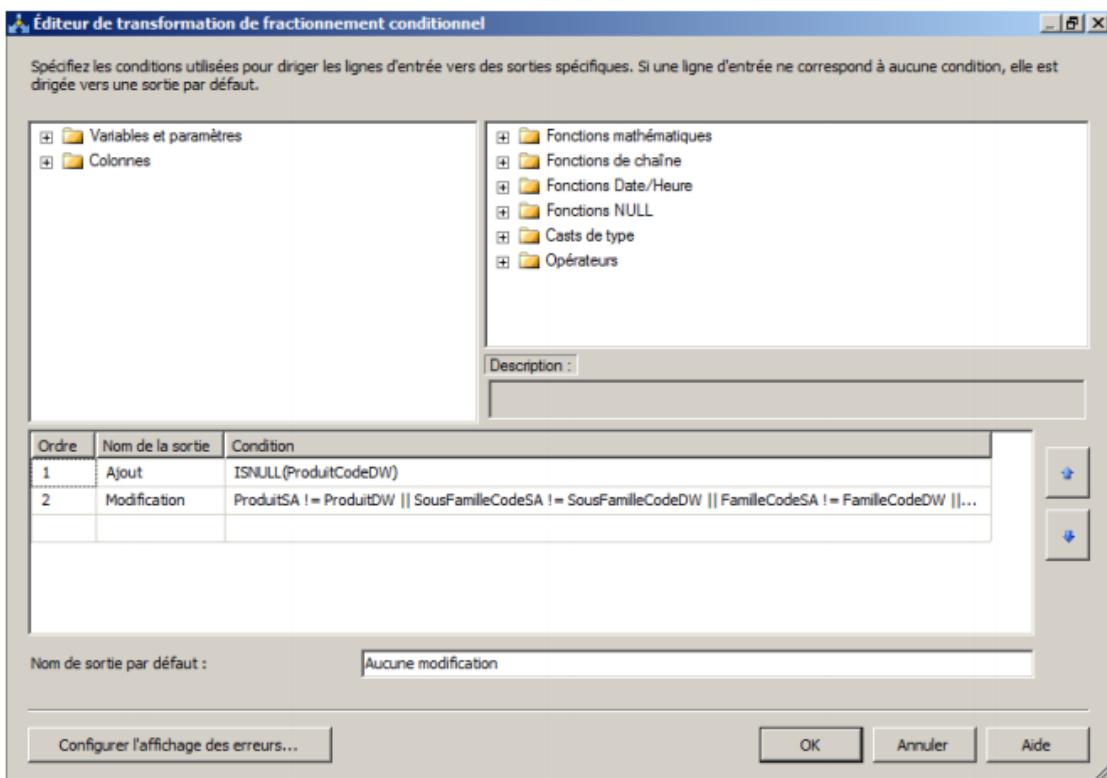
Les tâches de tri en entrée de la tâche de fusion sont nécessaires, d'une part pour des considérations de performance, d'autre part parce que c'est l'ordre de tri qui détermine la clé de jointure de la tâche de fusion.

Nous récupérons en sortie de la tâche de fusion les champs des deux sources de données.

La tâche Jointure de fusion externe à gauche est configurée ainsi :



En sortie de fusion, la tâche **Fractionnement conditionnel**, nommée **Identification des nouveaux produits et des produits à mettre à jour**, permet par des tests simples d'identifier les lignes en ajout, les lignes à modifier, les lignes pour lesquelles aucun traitement n'est nécessaire.



S'il s'agit d'une nouvelle ligne, une tâche de **Destination OLE DB** se charge de faire l'ajout dans la table DimProduit de DistrisysDW.

Il est important de laisser la clé produit_PK être gérée par SQL Server, en vérifiant bien qu'elle soit en incrémentation automatique.

S'il s'agit d'une modification, les lignes sont mises à jour à l'aide d'une tâche **Commande OLE DB**.

La requête est paramétrée, c'est-à-dire qu'elle est fonction de paramètres (identifiés par un ?) que nous venons mapper sur des champs disponibles dans le flux.

La requête exécutée est la suivante :

```
UPDATE [DimProduit]
    SET [Produit] = ?,[SousFamilleCode] = ?,
[SousFamille] = ?,[FamilleCode] = ?,[Famille] = ?
    WHERE [ProduitCode] = ?
```

Le mappage des paramètres de la requête peut être visualisé en double cliquant sur la tâche **Mise à jour des produits modifiés** puis sur l'onglet **Mappage de colonnes**:

À noter que l'ordre des paramètres est fonction de leur apparition dans la requête SQL.

Réalisons la procédure suivante afin de tester le fonctionnement du flux :

- Modifiez les connexions **DistrisysSA** et **DistrisysDW** afin de les adapter à votre environnement.
- Lancez le flux en cliquant sur **Exécuter le package**. Assurez-vous qu'au premier lancement, le flux ajoute une nouvelle ligne et met à jour une seconde ligne.
- Servez-vous de SSMS, afin de constater les modifications dans la table **DimProduit** avant et après l'exécution du flux.
- Avec **SSMS**, faites des modifications manuelles dans la table **Produit DistrisysSA** et relancez le flux. Vérifiez que vos modifications sont bien prises en compte.

Nous venons de voir comment réaliser un flux ETL de mise à jour de dimensions. Ce flux est typique; c'est-à-dire que la plupart des flux ETL de mise à jour des dimensions auront ce formalisme. Néanmoins, il existe des cas particuliers. C'est ce que nous verrons dans la partie suivante.

3-3-2 - Cas d'une dimension en SCD

Dans certains cas, la modification des données est trop brutale pour l'utilisateur. Il faut alors gérer les modifications dans la dimension de manière plus fine.

Prenons un exemple, chez Distrisys, la vente de cafetières bat des records en prenant une part de plus en plus conséquente dans le CA de la société.

La stratégie de l'entreprise est donc de développer la vente de cafetières. Une nouvelle organisation commerciale est décidée et un chef produit va être nommé à cet effet.

Cafetière est aujourd'hui une sous-famille de l'activité Petit Ménager. Distrisys souhaite que Cafetière devienne une famille Produit au même titre que Grand Ménager et Petit Ménager .

Si en mars l'équipe décisionnelle procède, comme le fait le flux précédent, à l'écrasement du code famille Petit Ménager par un nouveau code Cafetière, nous ne répondons pas au besoin de la direction. Nous perdons la vision de l'historique. Nous entrons alors dans un cas de configuration dit de la dimension à évolution lente.

Le concept de dimension à variation lente ou SCD (Slowly Changing Dimension) a été défini par Ralph Kimball, le père fondateur de l'entrepôt de données.

Ralph Kimball identifie plusieurs types de modifications au niveau de chaque champ de données de la dimension :

- **Le type 0 (attribut fixe)** : le champ ne peut être modifié. Dans notre cas, il s'agit du ProduitCode ou du Produit_PK par exemple.
- **Le type 1 (modification d'attribut)** : le champ peut être modifié. En cas de modifications, l'ancienne valeur est simplement écrasée par la nouvelle valeur. C'est le cas que nous avons mis en œuvre au flux précédent avec les champs Produit, SousFamilleCode, Sous Famille, FamilleCode et Famille.
- Enfin, il existe **le type 2** (attribut d'historique) qui suggère que les modifications du champ considéré n'impactent pas les valeurs passées mais seulement les valeurs futures.

Concrètement, les lignes de faits déjà chargées continueront à être affectées à l'ancien membre de la dimension. Les futures lignes de faits seront affectées à un nouveau membre de la dimension comportant les mêmes caractéristiques que le précédent, à l'exception près du champ de type 2 modifié.

Prenons l'exemple de notre dimension DimProduit actuelle :

A	B	C	D	E	F	G	H
Produit_PK	ProduitCode	Produit	SousFamilleCode	SousFamille	FamilleCode	Famille	Valide
1 LL1100	LAGON LL1100	LL	Lave-Linge	GM	Gros Menager	1	
2 LL1200	LAGON LL1200	LL	Lave-Linge	GM	Gros Menager	1	
3 LV1620	LAGON LV 1620	LV	Lave-Vaisselle	GM	Gros Menager	1	
4 SL1000	LAGON SL 1000	SL	Seche-Linge	GM	Gros Menager	1	
5 F120	Pierre Michel F120	F	Four	GM	Gros Menager	1	
6 R080	Pierre Michel R 080	R	Refrégirateur	GM	Gros Menager	1	
7 GP700	Cuccina GP 700	GP	Grille-Pain	PM	Petit Menager	1	
8 C470	Cuccina C 470	C	Cafetière	PM	Petit Menager	1	
9 RC3000p	Cuccina RC 3000+	RC	Robot Cuisine	PM	Petit Menager	1	
10 C260	Cuccina C 260	C	Cafetière	PM	Petit Menager	1	
11 C270	Cuccina C 270	C	Cafetière	PM	Petit Menager	1	

Nous ajoutons une colonne valide. La valeur 1 signifie que le produit est en activité. Une valeur à 0, signifie qu'il n'est plus en activité.

Les ventes de cafetières, dans la table de faits des factures, vont être affectées à la clé technique Produit_PK 8, 10 ou 11.

Imaginons que nous sommes le 1 er mai. La création de la nouvelle Famille Cafetière a lieu. Le champ FamilleCode étant de type 2, nous devrons avoir la nouvelle table DimProduit suivante :

Produit_PK	ProduitCode	Produit	SousFamilleCode	SousFamille	FamilleCode	Famille	Valide
1 LL1100	LAGON LL1100	LL	Lave-Linge	GM	Gros Menager	1	
2 LL1200	LAGON LL1200	LL	Lave-Linge	GM	Gros Menager	1	
3 LV1620	LAGON LV 1620	LV	Lave-Vaisselle	GM	Gros Menager	1	
4 SL1000	LAGON SL 1000	SL	Seche-Linge	GM	Gros Menager	1	
5 F120	Pierre Michel F120	F	Four	GM	Gros Menager	1	
6 R080	Pierre Michel R 080	R	Refrégirateur	GM	Gros Menager	1	
7 GP700	Cuccina GP 700	GP	Grille-Pain	PM	Petit Menager	1	
8 C470	Cuccina C 470	C	Cafetière	PM	Petit Menager	0	
9 RC3000p	Cuccina RC 3000+	RC	Robot Cuisine	PM	Petit Menager	1	
10 C260	Cuccina C 260	C	Cafetière	PM	Petit Menager	0	
11 C270	Cuccina C 270	C	Cafetière	PM	Petit Menager	0	
12 C470	Cuccina C 470	C	Cafetière	C	Cafetière	1	
13 C260	Cuccina C 260	C	Cafetière	C	Cafetière	1	
14 C270	Cuccina C 270	C	Cafetière	C	Cafetière	1	

Les lignes, avec les anciennes clés techniques identifiant les cafetières, les Produit_PK 8,10 et 11, ont été invalidées.

En revanche, trois nouvelles lignes sont créées en remplacement des trois précédentes : il s'agit des lignes 12, 13 et 14, et ces nouvelles lignes sont valides et contiennent le même ProduitCode que les anciennes lignes.

Ainsi, les ventes de cafetières, à compter de cette date, vont être affectées dans la table de faits des factures aux nouvelles clés Produit_PK 12, 13 et 14.

Les valeurs passées de la table de faits n'ont de ce fait pas été affectées par ce changement. SSIS nous aide à mettre en œuvre le SCD. Pour cela, nous disposons d'une tâche d'assistance, la tâche **Dimension à variation lente**. Réalisons maintenant le flux de chargement des produits utilisant le SCD.

- Si vous ne souhaitez pas le réaliser vous-même, le package **DW_DimProduit_Avec_SCD.dtsx** est disponible dans la solution DistrisysETL précédemment téléchargée.
- Avant toute chose, dans SSMS, **modifiez la structure de la table DimProduit** en ajoutant une nouvelle colonne **Valide** de type **bit**.

SERVEUR1.DistrisysDW - dbo.DimProduit*			
	Nom de la colonne	Type de données	Autoriser l...
!	Produit_PK	int	<input type="checkbox"/>
	ProduitCode	varchar(10)	<input type="checkbox"/>
	Produit	varchar(20)	<input type="checkbox"/>
	SousFamilleCode	varchar(10)	<input type="checkbox"/>
	SousFamille	varchar(20)	<input type="checkbox"/>
	FamilleCode	varchar(10)	<input type="checkbox"/>
	Famille	varchar(20)	<input type="checkbox"/>
►	Valide	bit	<input checked="" type="checkbox"/>

Propriétés des colonnes	
A Z	
(Général)	
(Nom)	Valide
Autoriser les valeurs Null	Oui
Type de données	bit
Valeur ou liaison par défaut	1

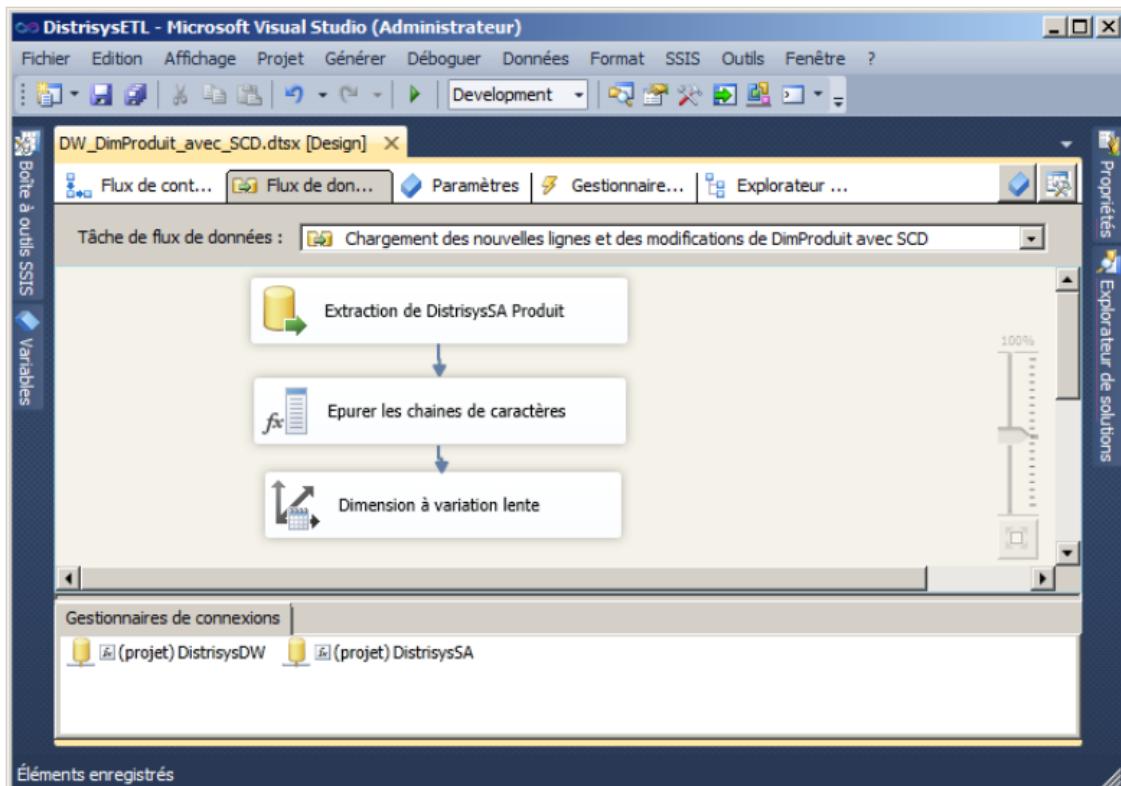
Spécifiez une **valeur par défaut** de valide à **True**.

Modifiez toutes les lignes de DimProduit pour que le champ **Valide** soit à **True**.

	Produit_PK	ProduitCode	Produit	SousFamilleC...	SousFamille	FamilleCode	Famille	Valide
►	0	INC	Inconnu	INC	Inconnu	INC	Inconnu	True
	1	LL1100	LAGON LL1100	LL	Lave-Linge	GM	Gros Menager	True
	2	LL1200	LAGON LL1200	LL	Lave-Linge	GM	Gros Menager	True
	3	LV1620	LAGON LV 1620	LV	Lave-Vaisselle	GM	Gros Menager	True
	4	SL1000	LAGON SL 1000	SL	Seche-Linge	GM	Gros Menager	True
	5	F120	Pierre Michel F120	F	Four	GM	Gros Menager	True
	6	R080	Pierre Michel R 080	R	Refrigérateur	GM	Gros Menager	True
	7	GP700	Cuccina GP 700	GP	Grille-Pain	PM	Petit Menager	True
	8	C470	Cuccina C 470	C	Cafeti�re	PM	Petit Menager	True
	9	RC3000p	Cuccina RC 3000+	RC	Robot Cuisine	PM	Petit Menager	True
	10	C260	Cuccina C 260	C	Cafeti�re	PM	Petit Menager	True
	11	C270	Cuccina C 270	C	Cafeti�re	PM	Petit Menager	True

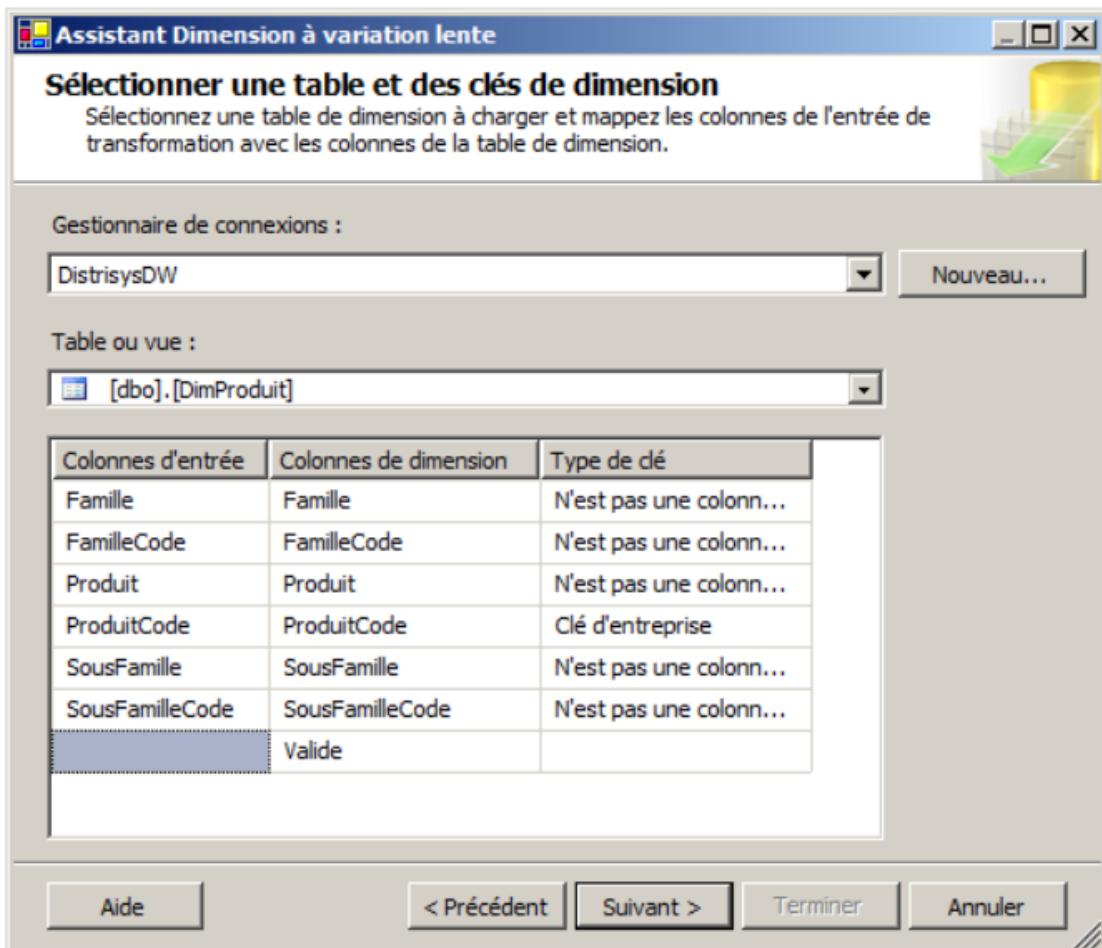
- Dans le projet SSIS, cr ez un nouveau Package **DW_DimProduit_avec_SCD.dtsx**.
- Cr ez les deux connexions sources  **DistrisysSA** et  **DistrisysDW**.
- Dans le flux de contrle, glissez une **t che de flux de donn es**.
- Puis, dans **Flux de donn es**, glissez une **t che Source OLE DB** se connectant  **DistrisysSA** et  la table **Produit**.
- Vous pouvez galement glisser une **t che Colonne d riv e**, pour faire le nettoyage des champs de type cha nes de caract res (cela est facultatif).
- Glissez ensuite la t che **Dimension  variation lente**.
- Connectez la t che de colonne d riv e  la t che **Dimension  variation lente**.

Vous avez alors les trois t ches suivantes :



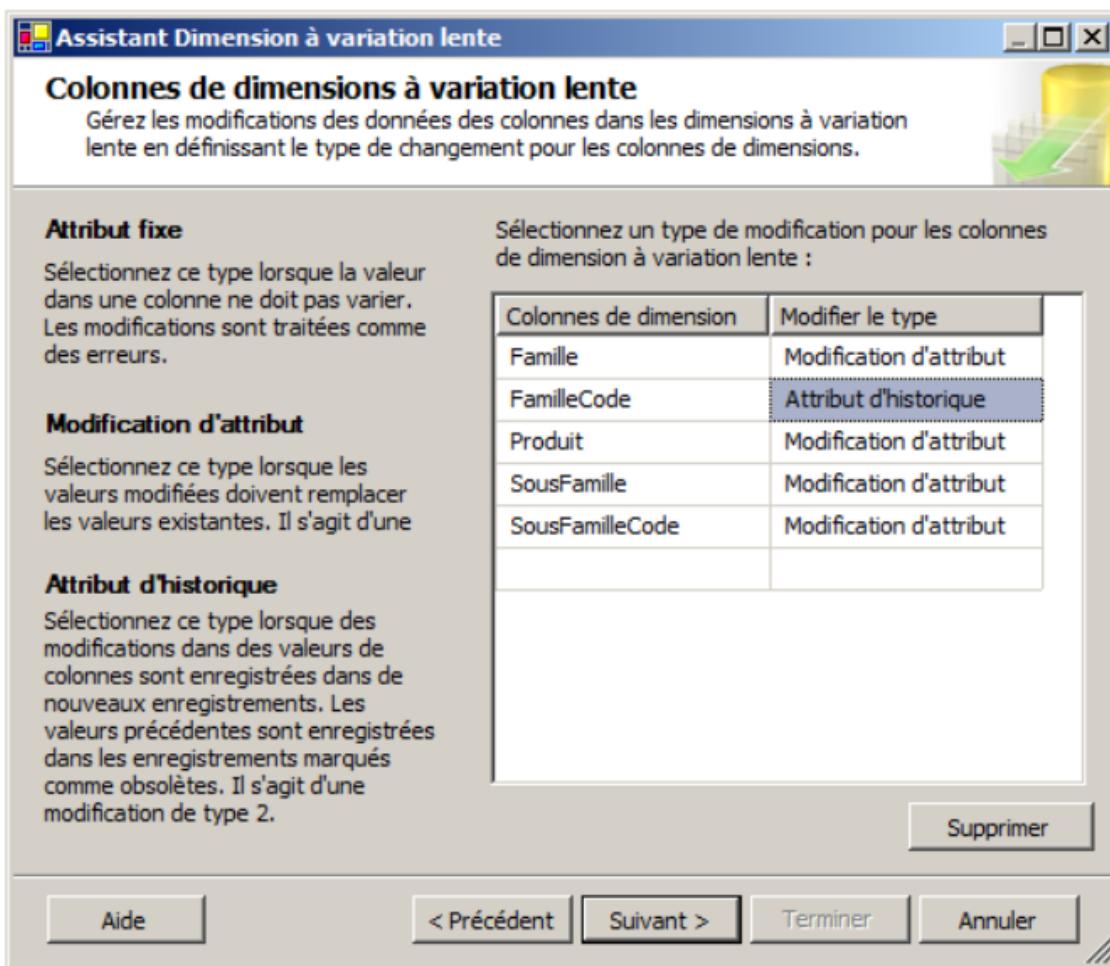
- Puis double cliquez pour modifier la t che **Dimension  variation lente**. Un assistant s'ouvre, cliquez sur **Suivant**.
- Identifiez  l' cran **Sélectionner une table et des cl s de dimension**, la connexion **DistrisysDW**, la table **DimProduit** et sélectionnez **ProduitCode** comme tant la **cl e d'entreprise**  l'aide du menu d roulant.

- Dans la **Colonnes d'entrée**, sélectionnez les mêmes noms que ceux apparaissant dans **Colonnes de dimension**. Puis cliquez sur le bouton **Suivant**.



Le SCD fait apparaître avec évidence la nécessité d'identifier des clés techniques différentes des clés d'entreprise. Dans notre cas, Produit_PK est une clé technique et ProduitCode est la clé d'entreprise.

Au niveau de l'interface, **Colonne de dimensions à variation lente**, sélectionnez le type d'attribut et procédez comme ce qui est indiqué ci-dessous.

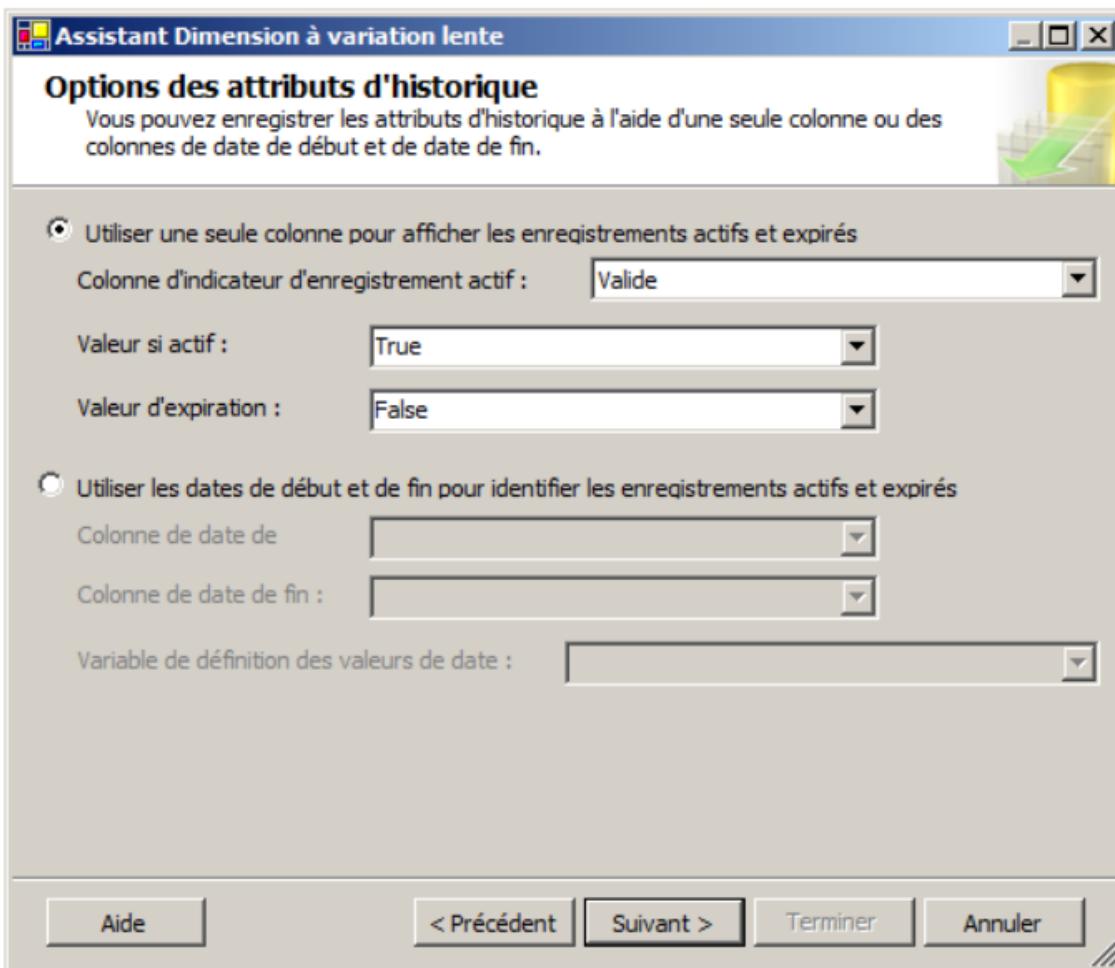


Dans notre cas, nous spécifions les champs **SousFamilleCode**, **SousFamille**, **Famille** et **Produit** en attribut de type 1 : **Modification d'attribut**.

En revanche, le champ **FamilleCode** est en type 2 : **attribut d'historique**.

- À l'écran **Options des attributs fixes et variables**, cliquez sur **Suivant**.

- À l'écran **Options des attributs d'historique**, sélectionnez l'option **Utiliser une seule colonne** pour afficher les enregistrements actifs et expirés, puis configurez les champs comme ci-dessous :



SSIS intègre plusieurs modes de gestion des lignes valides et obsolètes : soit la gestion par un champ unique (le champ Valide dans notre cas), soit la gestion par encadrement de dates . Un champ identifie alors la date de début de validité et un autre la date de fin de validité.

Sur l'écran Membres de la dimension inférés, décochez la case Activer la prise en charge des membres inférés.

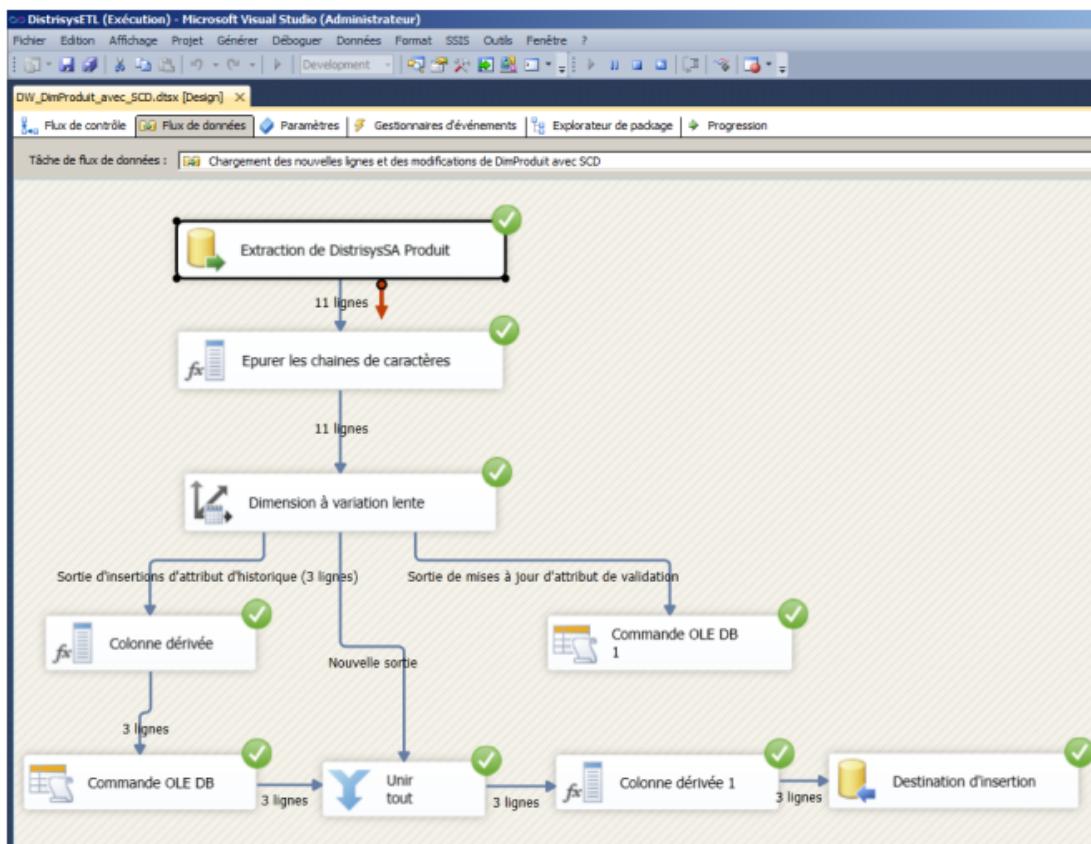
La gestion des membres inconnus ou des membres n/a est une gestion que je vous conseille de prendre en charge vous-même au sein de votre flux. C'est un aspect pris en charge dans la gestion d'audit.

Sur l'écran Fin de l'assistant Dimension à variation lente, cliquez sur Terminer.

En fin d'assistant, SSIS génère alors les tâches correspondantes au comportement attendu par le SCD.

Pour tester, faites des modifications avec SSMS dans la table Produit de DistrisysSA.

- Par exemple affectez les produits de type cafetière à une nouvelle FamilleCode **C** et **Famille Cafetière**.
- Puis exécutez le flux.



Fin d'exécution d'un flux de mise à jour d'une dimension à variation lente

Table DimProduit, avant exécution du SCD :

SERVEUR1.DistrisysDW - dbo.DimProduit								
	Produit_PK	ProduitCode	Produit	SousFamilleC...	SousFamille	FamilleCode	Famille	Valide
▶	0	INC	Inconnu	INC	Inconnu	INC	Inconnu	True
	1	LL1100	LAGON LL1100	LL	Lave-Linge	GM	Gros Menager	True
	2	LL1200	LAGON LL1200	LL	Lave-Linge	GM	Gros Menager	True
	3	LV1620	LAGON LV 1620	LV	Lave-Vaisselle	GM	Gros Menager	True
	4	SL1000	LAGON SL 1000	SL	Seche-Linge	GM	Gros Menager	True
	5	F120	Pierre Michel F120	F	Four	GM	Gros Menager	True
	6	R080	Pierre Michel R 080	R	Refrigérateur	GM	Gros Menager	True
	7	GP700	Cuccina GP 700	GP	Grille-Pain	PM	Petit Menager	True
	8	C470	Cuccina C 470	C	Cafetière	PM	Petit Menager	True
	9	RC3000p	Cuccina RC 3000+	RC	Robot Cuisine	PM	Petit Menager	True
	10	C260	Cuccina C 260	C	Cafetière	PM	Petit Menager	True
	11	C270	Cuccina C 270	C	Cafetière	PM	Petit Menager	True

Table DimProduit, après exécution du SCD :

	Produit_PK	ProduitCode	Produit	SousFamilleC...	SousFamille	FamilleCode	Famille	Valide
▶	0	INC	Inconnu	INC	Inconnu	INC	Inconnu	True
	1	LL1100	LAGON LL1100	LL	Lave-Linge	GM	Gros Menager	True
	2	LL1200	LAGON LL1200	LL	Lave-Linge	GM	Gros Menager	True
	3	LV1620	LAGON LV 1620	LV	Lave-Vaisselle	GM	Gros Menager	True
	4	SL1000	LAGON SL 1000	SL	Seche-Linge	GM	Gros Menager	True
	5	F120	Pierre Michel F120	F	Four	GM	Gros Menager	True
	6	R080	Pierre Michel R 080	R	Refrigérateur	GM	Gros Menager	True
	7	GP700	Cuccina GP 700	GP	Grille-Pain	PM	Petit Menager	True
	8	C470	Cuccina C 470	C	Cafetière	PM	Petit Menager	False
	9	RC3000p	Cuccina RC 3000+	RC	Robot Cuisine	PM	Petit Menager	True
	10	C260	Cuccina C 260	C	Cafetière	PM	Petit Menager	False
	11	C270	Cuccina C 270	C	Cafetière	PM	Petit Menager	False
	12	C470	Cuccina C 470	C	Cafetière	C	Cafetière	True
	13	C260	Cuccina C 260	C	Cafetière	C	Cafetière	True
	14	C270	Cuccina C 270	C	Cafetière	C	Cafetière	True

Nous obtenons exactement le comportement attendu initialement. Le SCD a fonctionné en créant trois nouveaux membres valides, et en invalidant les trois membres remplacés.

- Attention, dans la réalité la tâche SCD doit être réfléchie au moment de la conception de l'entrepôt de données et de la conception des flux. Le SCD doit être mis en œuvre uniquement dans le cas de dimension de taille raisonnable, soit environ moins de 20000 lignes et uniquement dans le cas de champs ne variant pas beaucoup, ou du moins lentement. Dans le cas de dimensions larges ou de dimensions à variation rapide, comme par exemple la dimension Abonné d'un opérateur téléphonique ou la dimension Produit en grande distribution, nous utiliserons plutôt le flux précédent et gérerons l'historisation avec un autre procédé qui affecte directement la modélisation du DW : le principe de mini-dimension.

Nous rappelons que ce flux est incomplet sans la mise en œuvre de l'audit.

3-4 - Réaliser un flux pour charger une table de faits

Nous allons à présent réaliser le flux de chargement des tables de faits FactFacture et FactFactureEntete de l'entrepôt de données DistrisysDW.

Dans tous les cas, un flux de chargement de tables de faits a les caractéristiques suivantes :

- Il fait suite au chargement et à la mise à jour de toutes les tables de dimension.
- Il doit s'assurer, avant l'insertion, des contraintes d'intégrité entre la table de faits et ses dimensions.
- Il se charge uniquement en insertion rapide.
- Il possède toutes les caractéristiques d'un flux ETL : Extraction simple d'une source (pas de grandes requêtes SQL), puis passage par des tâches de transformations et d'évaluation des données, et enfin chargement rapide des données.

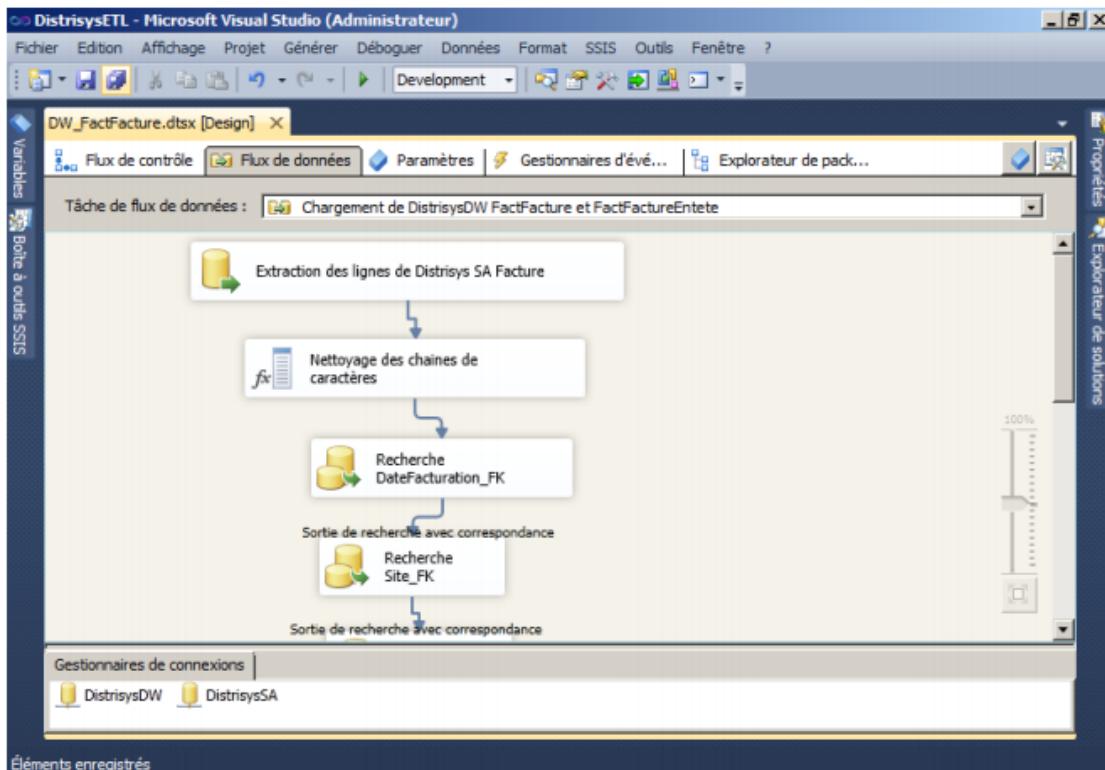
Nous ne ferons jamais de mise à jour de données par requête SQL update sur une table de faits. Si vous devez en arriver à de telles pratiques, révisez la stratégie ou la conception de vos flux, il y a forcément une meilleure solution.

Décrivons maintenant le flux de chargement des tables de faits **FactFacture** et **FactFactureEntete**.

Le package Dw_FactFacture.dtsx, il illustrant ce processus et détaillé dans cette partie, est disponible dans la solution Distrisys ETL précédemment téléchargée. Reportez-vous y pour obtenir des détails complémentaires.

- Attention, ce flux est brut, c'est-à-dire qu'il n'intègre pas encore la gestion des erreurs et des audits. Ce flux est donc incomplet mais suffisant pour en comprendre son essence.

Tout d'abord, ces deux tables de faits disposent en réalité d'une source unique : la table Facture du sas de données DistrisysSA. Le chargement de ses tables se fera alors à partir de la même extraction de données.

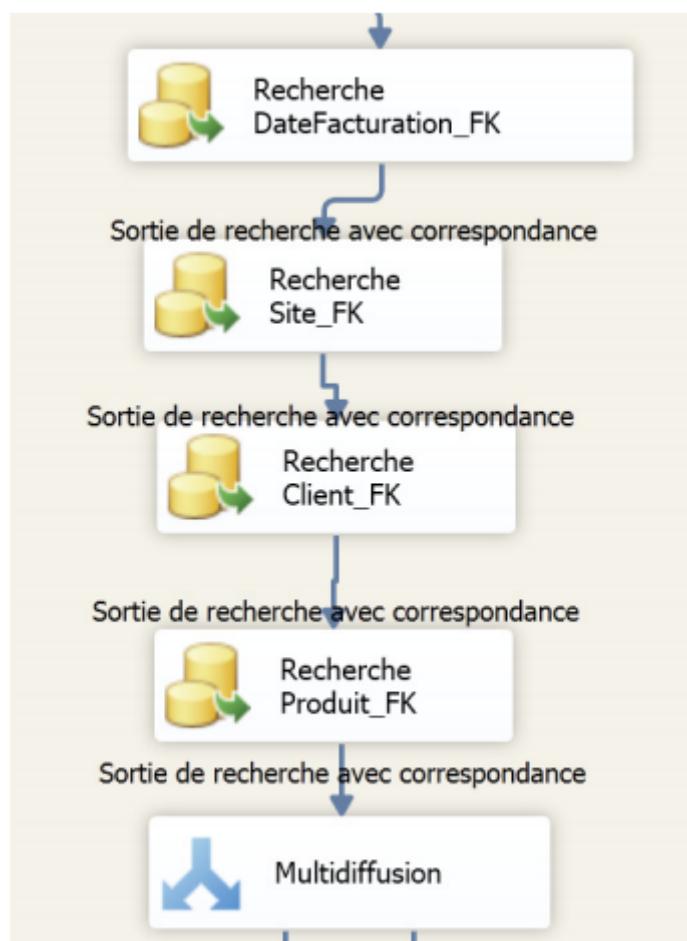


La tâche **Nettoyage des chaînes de caractères**, de type colonne dérivée, s'assure que des espaces, à droite ou à gauche de la chaîne de caractères identifiant notamment un code, ne viennent pas polluer la compréhension de la donnée.

Nom de la colonne d...	Colonne dérivée	Expression	Type de données
SiteCode	Remplacer 'SiteCode'	TRIM(SiteCode)	chaîne [DT_STR]
ProduitCode	Remplacer 'ProduitCode'	TRIM(ProduitCode)	chaîne [DT_STR]
ClientCode	Remplacer 'ClientCode'	TRIM(ClientCode)	chaîne [DT_STR]

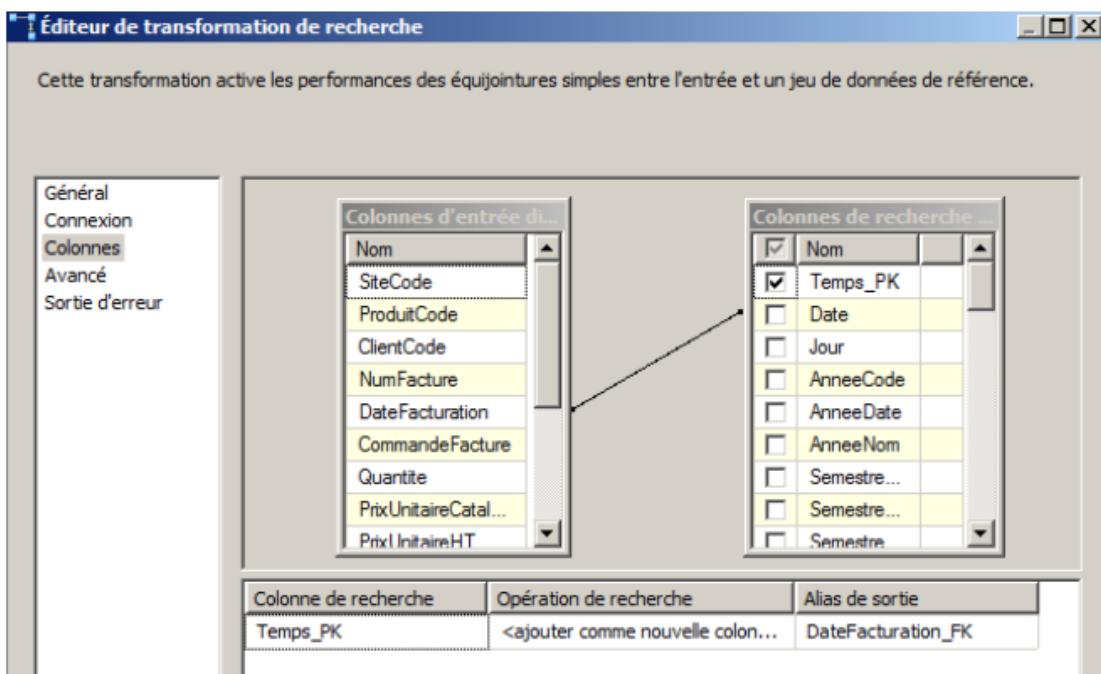
Ensuite, le flux va enchaîner une chaîne de tâches **Recherche**, visant à traduire la date de facturation ainsi que les codes produit, client et site en clé technique.

Cette succession de tâches **Recherche** constitue la meilleure vérification des contraintes d'intégrité entre tables de faits et de dimensions.



Au niveau de la tâche **Recherche DateFacturation_FK**, la configuration est la suivante :

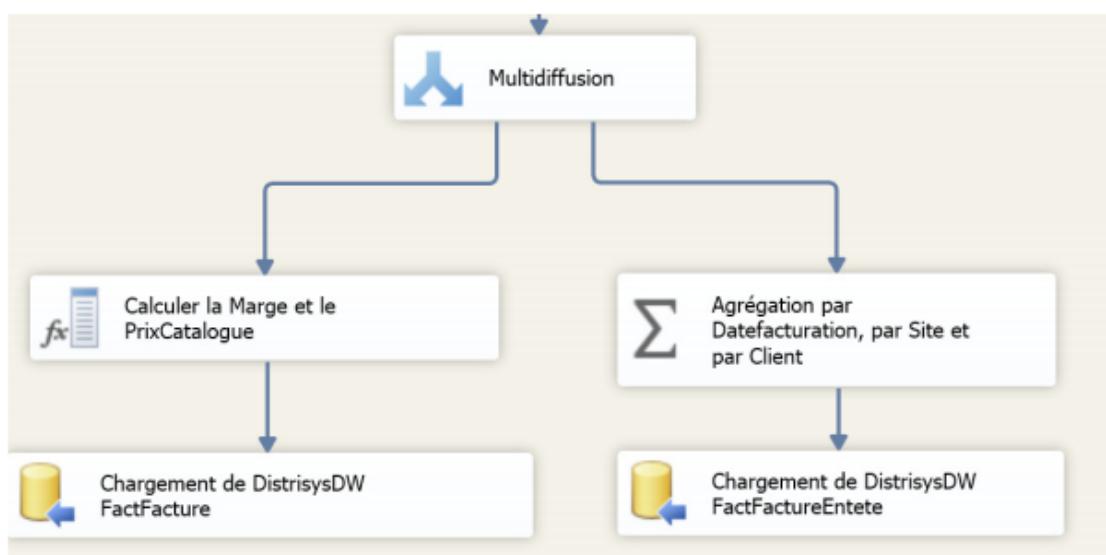
- La table de référence est **DimTemps**.
- Le mapping est réalisé entre le champ **DateFacturation** en provenance de la source de données et le champ Date de la table de la dimension DimTemps.
- Le champ **Temps_PK**, renommé **Datefacturation_FK**, est en sortie de correspondance.



Ensuite, le flux se divise en deux :

- Un premier flux va charger la table FactFacture.
- Un second flux va charger la table FactFactureEntete.

La division du flux sans condition est assurée par la tâche **Multidiffusion**.



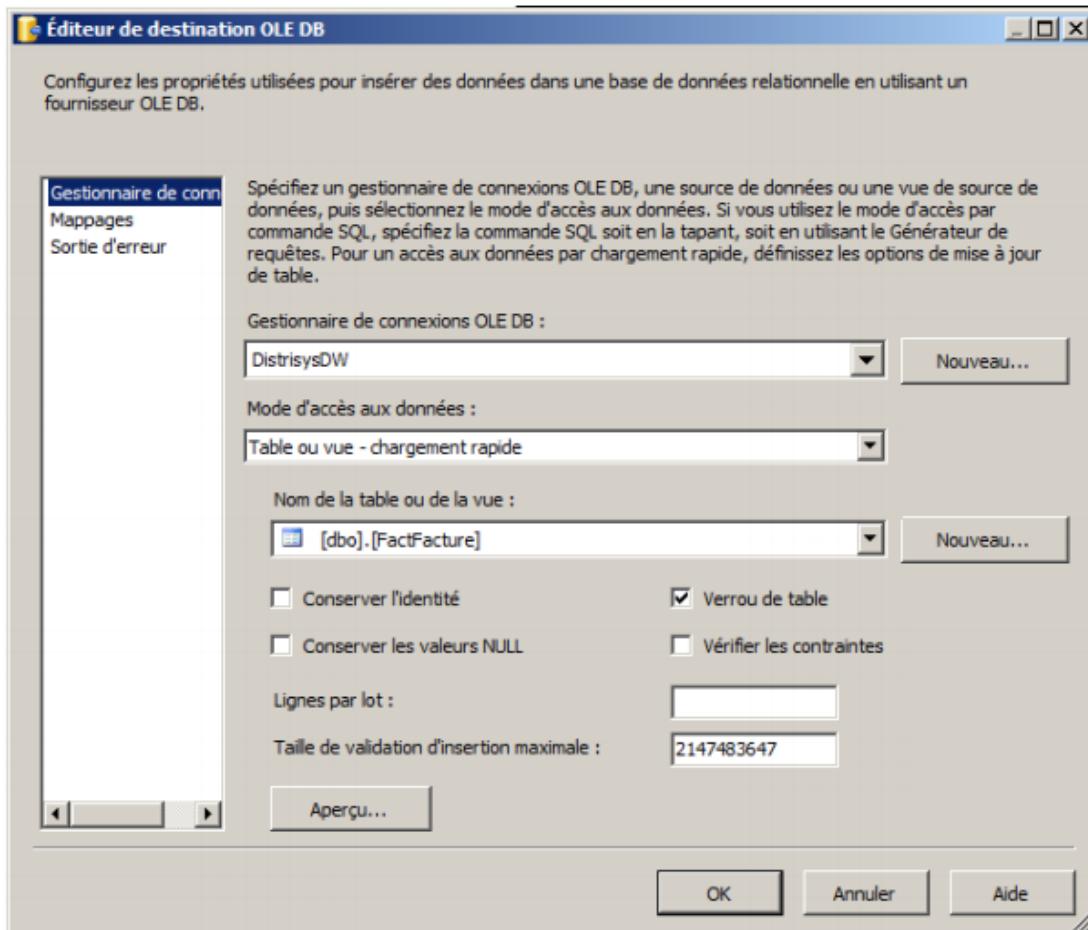
Avant de charger FactFacture, une dernière tâche de type Colonne dérivée **Calculer la Marge et le PrixCatalogue**, assure la création de deux champs manquants :

Nom de la c...	Colonne ...	Expression	Type
Marge	<ajouter...	(DT_NUMERIC,9,2)(PrixtotalHT - CoutDirectMatiere - CoutDirectMainOuvre - CoutIndirect)	numérique
PrixCatalogue	<ajouter...	(DT_NUMERIC,9,2)(PrixUnitaireCatalogue * Quantité)	numérique

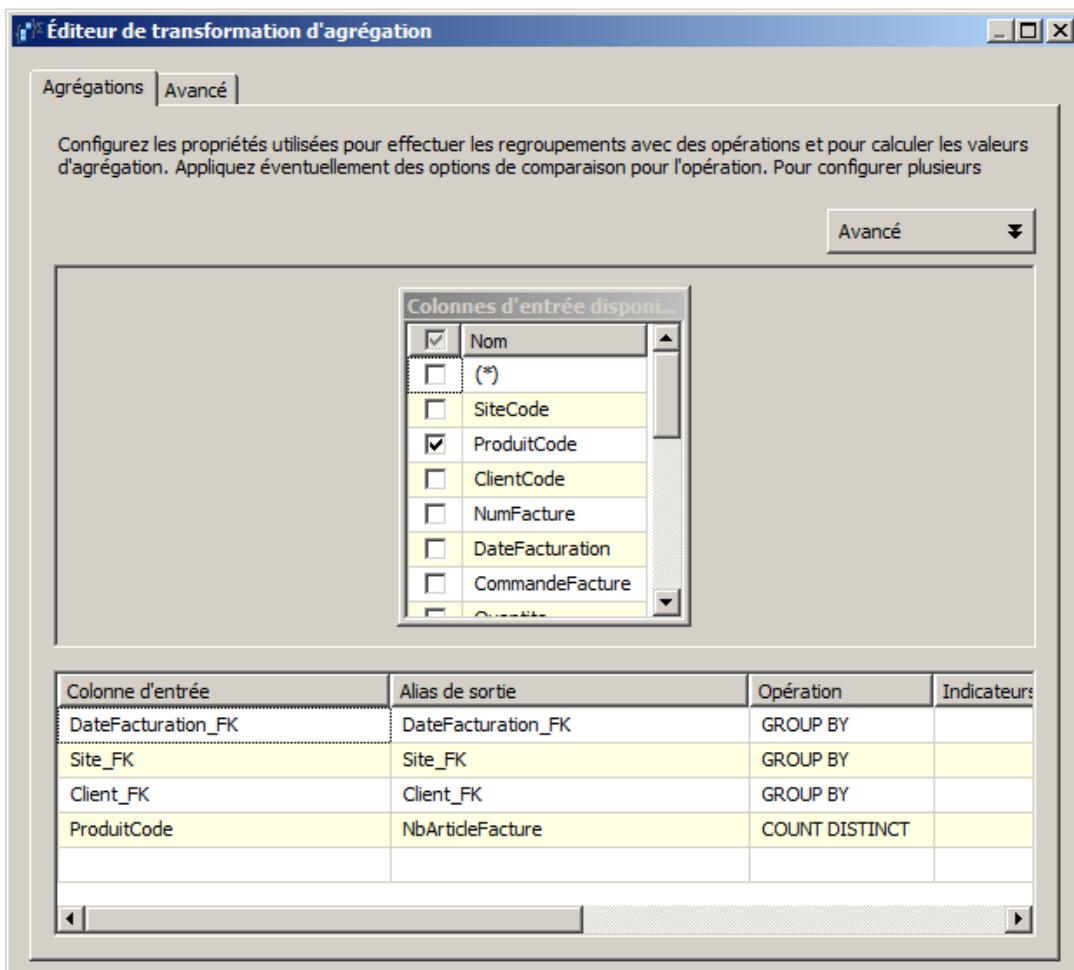
Puis la tâche effectue le chargement rapide vers la destination :

- L'option **Vérifier les contraintes** est désactivée, puisque les tâches de Recherche s'en sont déjà assurées. Le chargement n'en sera que plus rapide.

La disponibilité d'un grand nombre de disques durs est assez primordiale dans une architecture physique décisionnelle. Si les filegroup des tables Factfacture et FactFactureEntete étaient sur des disques physiques différents, le chargement n'en serait que moins risqué et plus rapide. Tout dépend ensuite du volume de données et du temps consenti au chargement ETL...



Le deuxième flux s'oriente vers une tâche de type **Agrégation** qui joue en fait le rôle d'un **Group By** en SQL. Cette tâche va regrouper les données suivant les colonnes DateFacturation_FK, Site_FK et Client_FK et compter le nombre de lignes distinctes Produit_PK, pour en déduire le nombre d'articles différents que comprend la facture.



La dernière tâche de type Destination OLE DB procède au chargement de FactFactureEntete sur les mêmes bases de configuration que FactFacture : chargement rapide et pas de vérifications des contraintes d'intégrité.

Nous venons de réaliser le flux de chargement de nos tables de faits FactFacture et FactFactureEntete. Mais ces flux ne sont pas achevés sans une gestion fine des erreurs et un audit du déroulement du flux.

À la prochaine étape, vous allez découvrir les quelques concepts liés à l'audit des flux ETL.

4 - L'audit des flux ETL

4-1 - Les objectifs de l'audit de flux ETL

Les exemples qui ont été présentés précédemment sont des flux inachevés, dans le sens où ses flux n'intègrent pas la gestion d'erreurs et l'audit du déroulement du flux.

Par expérience, l'audit de processus ETL, souvent appelé à tort gestion des rejets, génère soit beaucoup de fausses croyances, soit beaucoup de faux espoirs. Dans la plupart des cas, elle est même mise de côté. Une des principales idées reçues consiste à faire croire qu'un logiciel ou qu'un package miracle permet de mieux gérer la qualité des données. Dans les faits, l'audit des processus ETL est un travail de finesse du concepteur ETL, traitant un cas ou un contexte particulier. Si un cas peut difficilement être retracé à l'identique pour un autre cas, il en reste néanmoins des bonnes pratiques. C'est ce que nous allons voir au cours de cette partie.

Tout d'abord, nous allons nous poser les questions suivantes : qu'est donc l'audit de flux ETL ?

Quel est son objectif ?

En fait, l'audit de processus ETL poursuit des objectifs multiples et permet de répondre à de nombreuses questions. Cela signifie que suivant le contexte, on va rendre plus performant l'audit sur certains points plutôt que sur d'autres.

Les objectifs poursuivis par l'audit des processus ETL sont :

- L'audit de flux ETL permet d'informer du déroulement du processus ETL :

Le processus ETL a-t-il eu lieu ? Est-il terminé ? A-t-il terminé avec succès ? Quelle a été sa durée ?

- L'audit de flux ETL permet de traiter et d'alerter sur les erreurs rencontrées :

Quelle est la nature des erreurs rencontrées ? Combien y en a-t-il ? Quelles sont-elles ? Quelles sont les origines des problèmes ? Combien dénombre-t-on d'origines différentes ? Quelles sont les lignes concernées ? Combien de lignes sont concernées ?

- L'audit de flux ETL permet de suivre l'évolution de la performance du processus ETL :

Mon flux se fiabilise-t-il ? Génère-t-il de moins en moins d'erreurs ? Comment les durées d'exécution des différents flux évoluent-elles ?

- En cas d'erreur à rebours, sur une exécution de flux, l'audit de flux ETL doit permettre d'identifier les lignes concernées.

Un bon processus d'audit doit pouvoir répondre à toutes ces questions et peut-être même à d'autres, plus spécifiques à votre organisation.

Les maîtres mots de l'audit ETL sont **träçabilité** et **communication** :

- **Träçabilité**, vous l'aurez compris, pour répondre aux questions d'enquêtes classiques de type Qui ? Quoi ? Où ? Quand ? Comment ? Pourquoi ?
- **Communication**, car l'audit de processus ETL est intimement lié aux problématiques décisionnelles et donc aux prises de décision.

La délivrance de l'information ne vaut que si on est capable d'en estimer ou d'en évaluer son niveau de fiabilité.

Le système d'audit va être cette source d'informations. Les indicateurs issus du système d'audit ont la même valeur que les indicateurs métier. Et ces données présentées aux décideurs vont participer directement à la prise de décision.

4-2 - Conception d'un système d'audit de flux

Avant de réaliser le système d'au dit de flux, il est nécessaire de penser la stratégie de gestion d'erreurs.

Notre expérience nous a montré qu'il existe trois grandes stratégies ou scénarios de gestion d'erreurs, qui impliquent chacun des conceptions et des approches complètement différentes.

Ces stratégies sont les suivantes :

La publication garantie des données

La stratégie de chargement de 100 % des données, 100 % fiables.

Cette stratégie suggère que nous présentons des données aux utilisateurs qui sont complètes et qui sont complètement vérifiées, sans codification inconnue. Cette stratégie idéale implique que les informations fournies ne sont plus forcément de première fraîcheur. C'est la stratégie que retiennent le plus souvent les services de pilotage lorsqu'ils réalisent une publication mensuelle des données d'activité des mois écoulés; la publication étant réalisée plusieurs jours après la fin du mois.

L'utilisation d'une base de rejets

La stratégie de chargement de x% des données, 100 % fiables.

Cette stratégie suggère le rejet des lignes en erreurs et leur recharge par l'équipe informatique après correction. C'est la stratégie préférée mise en œuvre dans les systèmes jeunes. Mais c'est aussi celle qui, au final, est la plus difficile à entretenir à moyen et long terme, car elle implique deux choses : d'une part, que le système ne rejette que les lignes ayant un problème technique, ce qui est loin d'être la totalité des problèmes rencontrés et d'autre part parce qu'avec la croissance du système décisionnel, le temps passé par l'équipe informatique à la lecture des rejets, à leurs corrections et à leur recharge prend trop de temps et d'énergie. Au final, elle est très peu utilisée dans les systèmes matures.

Système avec reprise automatique d'erreurs

La stratégie de chargement de 100% des données, x % fiable.

Cette stratégie suggère que nous chargions l'ensemble des données, sans rejets, mais que nous acceptions et nous communiquons en conséquence sur la fiabilité des données qui ont été chargées. Pour être efficace, ce système doit communiquer suffisamment pour rendre les utilisateurs acteurs de la correction de leurs données. Dans ce système, l'équipe informatique doit déléguer autant que possible cette tâche de correction d'erreurs. Leur tâche doit se cantonner à rendre le système suffisamment lisible pour que ce soit les utilisateurs eux-mêmes qui apportent les corrections nécessaires. Enfin le système doit être suffisamment intelligent pour se reprendre de lui-même : il doit pouvoir revenir sur les flux précédemment exécutés et donc sur les données précédemment chargées avec erreurs. En espérant que les données ont bien évidemment été corrigées dans les systèmes opérationnels ou dans les référentiels de données.

Nous pouvons faire la synthèse de ces différentes stratégies dans le tableau ci-dessous :

Stratégie de gestion d'erreurs	Données du DW	Fiabilité des données du DW	Avantages	Inconvénients	Usages
Publication garantie des données	100% (complètes)	100% (garanties)	Très bonne visibilité pour le décideur final qui peut s'y fier.	Données qui ne sont généralement pas très fraîches. De quelques jours au mieux	Usage réservé généralement à des services métier dans le cadre de l'utilisation de magasins de données (hors du périmètre de l'ouvrage).
Utilisation d'une base de rejets x Utilisation d'une base de rejets Utilisation	x% (partielles)	100% (garanties sans erreurs techniques)	Fraîcheur des données.	Aucune évaluation de la qualité des données. Les données sont potentiellement fausses	Pour une mise en œuvre rapide, une démonstration.

d'une base de rejets					
----------------------	--	--	--	--	--

Stratégie de gestion d'erreurs	Données du DW	x% Fiabilité des données du DW	Avantages	Inconvénients	Usages
Stratégie de reprise automatique d'erreurs	100% (complètes)	x% (partielles)	Fraîcheur des données. la plupart des données en erreur sont visibles, car marqué d'un membre inconnu. Ce sont les utilisateurs qui corrigent leurs données et le système se reprend de lui-même.	Système à penser et à concevoir à la base. Pour être efficace le système doit être communiquant et offrir réellement la possibilité aux utilisateurs de corriger leurs données.	Usage préféré à mettre en place dans le cadre d'un entrepôt de données

Dans notre cas, nous allons opter pour la stratégie de reprise automatique d'erreurs. Nous allons voir dans la suite du chapitre comment mettre en œuvre une telle stratégie.

Concrètement, le système d'audit va être organisé autour de deux tables principales :

- **AuditFlux** : cette table va faire le bilan de l'exécution d'un flux en particulier.
- **AuditEvenement** : cette table va enregistrer les différents événements survenant lors de l'exécution du flux.

Le système de reprise de données va être le suivant :

- À chaque début de flux de chargement de tables de faits, le flux va parcourir la table AuditFlux pour identifier les flux à reprendre.
- Pour chaque flux à reprendre, il va supprimer les lignes, ajoutées précédemment par ce flux, et va rejouer le flux sur la même plage de dates, en espérant que les données d'erreurs précédemment remontées auront été alors corrigées.
- Une fois tous les flux rejoués, il va ajouter et initialiser une nouvelle ligne dans la table AuditFlux.
- Puis le flux du jour va s'exécuter normalement en chargeant les données de la plage de dates du jour.
- Lors de l'exécution du flux des événements vont s'ajouter dans la table AuditEvenement : des événements d'informations (nb de lignes extraites...) et des événements d'erreurs (code non trouvé, valeur impossible...) qui nécessiteront alors peut-être une reprise du flux.
- En fin d'exécution du flux, la ligne identifiant le flux sera mise à jour pour faire le bilan de l'exécution du flux.
- Si le flux est identifié comme étant à reprendre, il sera alors repris à la prochaine exécution.
- Attention, le système de reprise de données n'est valable que pour les flux de chargement de tables de faits. Ce système ne sera donc pas employé dans un flux de chargement d'une table de dimension pour lequel le processus de mise à jour est parfaitement acceptable.

- Il s'agit d'une proposition sur la façon de procéder pour réaliser une reprise de données. Il existe bien d'autres possibilités...
- Un bon système d'audit doit être adapté à votre contexte. Il doit être ni trop lourd, ni trop compliqué à mettre en place. Mais il doit cependant délivrer suffisamment d'informations.

Dans l'exemple qui va suivre, nous allons compléter le système par une table **AuditTraitement**, qui va nous permettre de suivre l'ensemble des exécutions de flux, flux de reprise inclus. Les tables d'audit seront internes à l'entrepôt de données DistrisysDW.

- Dans la base de données **DistrisysDW**, créez la table **AuditFlux** avec la structure ci-dessous, activez l'**incrémentation automatique** sur le champ **AuditFlux_PK** :

Nom de la colonne	Type de données	Autoriser l...
AuditFlux_PK	int	<input type="checkbox"/>
AuditTraitement_FK	int	<input type="checkbox"/>
NomFlux	varchar(50)	<input type="checkbox"/>
DateDebutFlux1ereEx...	smalldatetime	<input type="checkbox"/>
DateFinFlux1ereExec...	smalldatetime	<input checked="" type="checkbox"/>
DateDebutFluxDernier...	smalldatetime	<input type="checkbox"/>
DateFinFluxDerniereE...	smalldatetime	<input checked="" type="checkbox"/>
DateDebutPlageDonn...	smalldatetime	<input type="checkbox"/>
DateFinPlageDonnees	smalldatetime	<input type="checkbox"/>
NbErreurTechnique	int	<input checked="" type="checkbox"/>
NbAvertissement	int	<input checked="" type="checkbox"/>
FluxTermineAvecSuccess	char(1)	<input checked="" type="checkbox"/>
FluxAreprendre	char(1)	<input type="checkbox"/>
NbExecution	int	<input type="checkbox"/>

Les champs **DateDebutPlageDonnees** et **DateFinPlageDonnees** encadreront l'extraction des données source et fixeront donc le périmètre du flux.

Les champs **DateDebutFlux1ereExecution** et **DateFinFlux1ereExecution** déterminent la date de début et de fin lors de la première exécution. On pourra ainsi en déduire la durée du flux.

Le flux pouvant se reprendre, les champs **DateDebutFluxDerniereExecution** et **DateFinFluxDerniereExecution** déterminent la date de début et de fin lors de la dernière exécution du flux. Le champ **NbErreurTechnique** fait le bilan du nombre d'erreurs techniques survenues lors de la dernière exécution du flux.

Le champ **NbAvertissement** fait le bilan du nombre d'avertissements survenus lors de la dernière exécution du flux. Un avertissement identifiant un évènement de vigilance que le flux a su gérer.

En fin d'exécution du flux, au moment du bilan, c'est le champ **FluxAreprendre** qui déterminera si ce flux sera rejoué lors de la prochaine exécution. La valeur de ce champ sera déterminée par les évènements générés dans la table **AuditEvenement**, au cours de l'exécution du flux.

Le champ **NbExecution** indiquera le nombre de fois que le flux a été joué. Les flux pouvant se reprendre, un même flux peut donc être exécuté plusieurs fois. L'administrateur exploitant SSIS pourra se baser sur la valeur de ce champ pour déterminer si c'est encore utile de rejouer encore et encore le flux sur cette même plage de données.

Par exemple, un flux identifiant des avertissements depuis plus de 14 mois et joué plus de 40 fois mérite-t-il d'être de nouveau rejoué ? Dans ce cas, peut-être que les fonctionnels, n'ayant pas pris la peine de corriger le problème, considèrent que le problème n'en vaut pas la peine... Dans tous les cas un flux ne pourra pas être repris indéfiniment, soit il faudra corriger le problème, soit le problème sera considéré comme mineur et l'avertissement ignoré.

Puis créez la table **AuditEvenement** avec la structure ci-dessous et **activez l'incrémentation automatique** sur le champ **AuditEvenement_PK** :

	Nom de la colonne	Type de données	Autoriser l...
PK	AuditEvenement_PK	int	<input type="checkbox"/>
	AuditFlux_FK	int	<input type="checkbox"/>
	Evenement	varchar(50)	<input checked="" type="checkbox"/>
	EvenementType	varchar(50)	<input checked="" type="checkbox"/>
	TacheConcerne	varchar(50)	<input checked="" type="checkbox"/>
	ChampConcerne	varchar(50)	<input checked="" type="checkbox"/>
	ValeurEnErreur	varchar(50)	<input checked="" type="checkbox"/>
	CodeErreur	varchar(100)	<input checked="" type="checkbox"/>
	ErreurTechnique	char(1)	<input type="checkbox"/>
	Avertissement	char(1)	<input checked="" type="checkbox"/>
	FluxAreprendre	char(1)	<input type="checkbox"/>
	NbLignesComptabilisees	int	<input checked="" type="checkbox"/>

Les événements vont ponctuer le déroulement du flux pour réaliser la remontée d'informations.

En fait, il y a trois catégories d'évènements :

- Les évènements de **type comptage**. Ils comptabilisent le nombre de lignes. On comptera notamment le nombre de lignes extraites et le nombre de lignes chargées.
- Les évènements de **type erreur technique**. Ils remontent le code et la description technique de l'erreur due à une défaillance du flux.
- Les évènements de **type avertissement**. Ils correspondent aux alertes remontées par les points de vigilance. Les points de vigilance sont mis en œuvre au moment du développement du flux. Ces points de vigilance peuvent être techniques (non-correspondance d'un code), mais aussi fonctionnels (le CA facturé ne correspondant pas à celui de la comptabilité, la quantité en stock est négative, le coût est supérieur au prix de vente)

Les champs principaux sont :

- **Evènement** : nom de l'évènement. Ce nom est spécifique au flux. Dans notre cas, il s'agit de Nb Lignes Chargées FactFacture, Client Inconnu, Produit Inconnu
- **Evènement Type** : ce sera au concepteur SSIS de faire la liste des types d'évènements qu'il souhaite référencer. Dans notre cas il s'agit de Nb Lignes Source, Nb Lignes Destination, Avertissement et Erreur Technique.
- **Tâche concernée** : nom de la tâche sur lequel est intervenu l'évènement.
- **Champ concerné** : en cas d'avertissement, il s'agit du champ concerné par l'alerte.
- **Valeur en erreur** : en cas d'avertissement, il s'agit de la valeur du champ qui a généré l'alerte.

- **Code erreur** : correspond au message d'erreur technique généré par SSIS en cas de plantage.
- **Erreur Technique** : O pour oui et N pour Non. Permet d'identifier l'évènement comme une erreur technique.
- **Avertissement** : O pour oui et N pour Non. Permet d'identifier l'évènement comme un avertissement.
- **Flux à reprendre** : O pour oui et N pour Non. Permet d'identifier si l'évènement généré nécessitera une reprise du flux courant sur la même plage de donnée.
- **Nb Lignes Comptabilisées** : spécifie le nombre de lignes comptées (uniquement en cas d'évènement de comptage).

La comptabilisation des lignes en entrée et en sortie est considérée comme un évènement.

En effet dans un flux il y a potentiellement plusieurs sources de données, mais aussi potentiellement plusieurs destinations. C'est pour ce la que ces champs ne peuvent être spécifiés au niveau de la table AuditFlux.

Créez enfin la table **AuditTraitement** avec la structure ci-dessous et **activez l'incrémentation automatique** sur le champ **AuditTraitement_PK** :

Cette table fera simplement le bilan de toutes les exécutions de flux.

Enfin, pour mettre en place notre système d'audit, il est nécessaire de mettre à jour les tables de faits et de dimensions alimentées par un flux.

Au niveau de chaque table de dimension, ajoutez deux nouvelles colonnes d'audit :

- **AuditFluxAjout_FK** : pour identifier le flux qui a ajouté la ligne.
- **AuditFluxModification_FK** : pour identifier le flux qui a modifié la ligne pour la dernière fois.

Par exemple, pour DimProduit vous devriez avoir la structure suivante :

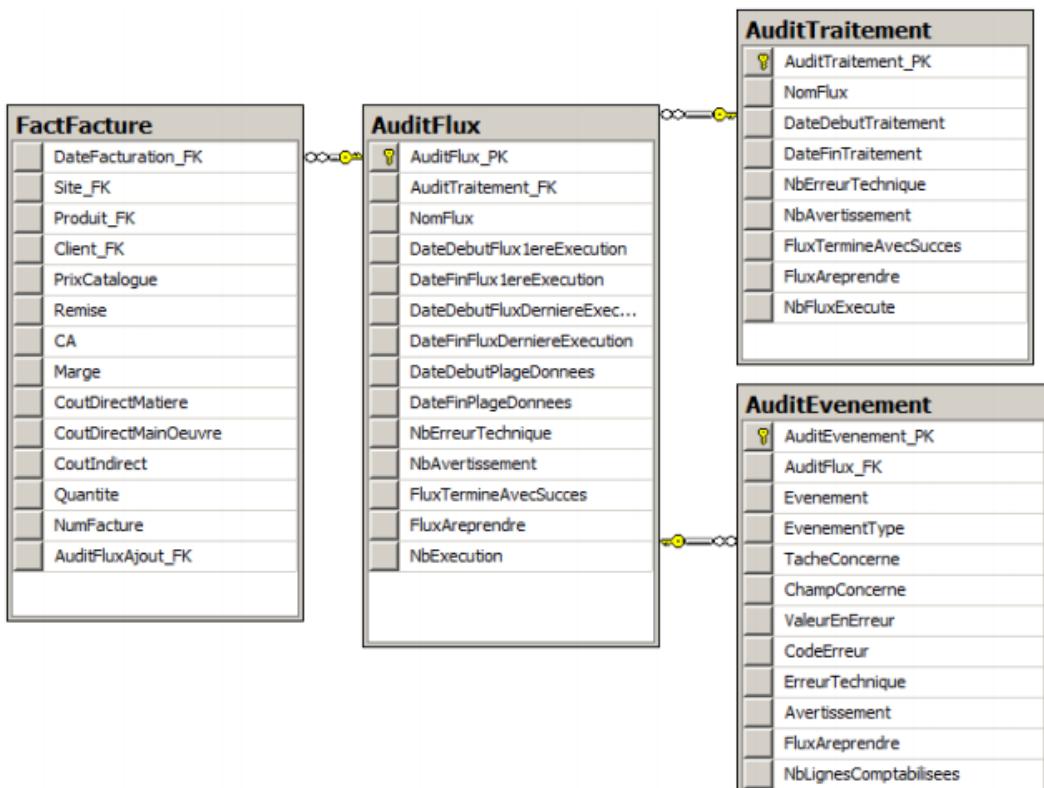
	Nom de la colonne	Type de données	Autoriser l...
PK	Produit_PK	int	<input type="checkbox"/>
	ProduitCode	varchar(10)	<input type="checkbox"/>
	Produit	varchar(20)	<input type="checkbox"/>
	SousFamilleCode	varchar(10)	<input type="checkbox"/>
	SousFamille	varchar(20)	<input type="checkbox"/>
	FamilleCode	varchar(10)	<input type="checkbox"/>
	Famille	varchar(20)	<input type="checkbox"/>
	Valide	bit	<input checked="" type="checkbox"/>
	AuditFluxAjout_FK	int	<input checked="" type="checkbox"/>
	AuditFluxModification_FK	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

De même, au niveau de chaque table de faits, ajoutez une nouvelle colonne d'audit :

AuditFluxAjout_FK, pour identifier le flux qui a ajouté la ligne.

- Ajoutez le champ **AuditFluxAjout_FK** aux tables FactFacture et FactFactureEntete.
- Créez les contraintes d'intégrité au niveau de FactFacture comme ci-dessous en créant les liens suivants :
 - entre AuditFlux **AuditFlux_PK** et FactFacture **AuditFluxAjout_FK**.
 - entre AuditTraitement **AuditTraitement_PK** et AuditFlux **AuditTraitement_FK**.

- entre AuditFlux **AuditFlux_PK** et AuditEvenement **AuditFlux_FK**.



- Créez ensuite les contraintes d'intégrité suivantes pour DimProduit :
- entre AuditFlux **AuditFlux_PK** et DimProduit **AuditFluxAjout_FK**.
- entre AuditFlux **AuditFlux_PK** et DimProduit **AuditFluxModification_FK**.

Enfin, nous ajouterons un membre inconnu dans chaque dimension. Idéalement, il faudrait que la clé technique soit toujours la même pour en faciliter la gestion.

Dans notre cas, la clé technique du membre inconnu sera **0**.

Ajoutez une nouvelle ligne dans chaque dimension avec pour **clé technique 0** et comme nom **inconnu**.

Par exemple au niveau de la table DimClient :

	Client_PK	GeographieClient_FK	ClientCode	Client	TypeClient	SegmentationClient
0	0		INC	Inconnu	Inconnu	Inconnu
1	1		C1	LaBoutiqueOnLine.com	Site Marchand	Bon Client
2	3		C2	Maison Discount	Discounteur	Bon Client
3	8		C3	Cuisine du sud	Spécialiste	Tiède
4	4		C4	Discount plus	Discounteur	Tiède
5	2		C5	EquiperSaMaison.com	Site Marchand	Très Bon Client
6	3		C6	Hypermarché Youpi	Grande surface	Très Bon Client
7	10		C7	EineKüche	Spécialiste	Bon Client
8	11		C8	Mercado Del Sol	Grande surface	Bon Client
9	1		C9	ElectroYoupa	Spécialiste	Bon Client
10	5		C10	Toutmoinscher.com	Site Marchand	Tiède

En fait, le système ne rejette aucune ligne. Or, il arrive régulièrement qu'un code client, notamment, ne soit pas identifié par le système : soit à cause d'une mauvaise saisie, soit parce que la réPLICATION des bases du CRM a eu une défaillance l'avant-veille...

Toujours est-il que même si ce code est inconnu par notre système, il faudra tout de même qu'il accepte la ligne. Pour cela nous utiliserons **le schéma de principe de l'erreur contrôlée** :

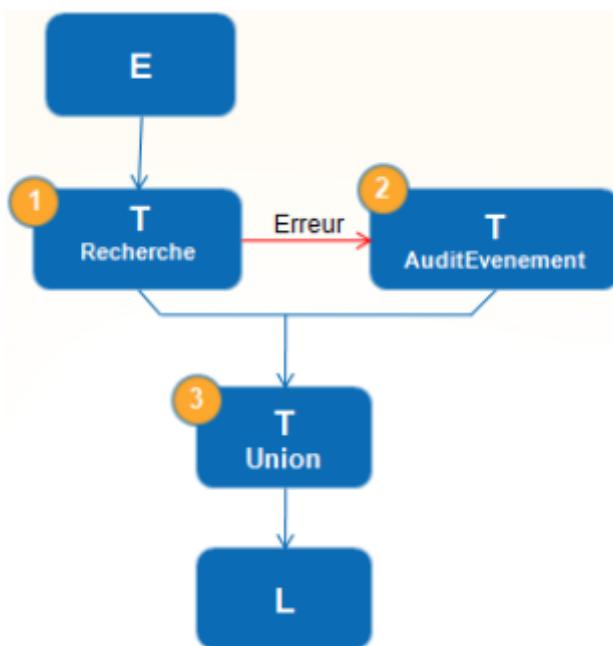


Schéma de principe de l'erreur contrôlée

La tâche 1 de type **Recherche** redirige les non-correspondances de code vers la tache 2.

La tâche 2 est un ensemble de tâches qui va :

- Ajouter une nouvelle ligne dans la table **AuditEvenement** identifiant le champ et la valeur en erreur.

Par exemple, le champ peut être ClientCode et la valeur sans correspondance C11.

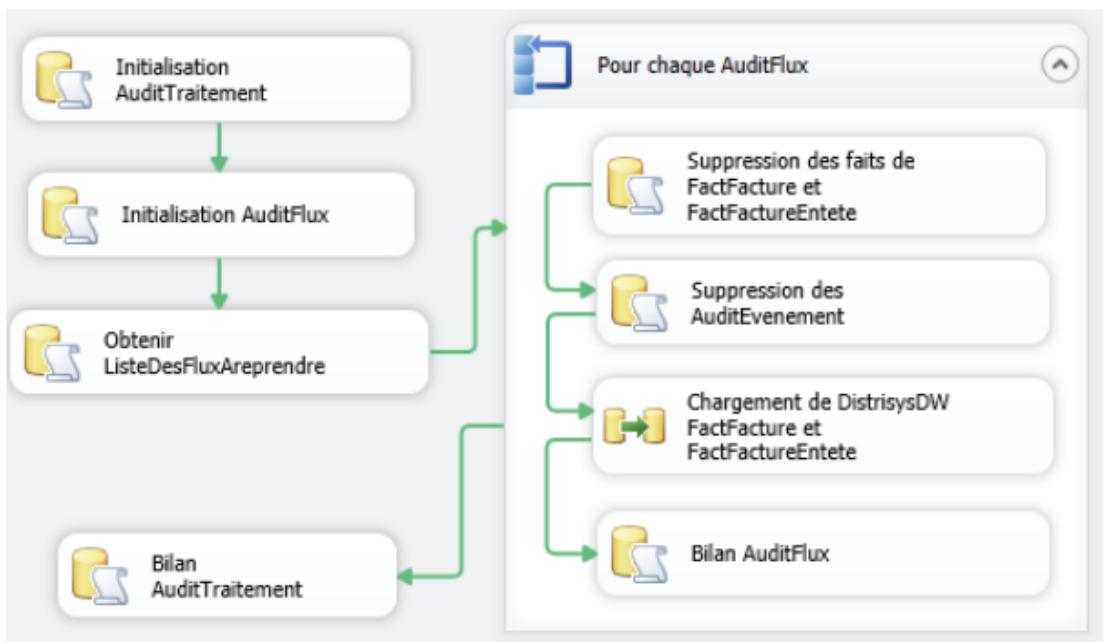
- Attribuer par défaut la clé 0(membre inconnu) aux lignes avec code sans correspondance. La tâche 3, d'union, va réconcilier les lignes avec correspondance et les lignes avec membre inconnu.

Dans la partie suivante nous allons étudier le flux de chargement intégrant cette fois-ci le système d'audit.

4-3 - Exemple de flux avec audit

Le flux de chargement des factures a été amélioré afin d'intégrer le système d'audit. Le package **DW_FactFacture_Avec_Audit.dtsx**, présenté dans ce chapitre est disponible dans la solution distrisys ETL en téléchargement sur le site des Éditions ENI.

Étudions tout d'abord le flux de contrôle ci-dessous :



Exemple de flux intégrant l'audit de flux

Ce flux de contrôle s'accompagne de la création de nouvelles variables :

Nom	Étendue	Type de donn...	Valeur	Expression
AuditFlux_FK	DW_FactFactu...	Int32	0	
AuditTraitement...	DW_FactFactu...	Int32	0	
DateDebutPlage...	DW_FactFactu...	DateTime	01/03/2012	
DateFinPlageDo...	DW_FactFactu...	DateTime	01/04/2012	
ListeFluxArepren...	DW_FactFactu...	Object	System.Object	
NomFlux	DW_FactFactu...	String	DW_Chargement Facture	

La tâche **Initialisation AuditTraitement** va insérer une nouvelle ligne dans la table AuditTraitement et va récupérer la valeur **AuditTraitement_FK** de la ligne générée dans la variable du même nom.

Pour obtenir ce comportement, configurez la propriété **SQL Statement** comme ci-dessous :

```

INSERT INTO [AuditTraitement] ([NomFlux], [DateDebutTraitement])
VALUES (?, GETDATE())
SELECT cast(SCOPE_IDENTITY() as int) AS AuditTraitementID
    
```

Le ResultSet doit être positionné sur Ligne Unique.

Dans l'onglet **Mappage des paramètres**, la variable **User::NomFlux** de direction Input et de type Varchar, est mappé avec le nom de paramètre 0.

Dans l'onglet **Jeu de résultats**, la variable **User::AuditTraitement_FK** est mappée avec le **Nom de résultats** AuditTraitementID ; en référence au nom de la colonne ramenée par la clause SELECT de la requête.

La tâche **Initialisation AuditFlux** va simplement insérer une nouvelle ligne dans la table AuditFlux.

Sa propriété SQL Statement est la suivante :

```

INSERT INTO [AuditFlux]
([NomFlux], [AuditTraitement_FK], [DateDebutFluxLereExecution],
[DateDebutFluxDerniereExecution], [DateDebutPlageDonnees],
[DateFinPlageDonnees], [FluxAreprendre], [NbExecution])
VALUES
(?, ?, GETDATE(), GETDATE(), (SELECT
MAX([DateFinPlageDonnees]) FROM [AuditFlux] WHERE
[NomFlux]='DW_Chargement Facture')
,GetDate(), 'O', 0)

```

Cette nouvelle ligne a, comme date de début de plage de données, la valeur maximum de la plage de données de fin des flux de même nom et comme date de fin de plage de données, la date actuelle. Bien entendu, tout ceci se configure et s'affine suivant les situations.

Comme nous l'avons déjà vu, lors du flux de chargement du SA, la tâche **ObtenirListeDesFluxAReprendre** lit la table AuditFlux pour récupérer la liste des flux à reprendre. Cette liste est récupérée dans la variable ListeDesFluxAReprendre.

C'est cette même variable qui est parcourue par la tâche Pour chaque **AuditFlux**.

À chaque itération, les variables AuditFlux_FK, DateDebutPlageDonnees et DateFinPlageDonnees sont réinitialisées pour prendre la valeur du flux courant.

De ce fait, pour chaque flux parcouru par la boucle, deux tâches vont supprimer les faits et les lignes AuditEvenement précédemment chargés.

Puis la tâche de flux de données va lancer l'extraction des données de DistrisysSA, pour effectuer le chargement dans DistrisysDW.

Enfin, les tâches Bilan AuditFlux et Bilan AuditTraitement font une requête d'update dans leur table respective, afin de faire le récapitulatif de l'exécution du processus :

- Bilan AuditFlux fait le bilan à partir des données de la table AuditEvenement.
- Bilan AuditTraitement fait le bilan à partir des données de la table AuditFlux.

Au final, voici le contenu de la table AuditTraitement :

	AuditTraitement_PK	NomFlux	DateDebutTraitement	DateFinTraitement	NbErreurTechnique	NbAvertissement	FluxTermineAvecSuccess	FluxArepandre	NbFluxExecute
1	8	DW_Chargement Facture	2012-07-29 21:34:00	2012-07-29 21:34:00	0	1	0	0	11
2	9	DW_Chargement Facture	2012-07-29 21:35:00	2012-07-29 21:36:00	0	1	0	0	2
3	10	DW_Chargement Facture	2012-07-30 13:37:00	2012-07-30 13:37:00	0	1	0	0	2
4	11	DW_Chargement Facture	2012-07-30 14:06:00	2012-07-30 14:06:00	0	1	0	0	2
5	12	DW_Chargement Facture	2012-07-30 14:34:00	2012-07-30 14:34:00	0	1	0	0	2
6	13	DW_Chargement Facture	2012-07-30 14:35:00	2012-07-30 14:35:00	0	1	0	0	2
7	14	DW_Chargement Facture	2012-08-04 19:11:00	2012-08-04 19:12:00	0	0	0	N	2

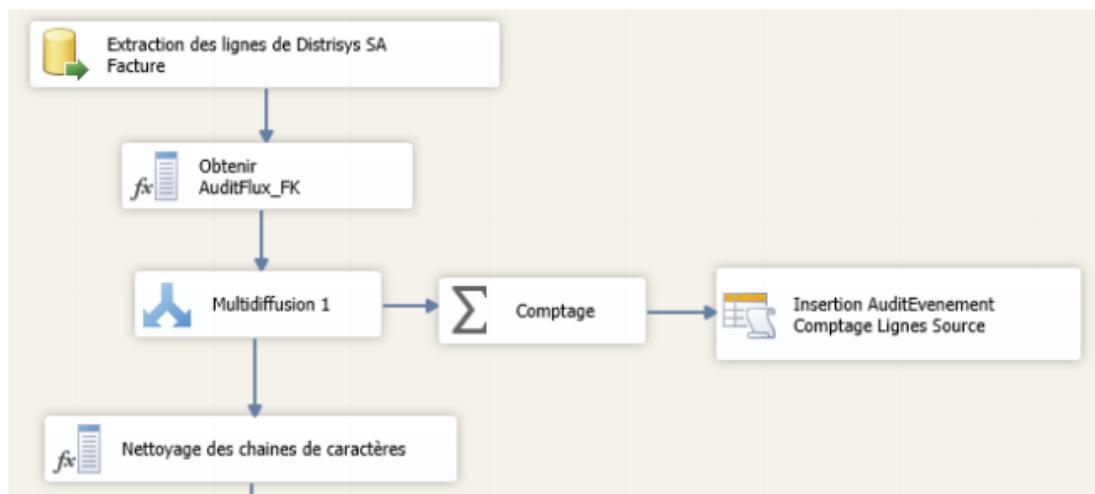
Et celui de la table **AuditFlux** :

AuditID	AuditType	NomFlux	DateDebutR	DateFinFlux1...	DateDebutR...	DateDebutR...	DateDebutP...	DateFinPlag...	NbErreurTechniq...	NbAvertissement	RésuméFlux	RésuméFlux	NbExecution
1	16	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-01 ...	2012-08-01 ...	0	0	O	N	4
2	17	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-08-01 ...	2012-08-01 ...	0	0	O	N	4
3	18	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-09-01 ...	2012-09-03 ...	0	0	O	N	5
4	19	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-09-03 ...	2012-09-29 ...	0	0	O	N	5
5	20	14	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-08-04 ...	2012-09-29 ...	2012-10-29 ...	0	0	O	13
6	21	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-10-29 ...	2012-11-29 ...	0	0	O	4
7	22	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-11-29 ...	2012-12-29 ...	0	0	O	4
8	23	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-12-29 ...	2012-12-29 ...	0	0	O	4
9	25	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2013-01-29 ...	2013-01-29 ...	0	0	O	3
10	26	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2013-01-29 ...	2013-03-29 ...	0	0	O	2
11	27	8	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2013-03-29 ...	2013-04-29 ...	0	0	O	1
12	28	9	DW_Changement Facture	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2012-07-29 ...	2013-04-29 ...	2013-05-29 ...	0	0	O	1
13	29	10	DW_Changement Facture	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2013-06-30 ...	2013-06-30 ...	0	0	O	1
14	30	11	DW_Changement Facture	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2013-06-30 ...	2013-06-30 ...	0	0	O	1
15	31	12	DW_Changement Facture	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2011-01-30 ...	2011-02-28 ...	0	0	O	1
16	32	13	DW_Changement Facture	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2012-07-30 ...	2011-02-28 ...	2011-03-30 ...	0	0	O	1
17	33	14	DW_Changement Facture	2012-07-04 ...	2012-08-04 ...	2012-08-04 ...	2012-08-04 ...	2011-03-30 ...	2013-06-04 ...	0	0	O	1

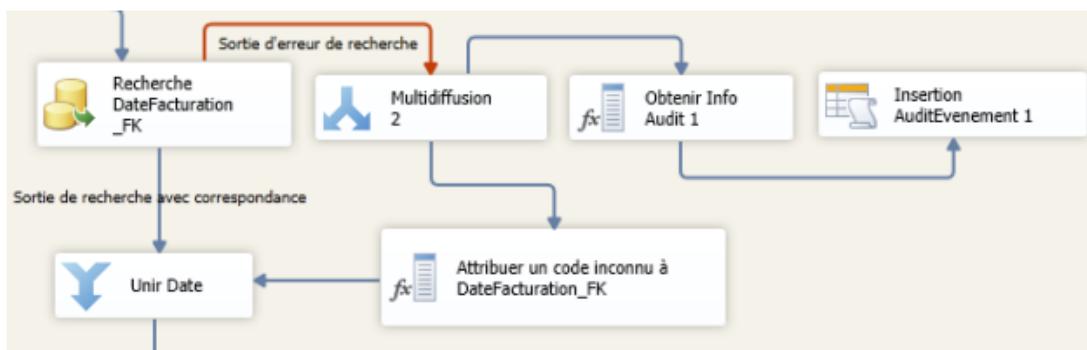
Étudions maintenant le contenu du flux de données.

Nous avons simplement ajouté de nombreux points de vigilance, afin de faire des remontées d'informations dans la table **AuditEvenement**.

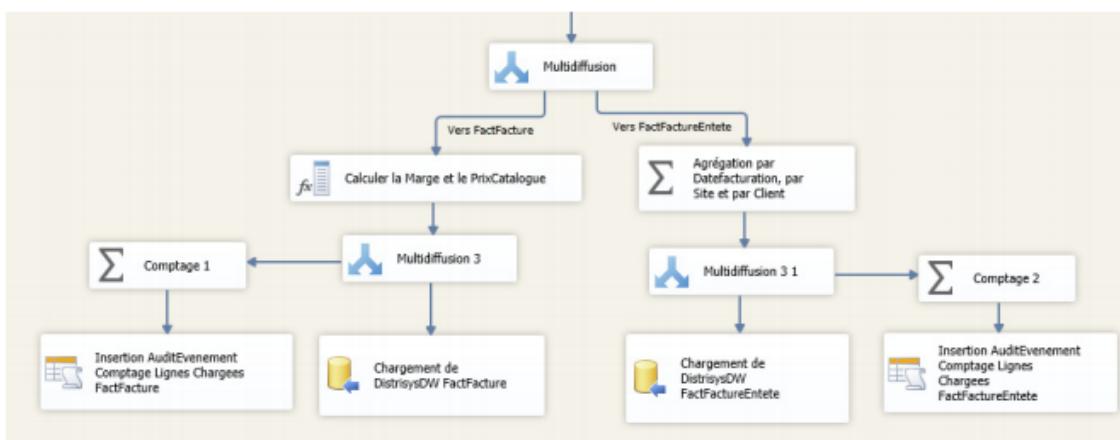
La comptabilisation des lignes extraites va se faire avec les tâches ci-dessous :



Le schéma de principe de l'erreur contrôlée est mis en pratique par les tâches ci-dessous :



Enfin, la comptabilisation des lignes chargées dans la table FactFacture est réalisée par les tâches ci-dessous :



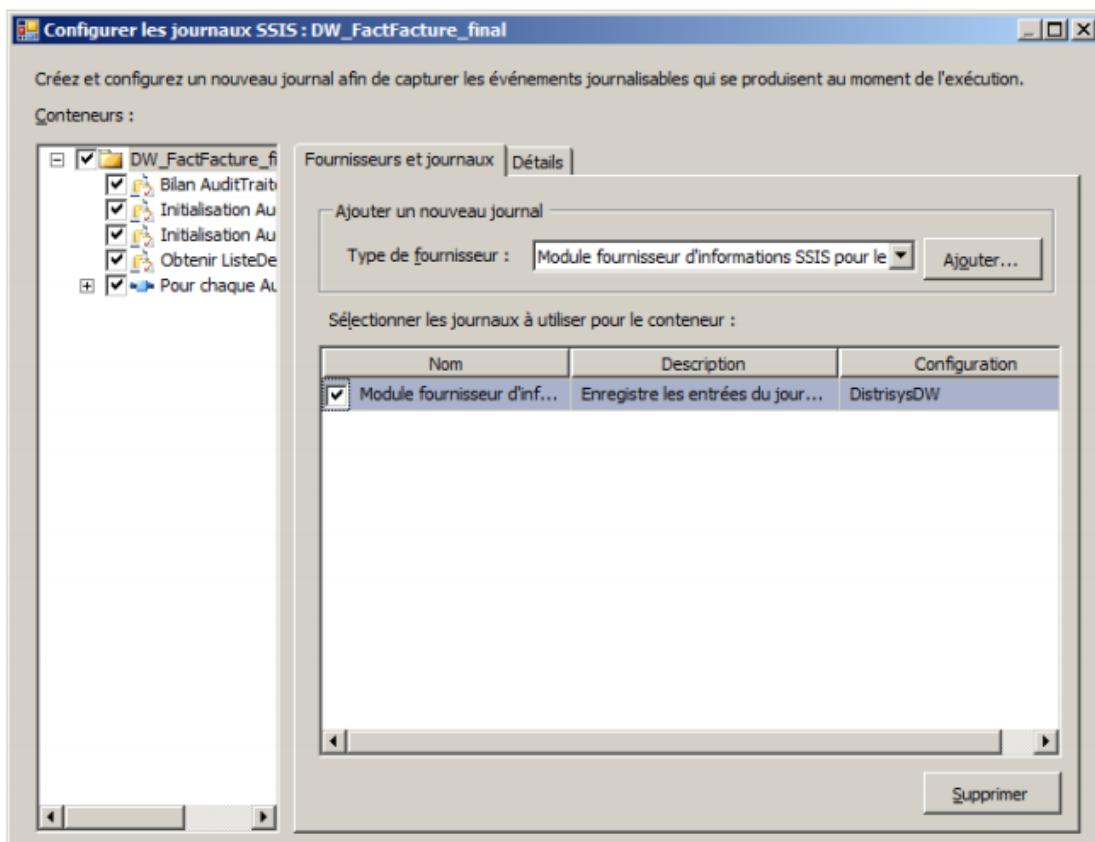
La table **AuditEvenement** ainsi chargée est la suivante :

	Audi...	Au...	Evenement	EvenementT...	TacheCo...	ChampC...	Vale...	CodeErreur	Er...	Avertisse...	RuxAprendre	NbLignesComptabilisees
26	183	27	Nb Lignes chargées FactFacture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	92
27	184	27	Nb Lignes chargées FactFactureEntete	Comptage U...	Comptage	NULL	NULL	NULL	N	N	N	90
28	189	28	Nb Lignes sources Facture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	89
29	190	28	Nb Lignes chargées FactFacture	Comptage U...	Comptage	NULL	NULL	NULL	N	N	N	89
30	191	28	Nb Lignes chargées FactFactureEntete	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	86
31	196	29	Nb Lignes sources Facture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	95
32	197	29	Nb Lignes chargées FactFacture	Comptage U...	Comptage	NULL	NULL	NULL	N	N	N	95
33	198	29	Nb Lignes chargées FactFactureEntete	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	91
34	203	30	Nb Lignes sources Facture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	44
35	204	30	Nb Lignes chargées FactFactureEntete	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	41
36	205	30	Nb Lignes chargées FactFacture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	44
37	210	20	Nb Lignes sources Facture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	90
38	211	20	Client inconnu	Avertissement	Recherc...	C11	NULL	N	O	O	NULL	NULL
39	212	20	Nb Lignes chargées FactFacture	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	90
40	213	20	Nb Lignes chargées FactFactureEntete	Comptage Li...	Comptage	NULL	NULL	NULL	N	N	N	86

Nous venons ainsi de réaliser un flux de chargement des factures intégrant un système d'audit et de reprise de données en automatique.

Malgré le système d'audit, il est conseillé d'activer la gestion des logs de SSIS. Si le système d'audit vous fournit énormément d'informations structurées, la gestion de logs SSIS peut vous fournir des informations complémentaires plus techniques.

- Pour activer la gestion des logs, cliquez sur **SSIS** dans la barre de menu, puis sur **Journalisation**.
- Dans le **type de fournisseurs**, sélectionnez **Mode fournisseur d'informations pour SQL Server** puis cliquez sur **Ajouter** afin d'écrire les logs directement dans une table. L'information sera ainsi plus simple à retrouver et à exploiter.
- Puis, dans la colonne Configuration, sélectionnez la connexion **DistrisysDW**.
- Sélectionnez tous les **conteneurs** en cochant DW_FactFacture_avec_audit dans la fenêtre Conteneurs et dans l'onglet Détails, sélectionnez **tous les types d'évènements**.
- Enfin, dans Sélectionner les journaux à utiliser pour le conteneur, cochez la case **module fournisseur**.
- Puis terminez en cliquant sur **OK**.



Écran de configuration des journaux SSIS

Une nouvelle table **sysssislogs**, identifiée comme table système, s'ajoute à la base de données DistrisysDW.

Ses principaux champs enregistrés par les logs sont les suivants :

- **Event** : référence le type d'évènement à l'origine de l'entrée de log (OnPreValidate, OnInformation, OnProgress...).
- **Computer** : le nom du serveur qui a exécuté le flux.
- **Operator** : le compte de service qui a exécuté le flux.
- **Source** : nom de la tâche à l'origine de la ligne de log.
- **Sourcelid** : identifiant technique SSIS de la tâche à l'origine de la ligne de log.
- **ExecutionId** : identifiant technique du traitement.
- **StartTime** : date de début de l'évènement.
- **EndTime** : date de fin de l'évènement.
- **Message** : décrit le résultat de l'évènement et affiche notamment en détail un message d'erreur.

La traçabilité de nos flux est ainsi complète. Néanmoins, nous n'avons fait que la moitié du travail. Il vous sera nécessaire, par la suite, de créer les rapports d'audit, mais aussi potentiellement les mesures et indicateurs de la performance de votre processus ETL, à présenter à vos utilisateurs. Le chapitre Restituer les données décisionnelles - Reporting Services traite de la réalisation de tels rapports, je vous suggère de vous y reporter pour plus de détail.

De plus, la réussite de ce système ne vaut que si les utilisateurs s'emploient au quotidien à corriger les erreurs que vous leur présentez. La qualité des rapports et des indicateurs que vous leur fournirez et leurs disponibilités au côté des données décisionnelles sont donc essentielles.

5 - Gestion des paramètres de flux et mise en production

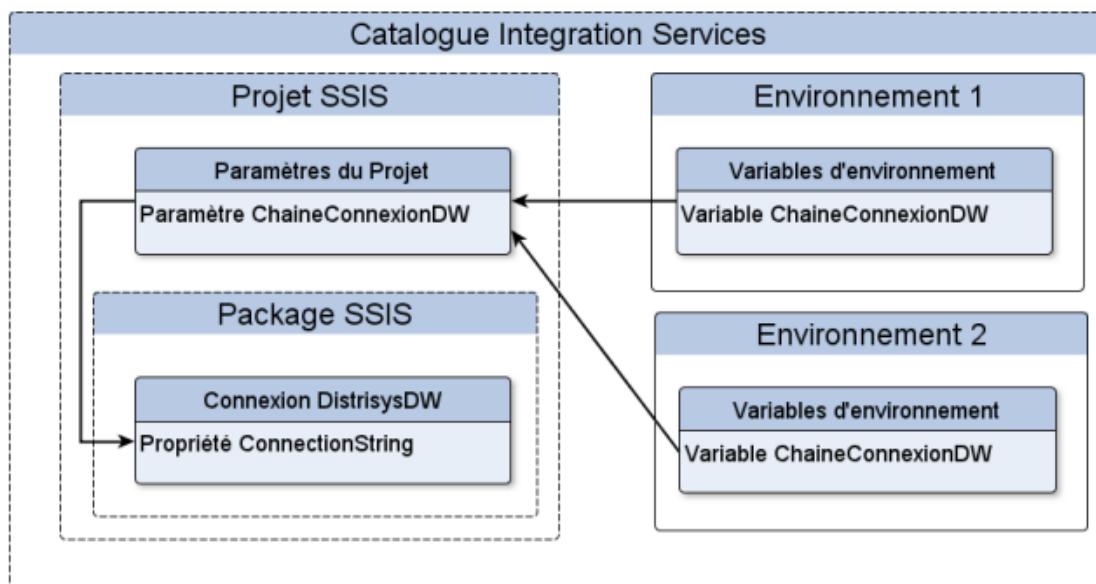
Dans les parties précédentes, vous avez appris à développer des flux pour charger une dimension, une table de faits ou bien des fichiers plats. À l'état actuel, vos flux fonctionnent bien tant que vous travaillez en local et seul.

Dans cette partie, nous allons aborder le paramétrage et la portabilité de vos flux sur des environnements différents (recette, production...).

Pour pouvoir porter les flux d'un environnement à un autre, nous allons devoir utiliser les paramètres du projet SSIS, qui vont nous permettre, par exemple, de variabiliser les chaînes de connexion du projet.

Ensuite, nous importerons notre projet SSIS dans un catalogue Integration Services, qui est en fait une base de données particulière du serveur SQL Server. Ce catalogue nous permettra notamment de déployer et exécuter nos packages.

Dans ce catalogue, nous définirons des environnements. Un environnement contient un ensemble de valeurs qui lui sont propres. Nous pourrons ensuite lier un projet SSIS à un environnement, et ainsi valoriser les paramètres du projet à partir des données de l'environnement. On peut lier un projet à plusieurs environnements. C'est à l'exécution des flux que l'on choisira l'environnement et donc les valeurs des paramètres que l'on veut utiliser pour cette exécution.



Principe de paramétrage et gestion des environnements dans SSIS

5-1 - Paramétrage des flux

Actuellement, les connexions du Gestionnaire de connexions sont définies en dur. Nous allons donc les configurer afin que leurs chaînes de connexion soient issues d'un paramètre pour lequel vous pourrez définir une valeur en fonction de l'environnement. Vous faciliterez ainsi la portabilité de vos flux. Pour cela, nous allons devoir créer deux paramètres :

- ChaineConnexionSA de type string contiendra la chaîne de connexion à la base DistrisysSA
- ChaineConnexionDW de type string contiendra la chaîne de connexion à la base DistrisysDW

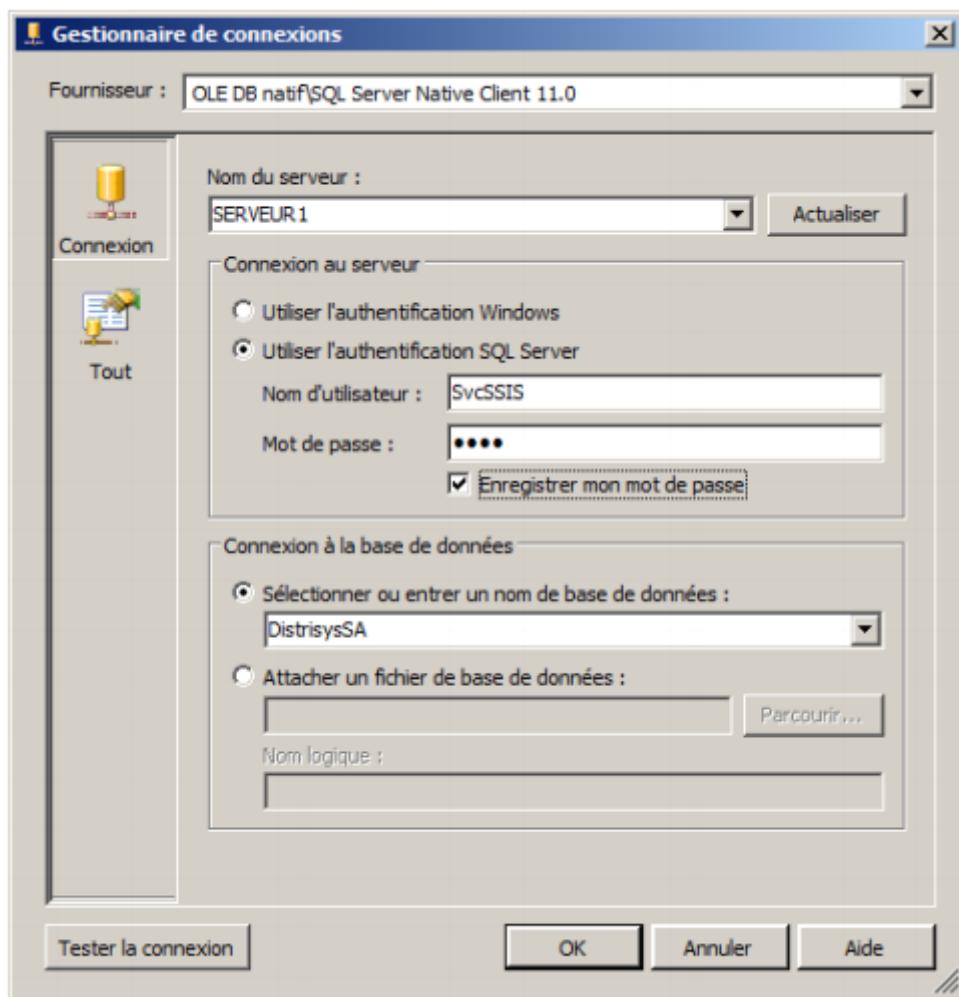
Si vous prévoyez l'exécution de vos flux de production avec l'agent SQL Server, vos flux seront exécutés avec le compte de service de l'agent SQL Server. Prévoyez donc de **donner les droits d'accès en lecture et en écriture aux bases de données** auxquelles ce compte devra accéder.

Si par contre vos flux doivent être exécutés par un ordonnanceur autre que l'agent SQL, il vous faudra intégrer à la chaîne de connexion un login et mot de passe d'un compte SQL Server.

- Créez donc ce compte et attribuez-lui les droits d'accès aux bases de données auxquelles il devra accéder. Attention, toutefois de ne pas lui attribuer plus de droits que le compte n'en a besoin.

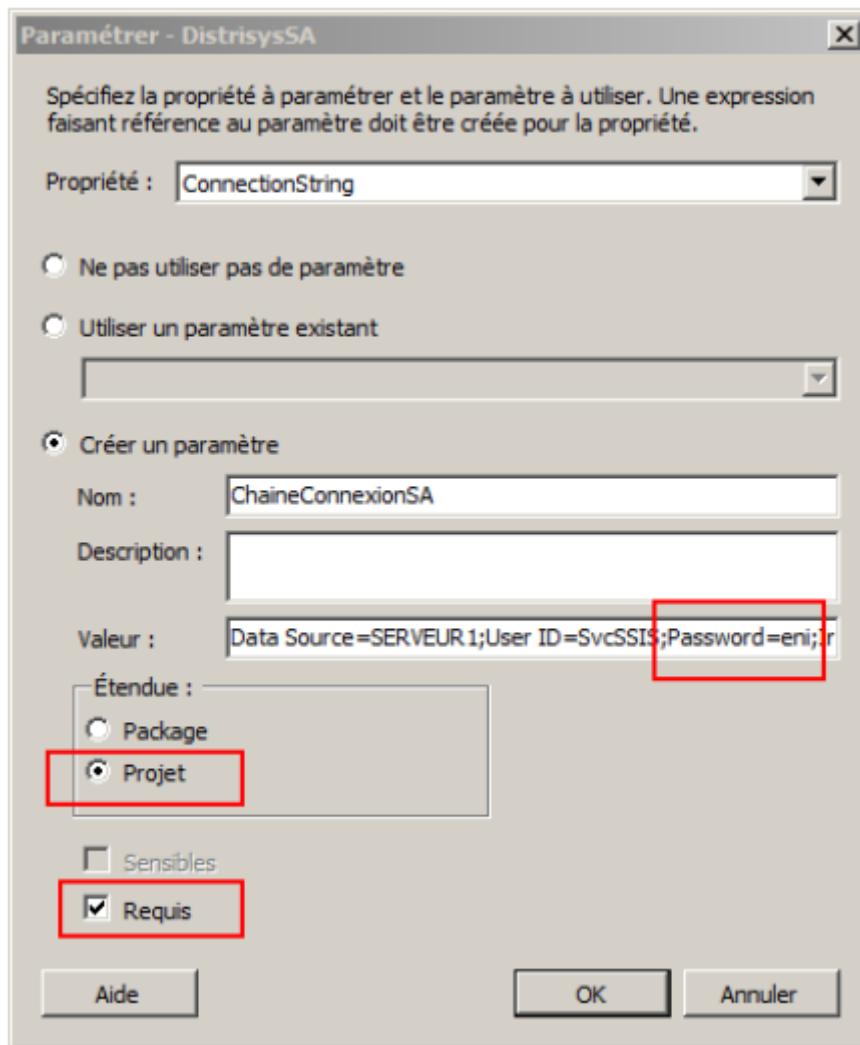
Créez donc ce compte et attribuez-lui les droits d'accès aux bases de données auxquelles il devra accéder. Attention, toutefois de ne pas lui attribuer plus de droits que le compte n'en a besoin.

- Modifiez la connexion **DistrisysSA** de votre flux si besoin puis fermez le Gestionnaire de connexions :



- Cliquez avec le bouton droit sur la connexion **DistrisysSA** et sélectionnez **Paramétrer** :
- Sélectionnez **Créer un paramètre** et renommez-le en **ChaineConnexionSA**.
- Sélectionnez **Projet** pour l'étendue du paramètre (il sera ainsi disponible pour tous les packages du projet) et **Requis** (le projet ne pourra pas être exécuté si le paramètre n'est pas renseigné).

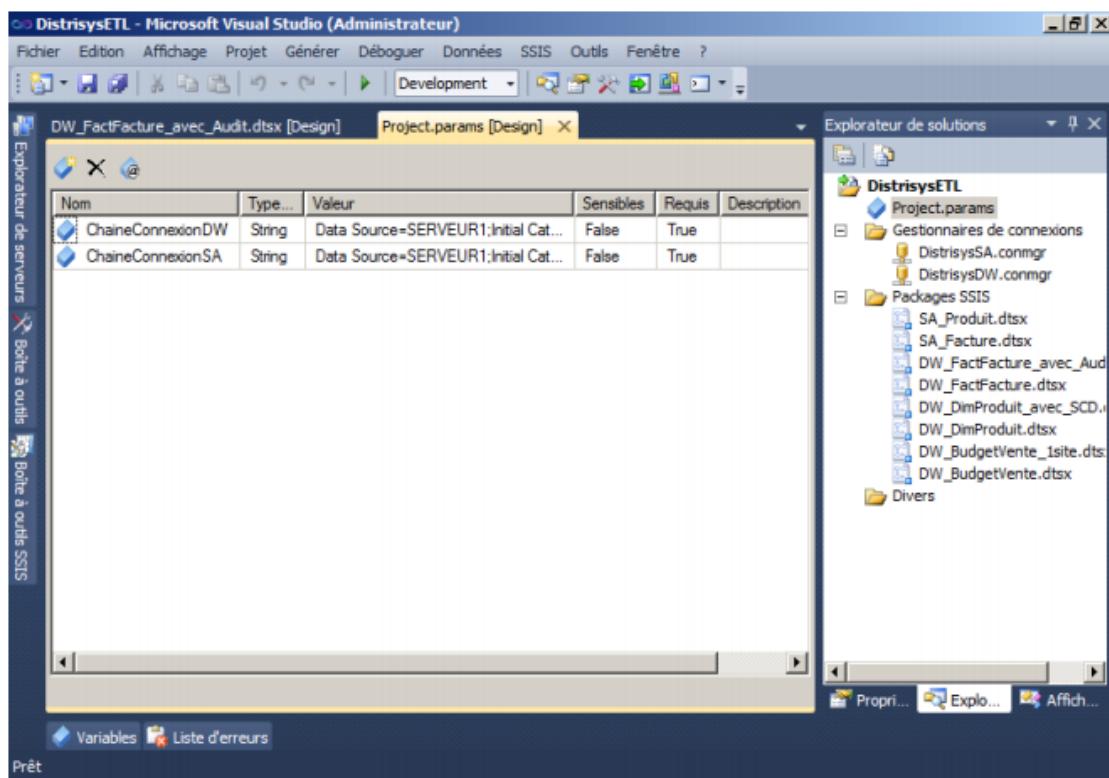
Puis ajoutez manuellement le mot de passe à la chaîne de connexion de la variable, comme indiqué ci-dessous :



Paramétrage d'une connexion

- Faites de même pour la connexion à **DistrisysDW** en créant le paramètre **ChaineCon-nexionDW**.

La liste des paramètres du projet est accessible depuis l'Explorateur de solutions, en double cliquant sur **Project.params** :



Paramètres du projet

À ce stade, vos connexions sont pilotées par le contenu des paramètres ChaineConnexionSA et ChaineConnexionDW.

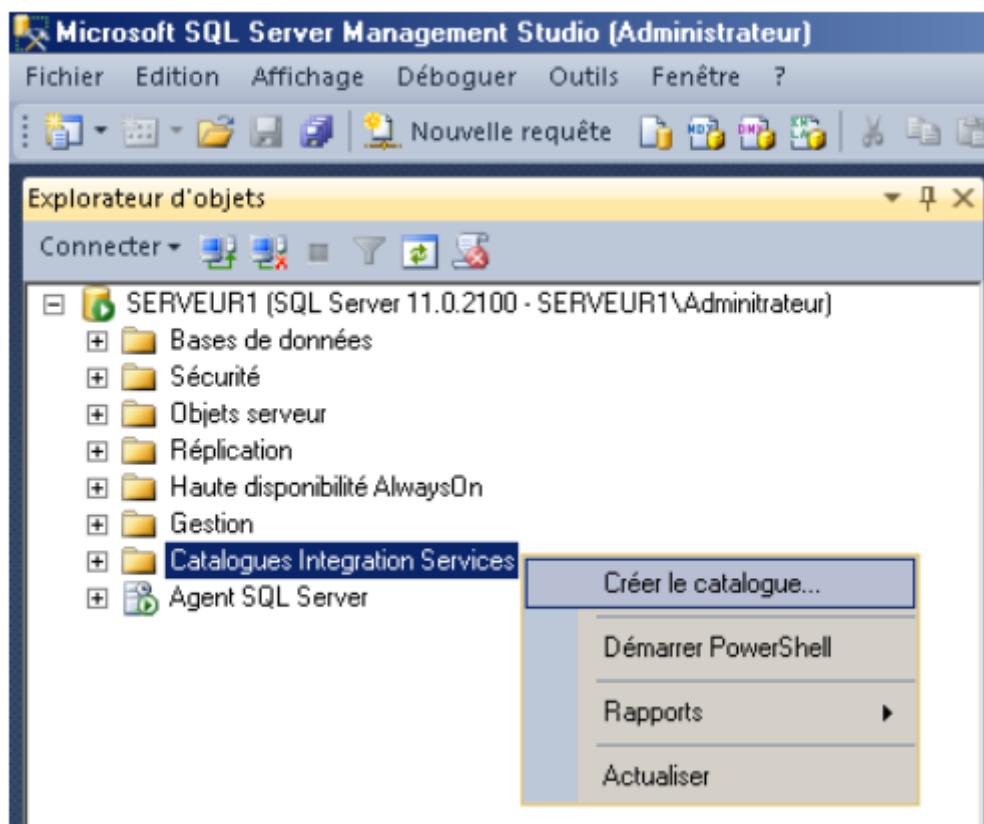
- Testez de nouveau votre flux pour vous assurer qu'il continue à bien fonctionner.

Mais les valeurs de vos paramètres sont toujours liées à votre serveur de développement. Avant de les lier à un environnement, nous devons déployer le projet SSIS dans un catalogue Integration Services sur le serveur de base de données.

5-2 - Création du catalogue Integration Services

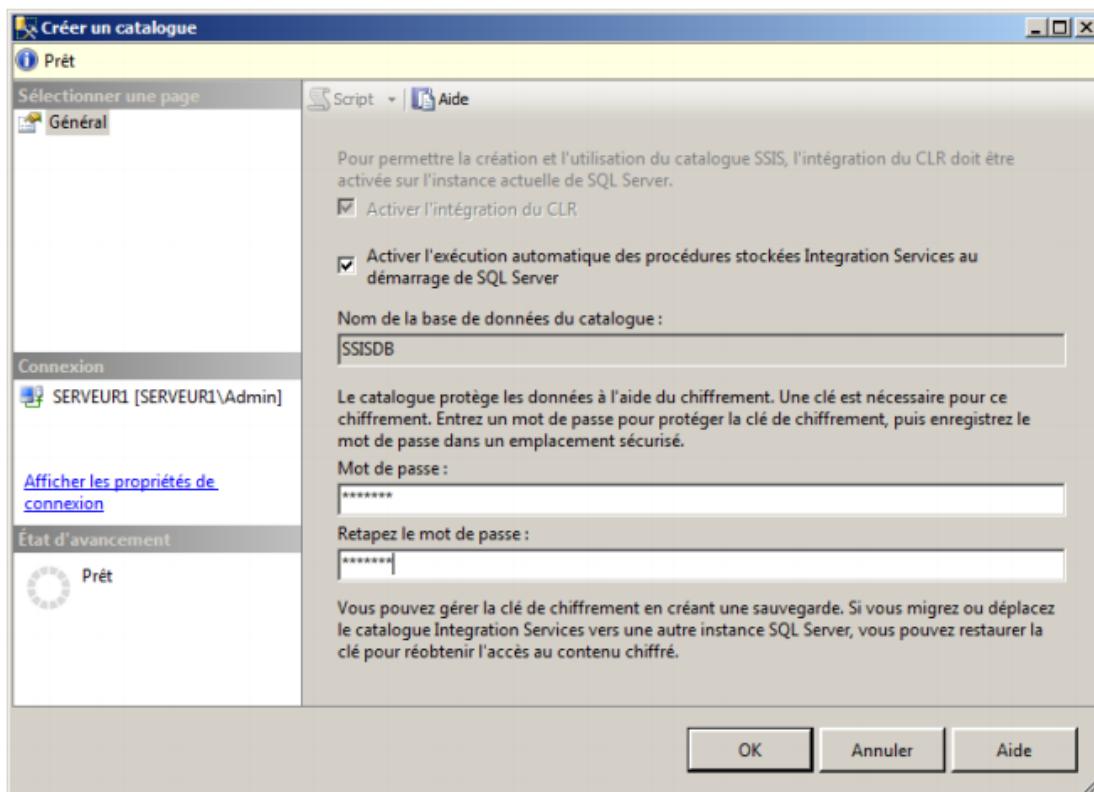
Vous devrez créer un catalogue Integration Services sur chaque serveur où vous déploierez vos projets SSIS.

Pour créer le catalogue, connectez-vous depuis SSMS au serveur de base de données, avec un compte Windows. Cliquez avec le bouton droit sur **Catalogues Integration Services** et sélectionnez l'option **Créer le catalogue**.



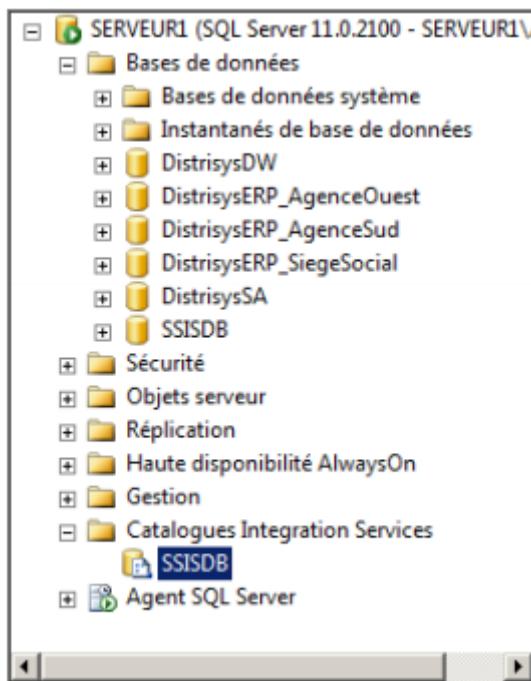
Création d'un catalogue

- Renseignez un mot de passe qui sera utilisé pour le cryptage des données, puis cliquez sur le bouton **OK**.



- On ne peut pas renommer le catalogue, ni sa base de données.

La base de données et le catalogue SSISDB sont maintenant créés.

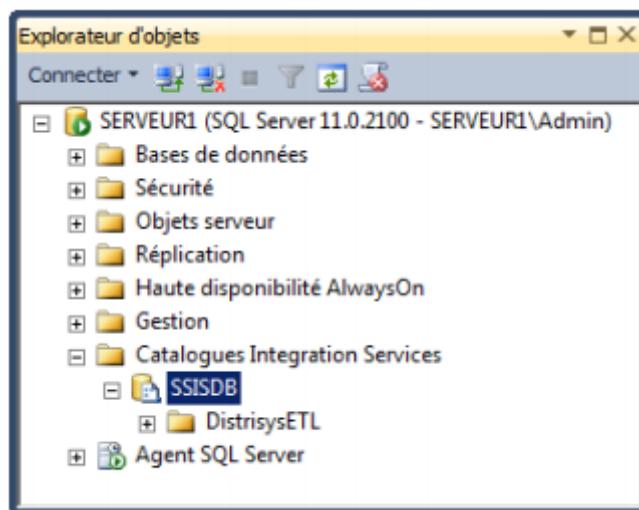


Le catalogue SSISDB et la base de données associée

- Vous n'utiliserez jamais la base de données directement, sauf pour des opérations de maintenance. C'est le catalogue Integration Services qui vous permettra de gérer vos projets.

Nous allons maintenant créer un répertoire dans le catalogue afin d'y déployer notre projet SSIS.

Cliquez avec le bouton droit sur le catalogue SSISDB et sélectionnez **Créer un dossier**. Créez le dossier **DistrisysETL**.



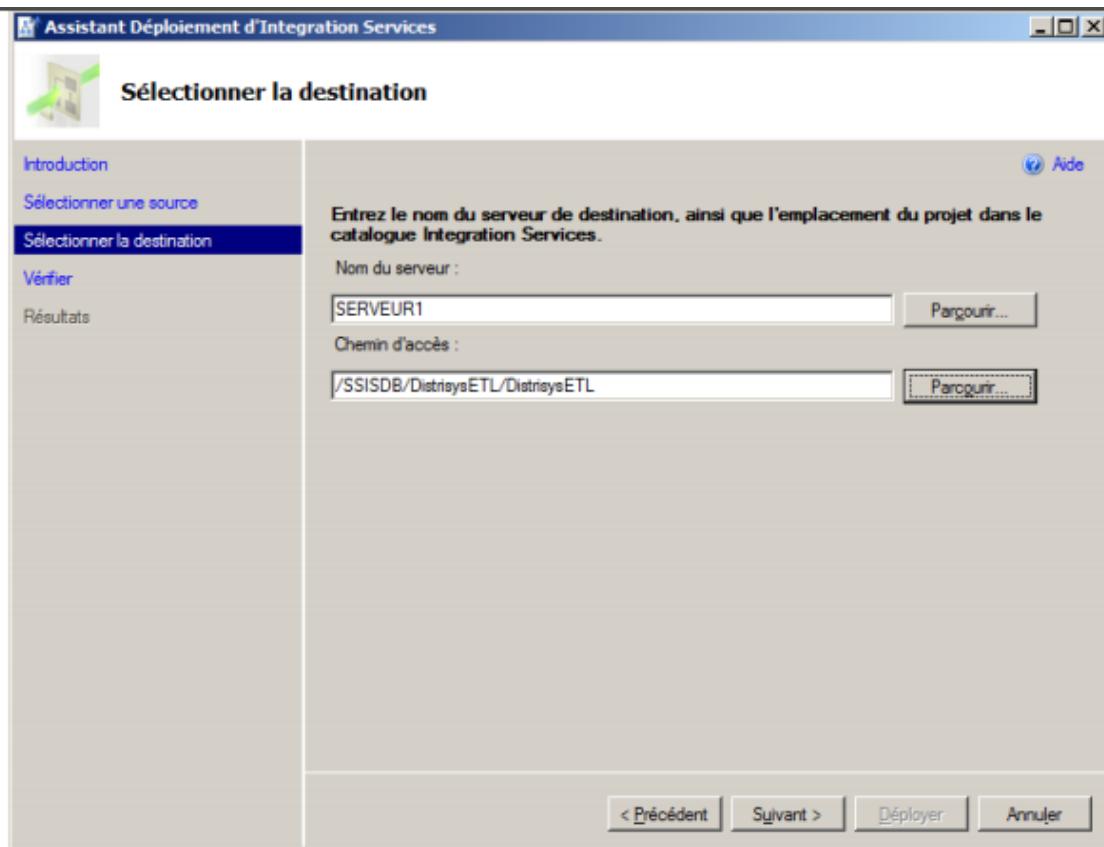
Création d'un répertoire dans le catalogue

Nous pouvons maintenant déployer notre projet SSIS **DistrisysETL** dans le catalogue que nous venons de créer.

5-3 - Déploiement du projet SSIS sur le serveur de développement

Le déploiement d'un projet SSIS dans un catalogue consiste à l'importer dans la base de données SSISDB du serveur. À partir de là, nous pourrons l'exécuter et surtout, valoriser ses paramètres.

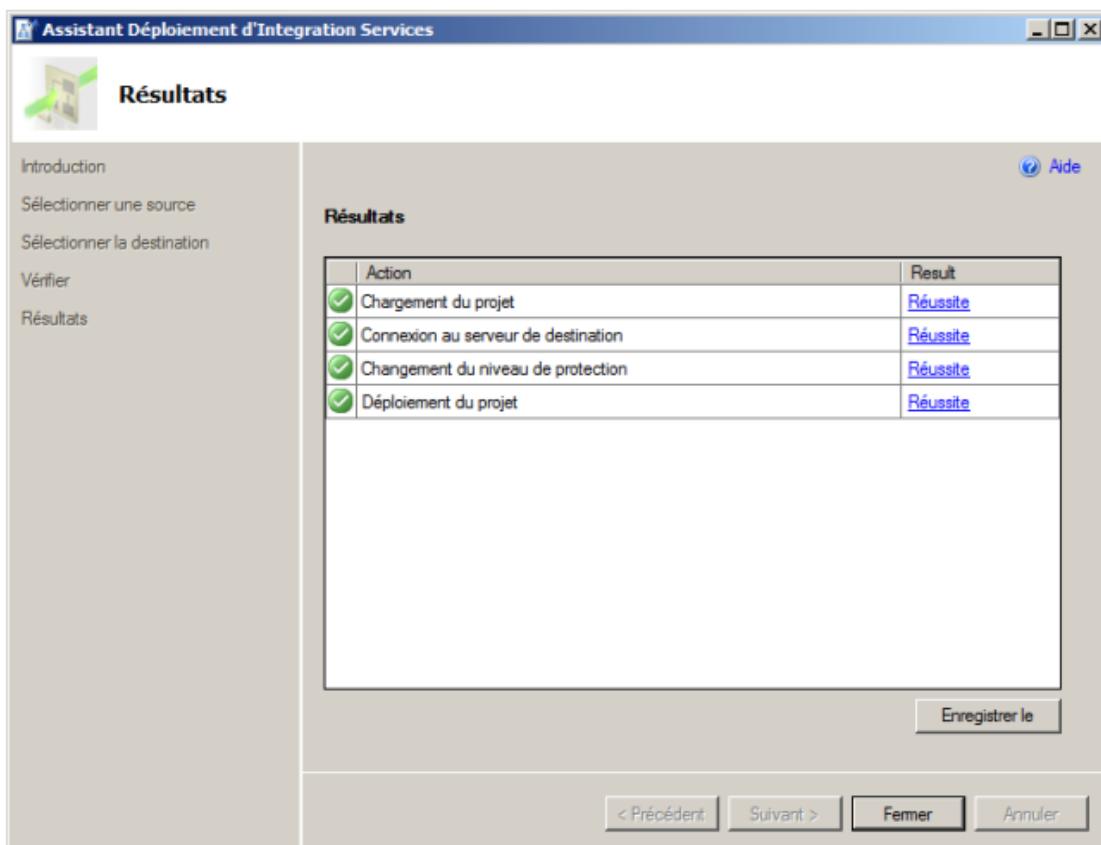
- Dans SSDT, ouvrez le projet **DistrisysETL**, sélectionnez dans le menu **Projet** l'option **Déployer**.
- L'assistant de déploiement s'ouvre. Cliquez sur **Suivant** pour passer la première fenêtre.
- La source est par défaut le projet courant.
- Dans l'écran de sélection de la destination, entrez le nom du serveur et sélectionnez le répertoire dans lequel vous voulez déployer votre projet :



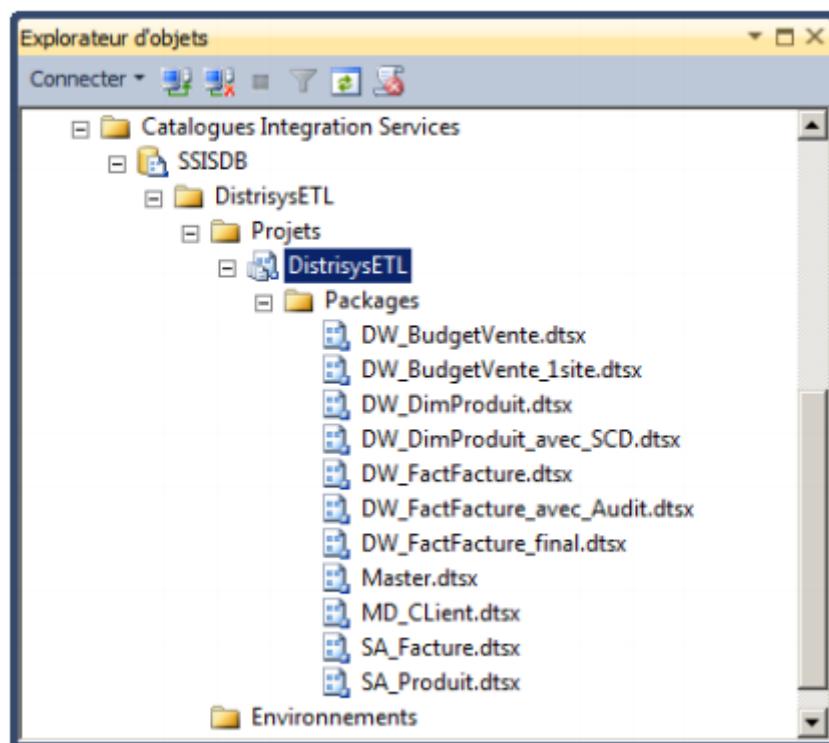
Déploiement d'un projet dans le catalogue

- Cliquez sur **Suivant** puis vérifiez vos sélections et cliquez sur **Déployer**.

Votre projet est maintenant disponible dans le catalogue SSISDB :



- Retournez dans SSMS. Dans le répertoire **DistrisysETL** du catalogue **SSISDB**, vous avez maintenant une nouvelle arborescence avec la liste des packages du projet, ainsi qu'un sous-répertoire **Environnements**.



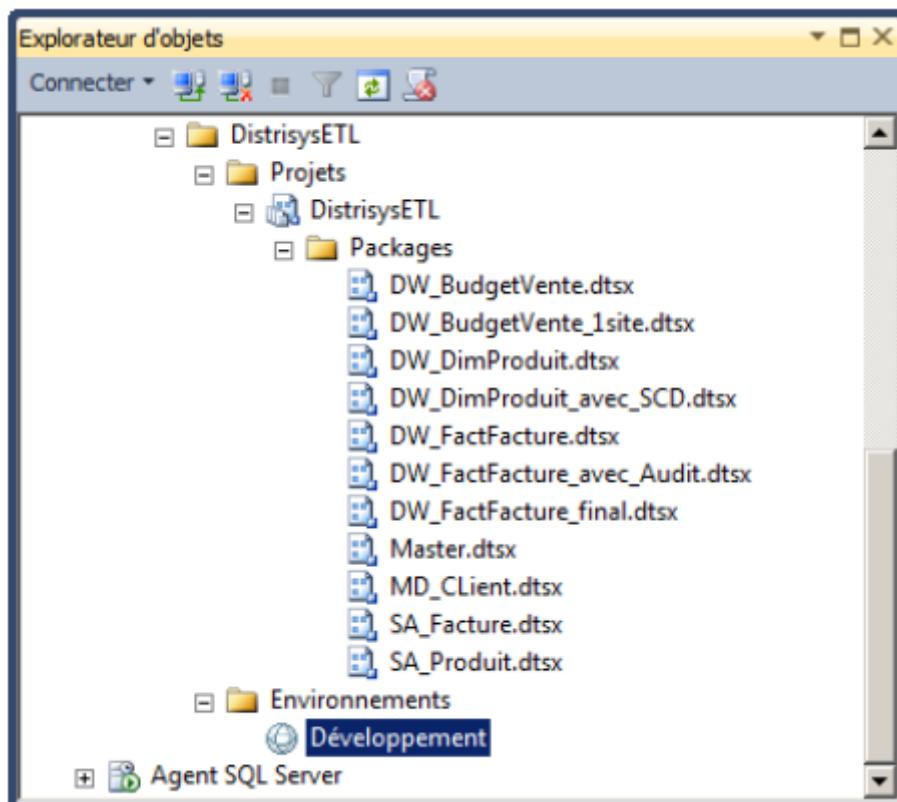
Contenu du catalogue SSISDB

C'est dans ce sous-répertoire que nous allons pouvoir définir notre environnement qui contiendra les variables qui serviront à renseigner les paramètres des packages à leur exécution.

5-4 - Les environnements

Un environnement contient un ensemble de valeurs utilisées pour valoriser les paramètres d'un projet à son exécution.

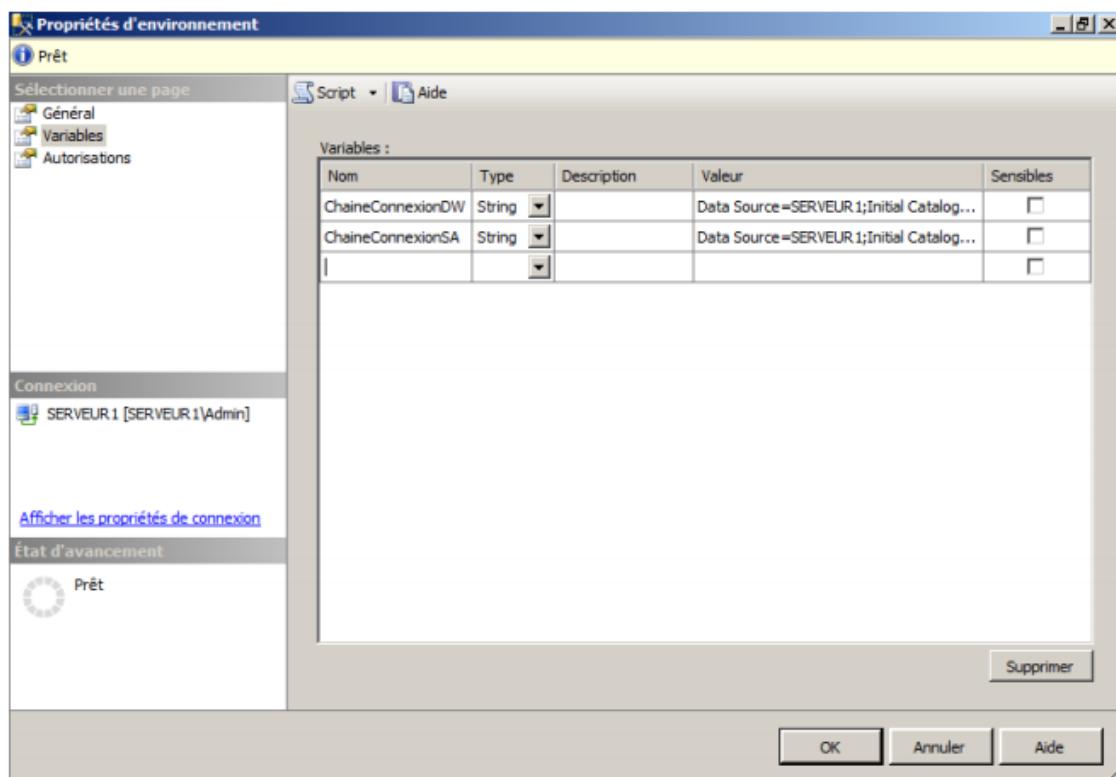
- Créez un nouvel environnement que vous appellerez **Développement** en cliquant avec le bouton droit sur le sous-répertoire **Environnements** et en sélectionnant **Créer l'environnement**.



Création d'un environnement

- Cliquez ensuite avec le bouton droit sur l'environnement créé et sélectionnez **Propriétés**.

Dans la page **Variables**, nous allons créer deux variables ChaineConnexionSA et ChaineConnexionDW, de type String, qui auront pour valeur les chaînes de connexion aux bases de données DistrisysSA et DistrisysDW.



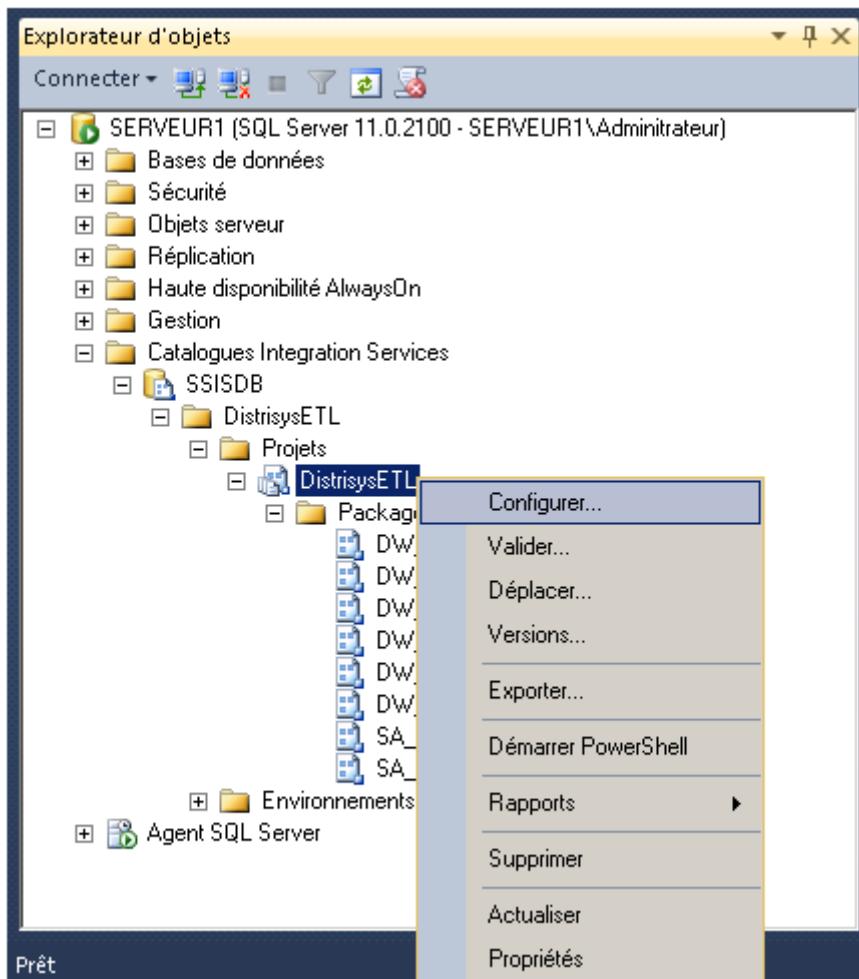
Définition de variables d'environnement

Reprenez les valeurs des paramètres SSIS pour renseigner les valeurs des chaînes variables de l'environnement.

- Cliquez sur le bouton **OK** pour fermer la fenêtre.

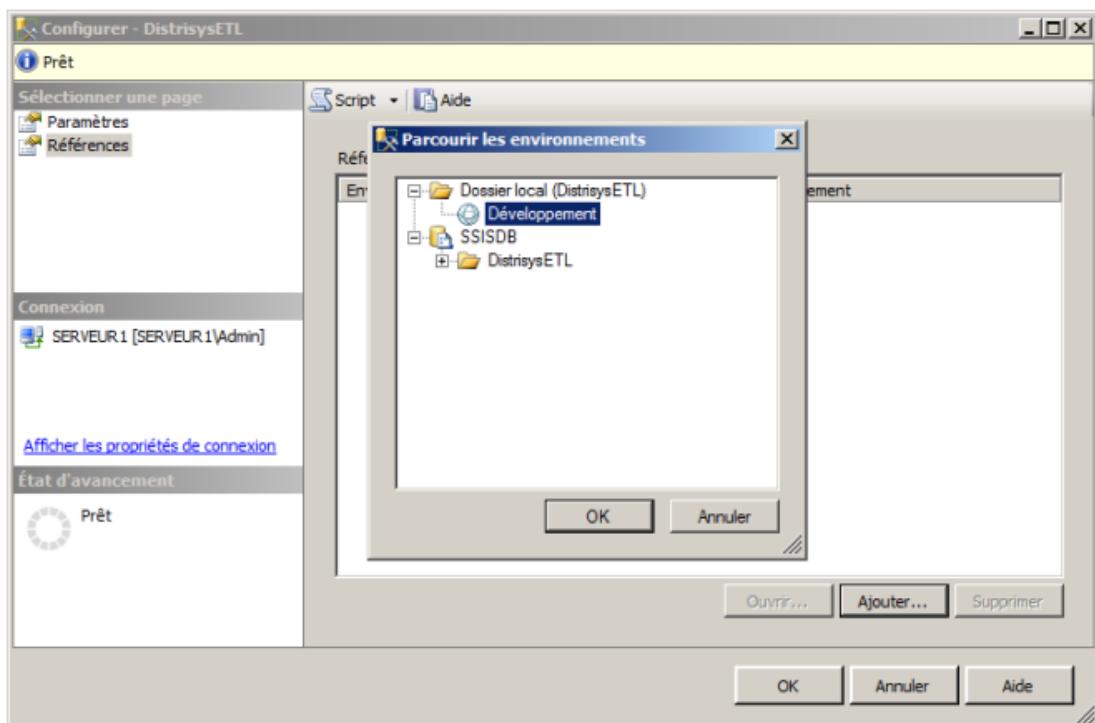
Nous allons maintenant relier les paramètres du projet SSIS aux variables de l'environnement que nous venons de créer.

- Cliquez avec le bouton droit sur le projet **DistrisysETL** et sélectionnez dans le menu contextuel l'option Configurer :



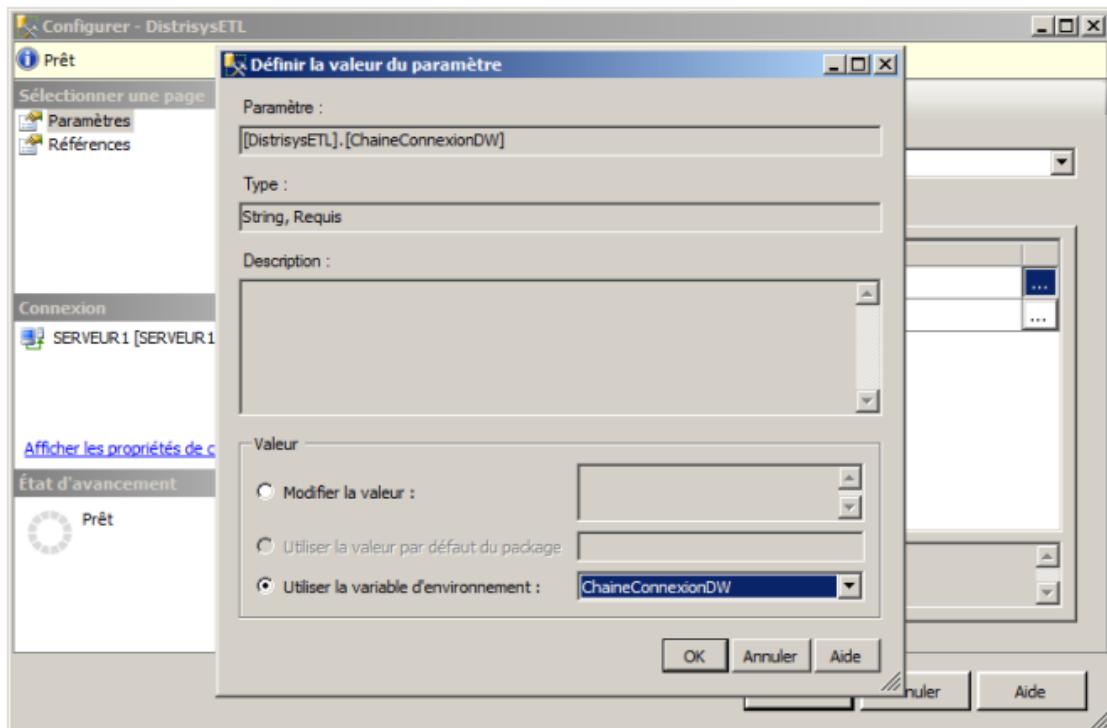
Configuration d'un projet

- Dans la fenêtre **Configurer-DistrisysETL**, au niveau de l'onglet **Références**, nous allons lier le projet à l'environnement **Développement**. Cliquez sur Ajouter et sélectionnez l'environnement **Développement**.



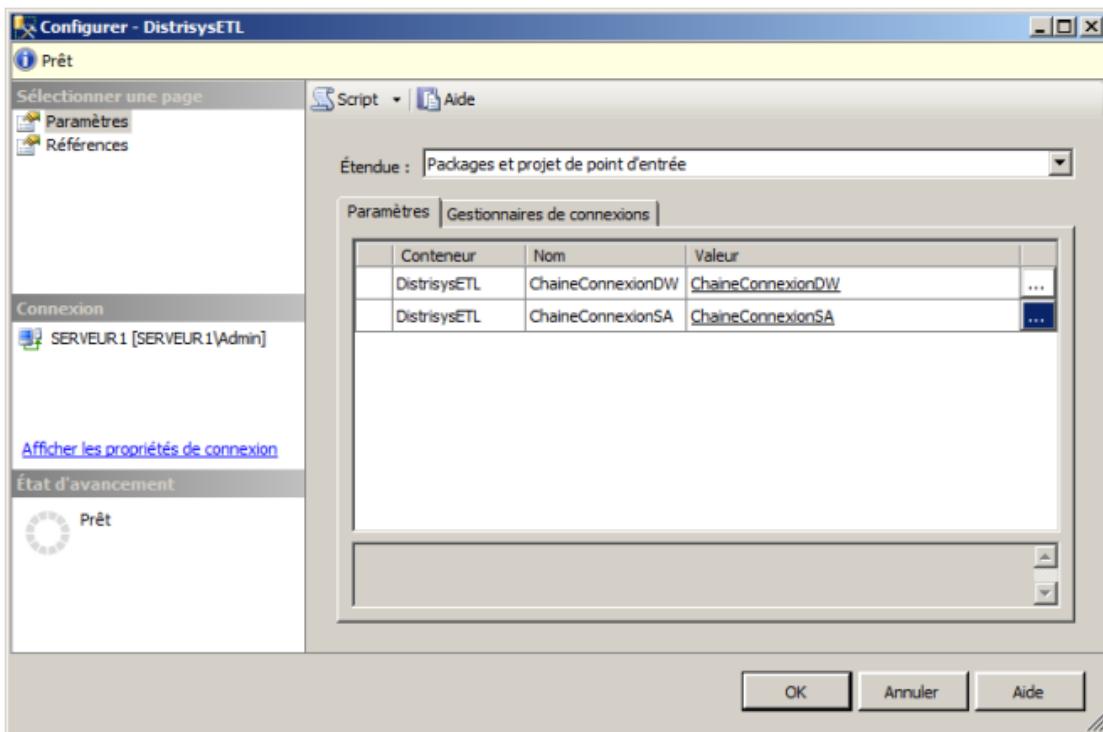
Lier un environnement à un projet

Au niveau de l'onglet **Paramètres**, sur la ligne du paramètre **ChaineConnexionDW**, cliquez pour faire apparaître la fenêtre **Définir la valeur du paramètre**. Sélectionnez alors la variable d'environnement **ChaineConnexionDW** comme valeur du paramètre.

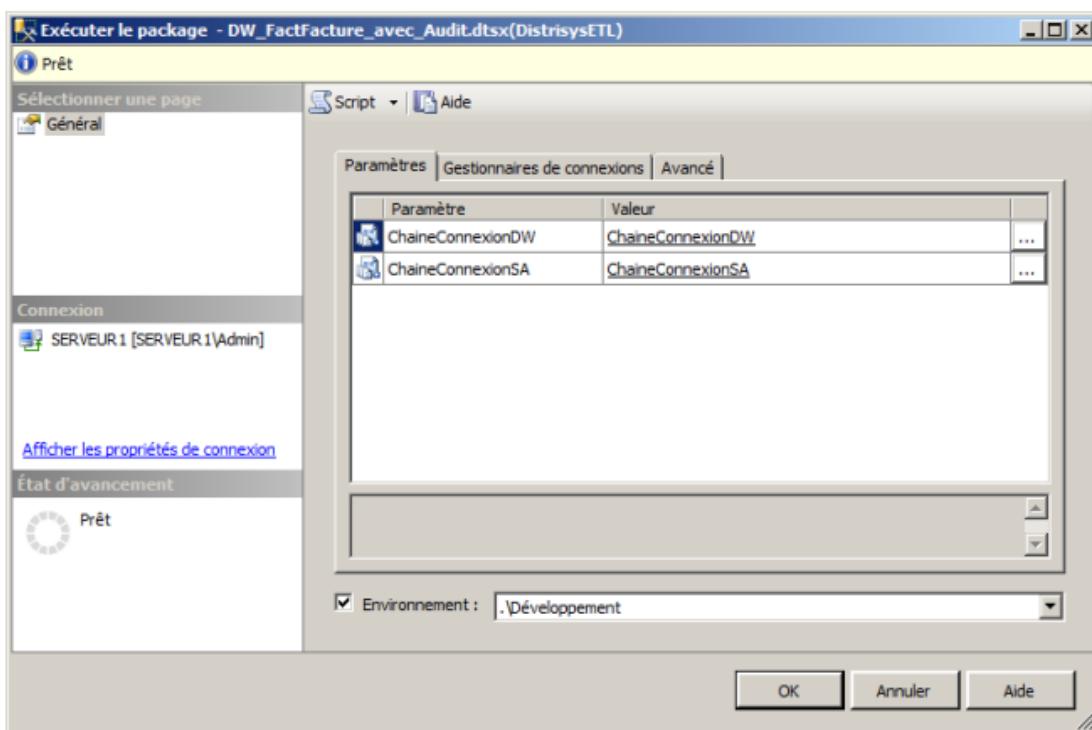


Lier un paramètre du projet à une variable d'environnement

- Faites de même pour le paramètre **ChaineConnexionSA** en le reliant à la variable d'environnement **ChaineConnexionSA**.



- Cliquez sur le bouton **OK**.
- Vous pouvez maintenant exécuter le package **DW_FactFacture_avec_Audit** depuis le catalogue **SSISDB**. Pour cela, cliquez avec le bouton droit sur le package pour faire apparaître le menu contextuel, puis cliquez sur **Exécuter**.
- Cochez la case **Environnement** pour activer le lien avec les variables d'environnement et cliquez sur le bouton **OK** pour exécuter le flux.



Lien entre un projet et un environnement au moment de l'exécution

Lors de l'exécution du flux, les paramètres sont initialisés avec les valeurs des variables de l'environnement Développement.

Le catalogue Integration Services fournit des rapports standards sur les exécutions des packages, accessibles dans l'arborescence du catalogue avec le menu contextuel.

Nous pouvons maintenant déployer le projet en production.

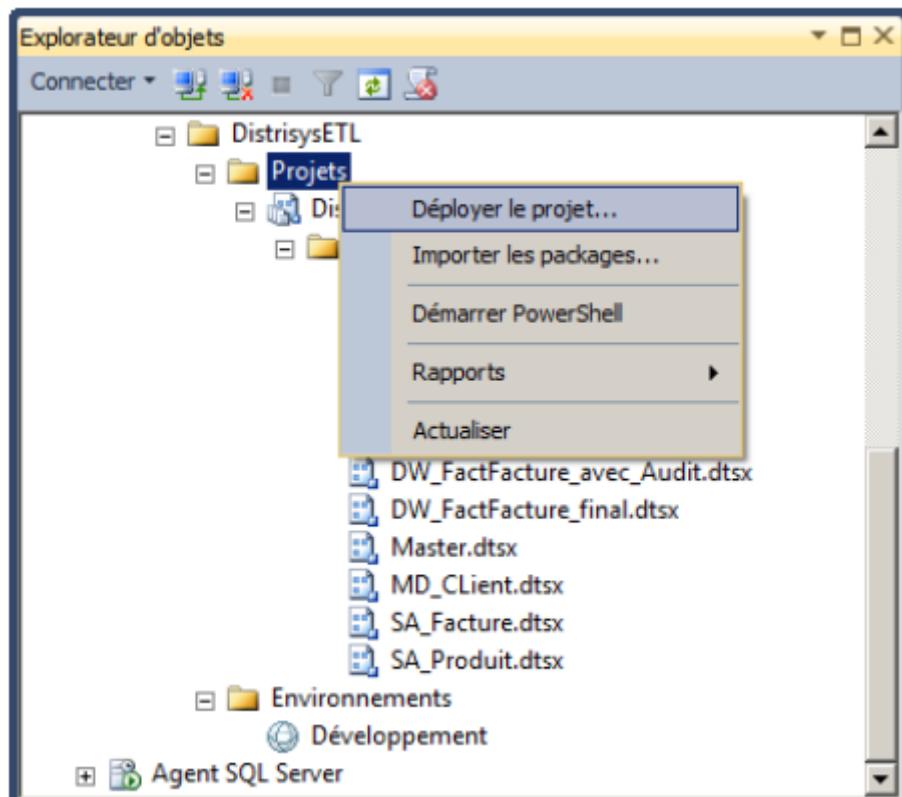
5-5 - Mise en production du projet SSIS

Avant de pouvoir déployer votre premier projet sur votre serveur de production, vous devez y créer un catalogue SSISDB. Dans ce catalogue, vous devez, comme précédemment sur votre serveur de développement, créer un répertoire ainsi qu'un environnement qui contiendra les mêmes variables que celui de développement, mais avec les valeurs liées au contexte de production.

- Connectez-vous avec SSMS à la base de production.
- Créez un catalogue Integration Services à partir du répertoire **Catalogues Integration Services**.
- Créez ensuite dans ce catalogue SSISDB un nouveau dossier nommé **DistrisysETL**.

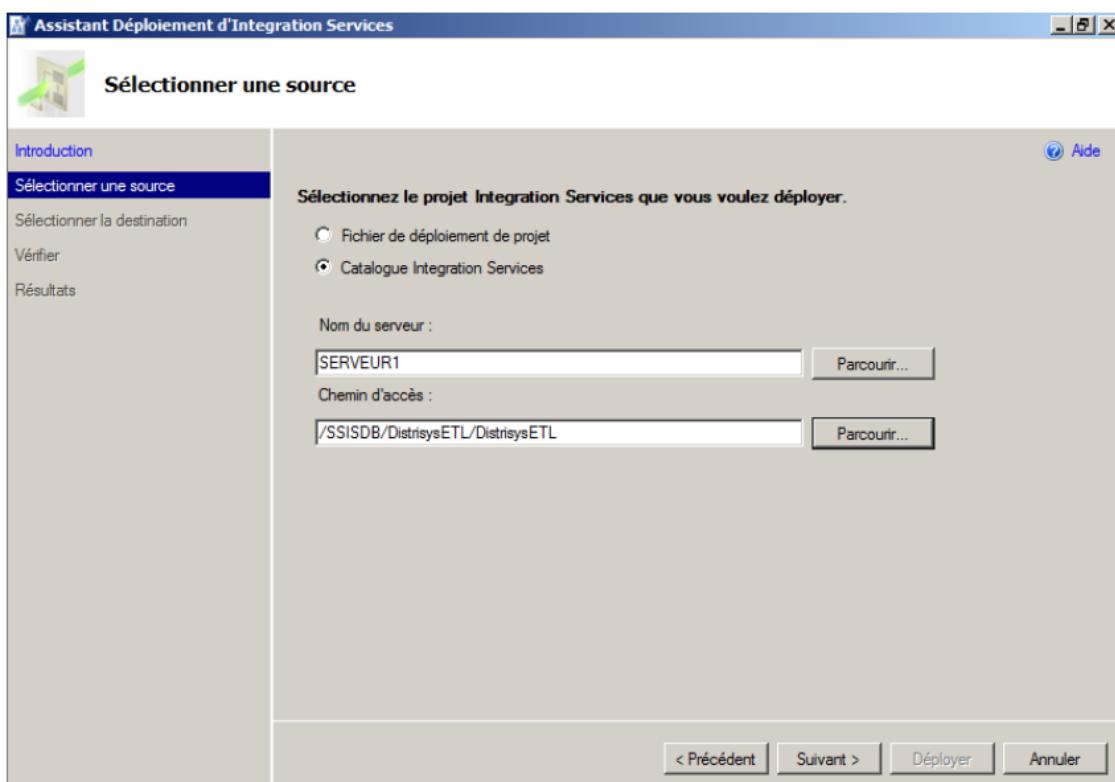
Le déploiement d'un projet sur l'environnement de production se fait à partir du catalogue **SSISDB** de l'environnement de développement.

Connectez-vous avec SSMS au serveur de développement, et dans le catalogue Integration Services, cliquez avec le bouton droit sur le répertoire Projets du **projet** à déployer, puis sélectionnez Déployer le projet.



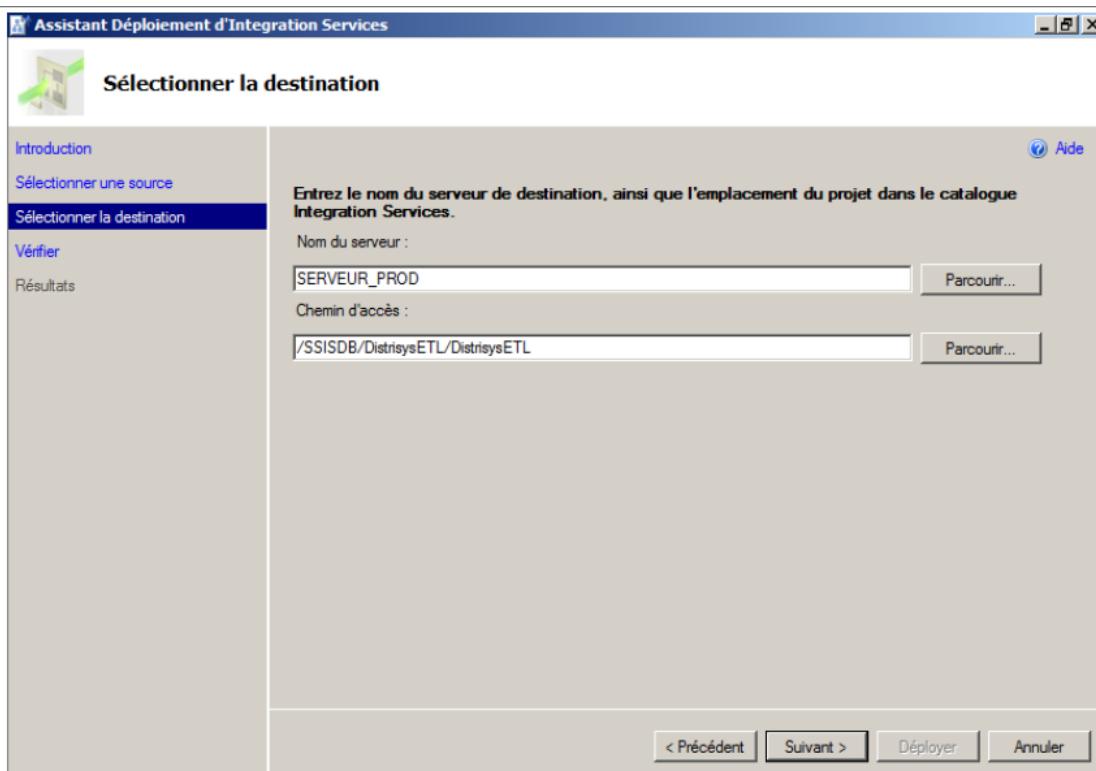
Déploiement d'un projet depuis un catalogue SSISDB

- Comme **Source**, sélectionnez **Catalogue Integration Services**, puis renseignez le nom du serveur source, et grâce au bouton **Parcourir**, sélectionnez le projet à déployer.



Sélection du projet à déployer

- Cliquez sur **Suivant**. Renseignez le nom du serveur de production, puis grâce au bouton **Parcourir**, sélectionnez le répertoire où déployer le projet.



Sélection du catalogue où déployer le projet

- Cliquez sur **Suivant** puis vérifiez vos sélections et validez en cliquant sur **Déployer**.

Le projet est maintenant présent sur votre serveur de production. Lors du premier déploiement d'un projet dans un nouveau catalogue, il faut configurer les liens entre le projet et les environnements du catalogue, ainsi qu'entre les variables des environnements et les paramètres du projet.

- Connectez-vous avec SSMS au catalogue SSISDB du serveur de production.
- Configurez le projet DistrisysETL pour le lier à l'environnement et lier les paramètres du projet aux variables de l'environnement.

Cette opération n'est à faire qu'une seule fois après le premier déploiement du projet. Lors du déploiement d'une nouvelle version du projet, les liens avec l'environnement sont conservés.

Nous avons vu dans la partie précédente comment configurer différents environnements et comment procéder à la mise en production de vos flux. Dans cette dernière partie, nous allons maintenant aborder la planification des flux.

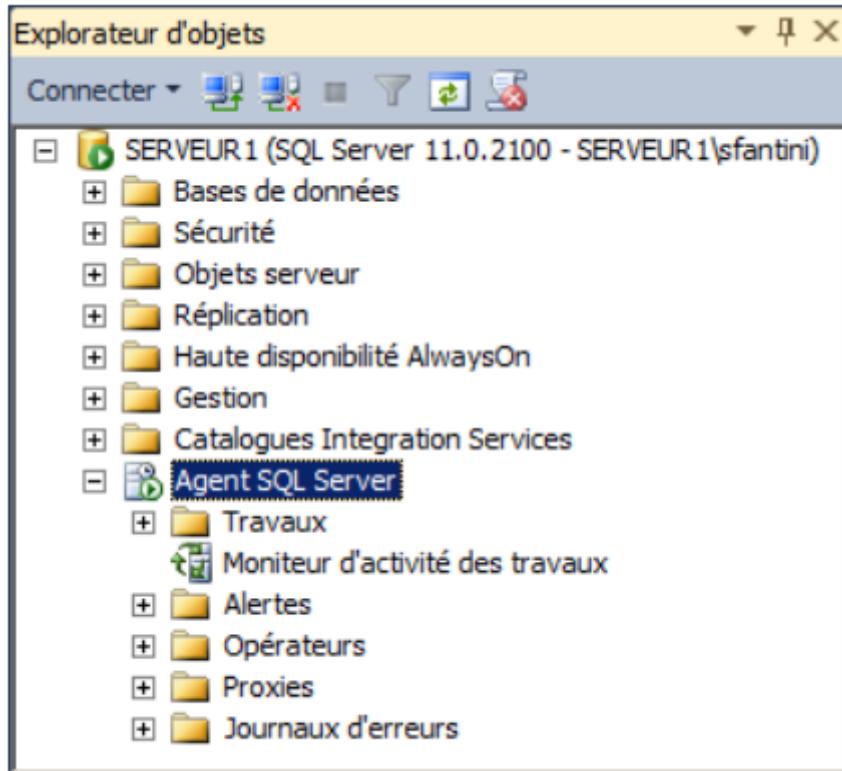
5-6 - Planifier un flux SSIS

La grande majorité des flux d'un entrepôt de données sont réalisés afin d'être exécutés périodiquement. Les services informatiques des grandes organisations disposent généralement d'un ordonnanceur d'entreprise chargé de coordonner l'ensemble des flux des différents systèmes ou applications. Si un tel ordonnanceur est disponible dans votre organisation, nous vous suggérons bien évidemment de planifier vos flux SSIS à l'aide de cet outil. À moins que l'ordonnanceur ne dispose pas d'un connecteur spécifique à SSIS, l'exécution des flux SSIS se fera alors vraisemblablement avec la commande **dtexec**. Si tel est votre cas, reportez-vous à la webographie pour plus de détails sur l'utilisation de cette commande.

Si votre organisation ne dispose pas d'un tel outil, la planification des flux SSIS va alors se réaliser avec l'agent SQL Server.

La manipulation qui va suivre illustre la façon de procéder avec l'agent SQL Server, afin de créer une nouvelle tâche de planification exécutant un flux SSIS précédemment réalisé et publié sur un environnement de production.

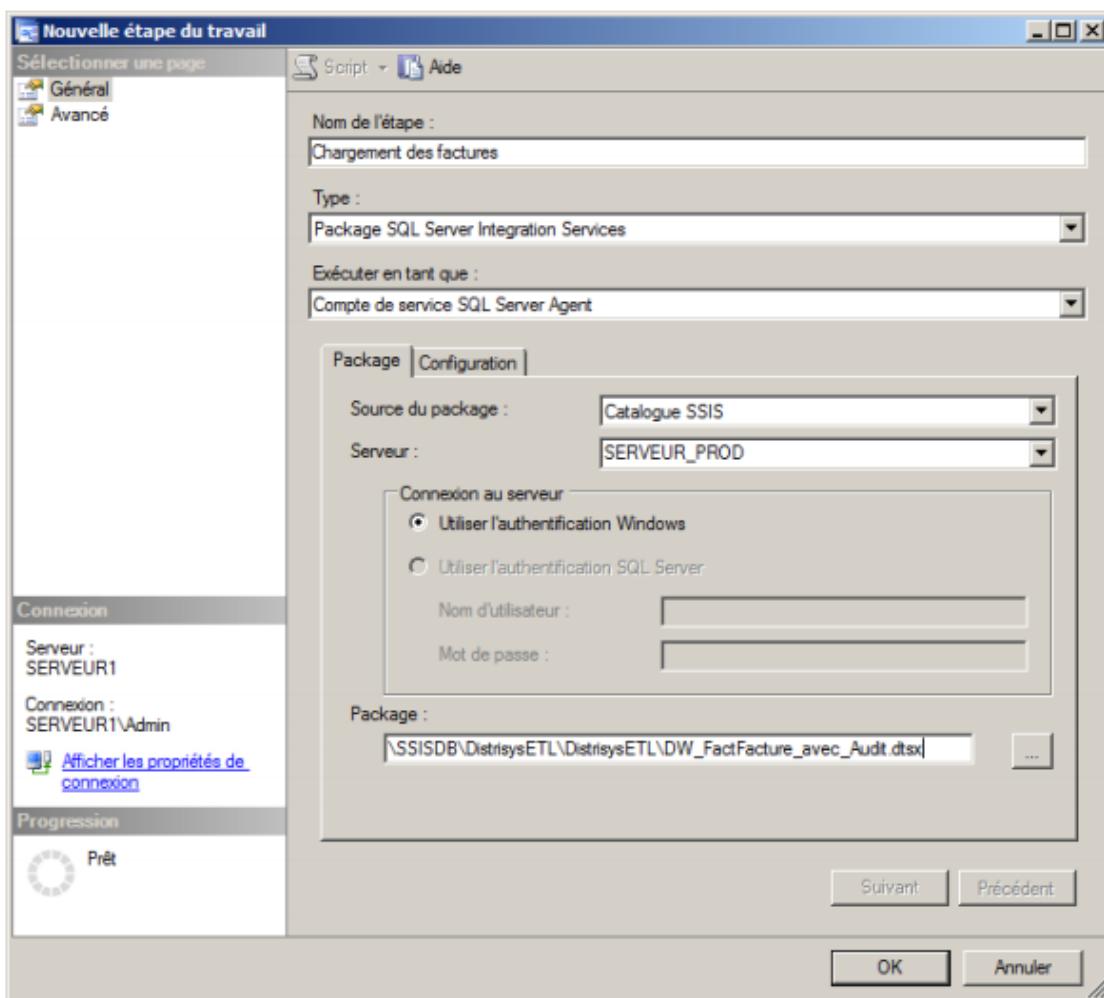
- L'agent SQL Server est disponible sous SSMS. Ouvrez donc l'interface **SSMS**.
- Connectez-vous à votre instance de base de données et déployez le menu **Agent SQL Server**.



La console d'administration de l'agent SQL Server

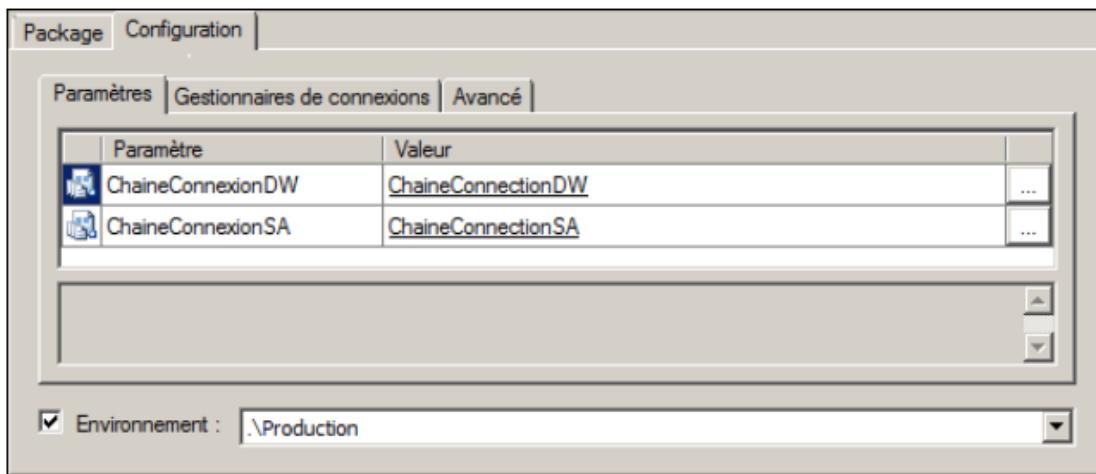
Nous allons maintenant procéder à la création d'un nouveau travail dont le rôle sera d'exécuter le flux SSIS.

- Faites un clic droit sur l'élément **Travaux** et sélectionnez **Nouveau Travail**.
- Spécifiez un nom, par exemple **Chargement des factures**.
- Puis sélectionnez l'onglet **Étapes**.
- Cliquez sur le bouton **Nouveau** pour créer une nouvelle étape.
- Nommez l'étape **Chargement des factures**.
- Dans le type de package, sélectionnez **Package SQL Server Integration Services**.
- Dans l'onglet **Général**, dans **Source du package**, sélectionnez **Catalogue SSIS**, puis dans **Serveur** le nom de votre serveur.
- Dans **Package**, parcourez l'arborescence jusqu'à sélectionner le package à exécuter.



Définition d'une étape SQL Agent pour exécuter un flux SSIS

- Sélectionnez l'onglet **Configuration** pour lier le package à un environnement et renseigner les paramètres.



Lien entre un package SSIS et un environnement dans une étape SQL Agent

- Cliquez sur **OK**, puis allez sur l'onglet **Planifications** pour paramétriser son lancement.
- Définissez ainsi les étapes, la configuration et la planification du lancement de vos flux.

Au cours de ce chapitre, vous avez donc appris à configurer la portabilité d'un package et à programmer son lancement à l'aide de l'agent SQL Server.

Ce chapitre étant clos, vous possédez les prérequis pour traiter du chargement de données. Vous devrez alors être à même de créer vos propres flux et de penser votre système d'audit.

La base de données DistrisysDW finale, le catalogue SSISDB, ainsi que les packages SSIS abordés lors de ce chapitre, sont disponibles en téléchargement sur le site des Éditions ENI. N'hésitez pas à vous y reporter et à vous inspirer des développements réalisés.