

Talend Open Studio for Data Integration

Prendre en main le composant tMap - Créer une jointure

Dans ce tutorial, vous utiliserez les métadonnées créées dans le tutorial "Créer une métadonnée de connexion à un fichier délimité" pour découvrir les particularités du composant tMap. Il sera utilisé pour sélectionner les données entrantes et créer trois sorties différentes :

- les clients de l'Ohio ;
- les clients des autres Etats ;
- les erreurs issues de la jointure entre les deux fichiers d'entrée.

Pour une meilleure consultation, ce tutorial est présenté en quatre parties :

- Prendre en main le composant tMap
- Prendre en main le composant tMap - Fonctionnalités essentielles
- Prendre en main le composant tMap - Créer une jointure
- Prendre en main le composant tMap - Filtre et rejet

Pré-requis :

Pour suivre ce tutorial, vous avez besoin d'extraire et d'installer les fichiers customer.csv et state.txt



1 Ajouter une variable modifiant les données de la table de sortie

Dans le Job Designer :

Double-cliquez sur le tMap pour ouvrir l'éditeur du tMap.

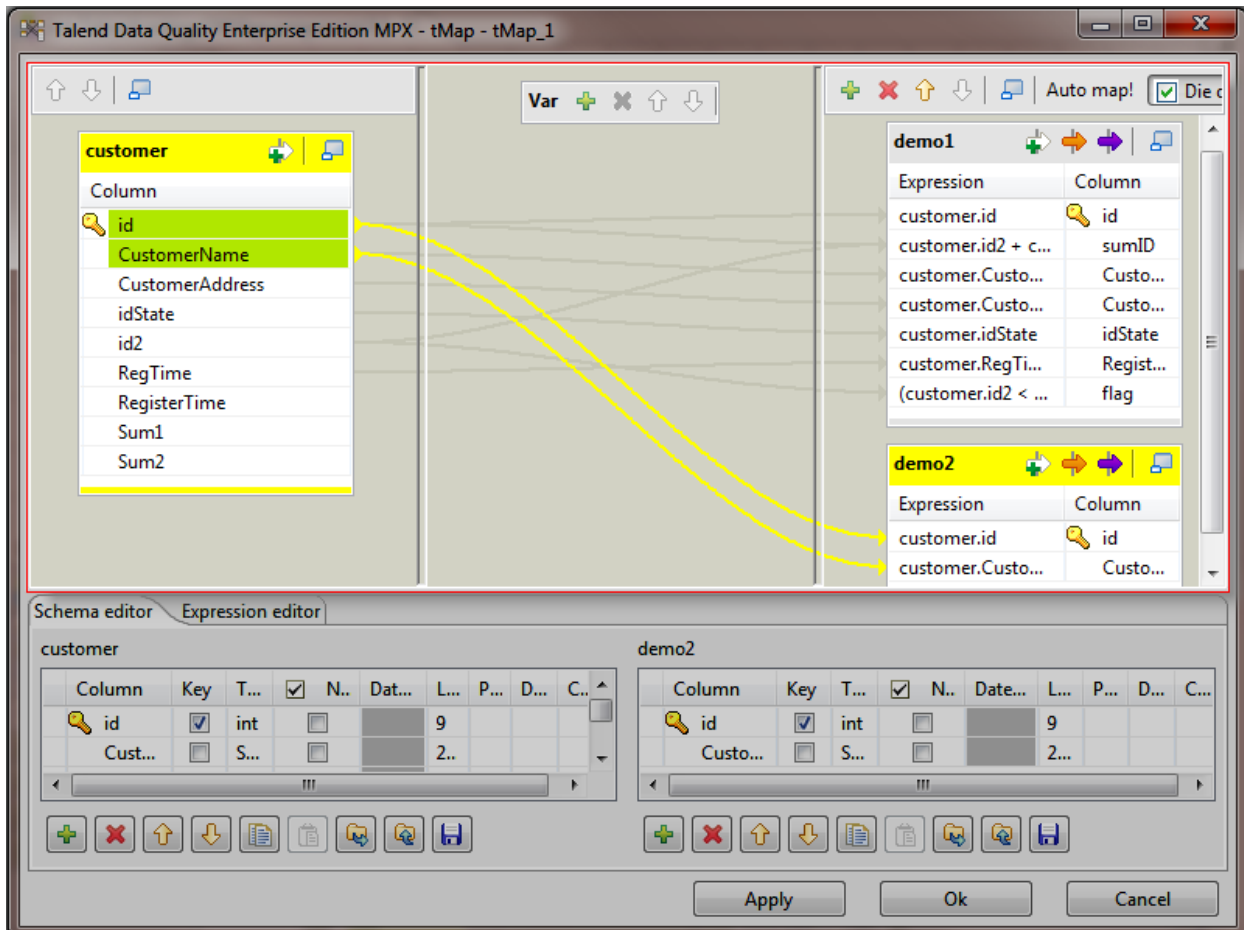
customer (Main)		demo1 (Main)	
id	4	id	4
Customer...	Facelift Kitchen...	sumID	19
Customer...	unknown	Customer...	Facelift Kitchen...
idState	0	Customer...	unknown
id2	15	idState	0
ReqTime	22/08/2002 09:...	RegisterT...	22/08/2002 09:...
RegisterT...	1972-04-23 18:...	flag	inferior

The screenshot shows the Talend Open Studio interface. At the top, a job diagram is visible with components: 'customers', 'customer (Main)', 'tMap_1', 'demo1 (Main)', and 'tLogRow_1'. The 'tMap_1' component is highlighted with a red box. Below the diagram, a data preview window for 'tMap_1' shows two columns: 'customer (Main)' and 'demo1 (Main)'. The table above contains the data for the current row (row 4). The bottom part of the screenshot shows the 'Job TakeadvantageOfMapcomponent' configuration window, including a 'Context' tab with a table, an 'Execution' section with 'Pause' and 'Kill' buttons, and a 'Stats & Traces' section with 'Statistics' and 'Traces' checkboxes. A log window at the bottom shows execution details.

Dans l'éditeur du tMap :

Cliquez sur (+) dans la barre d'outils au dessus de la table demo1, pour ajouter une nouvelle table de sortie. Dans l'assistant Add a new output table, saisissez le nom que vous souhaitez donner à la table (demo2) et cliquez sur OK.

Dans la table customer, appuyez sur Ctrl+Shift tout en sélectionnant les colonnes id et CustomerName et en les glissant dans la table demo2.



Dans la table Var :

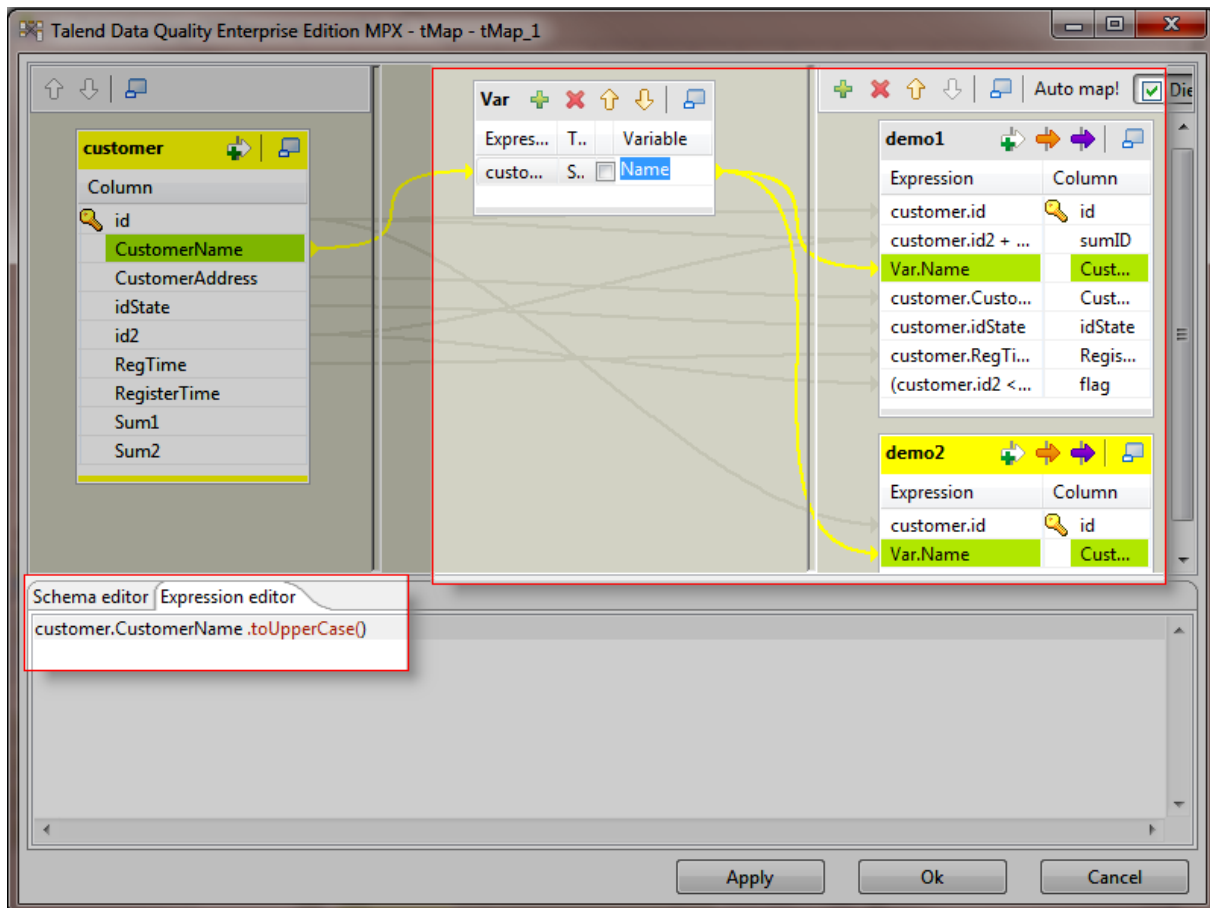
Cliquez sur (+) pour ajouter une ligne.

Dans cette ligne, renommez var1 en Name dans le champ Variable, sélectionnez String dans la liste Typeet cliquez dans le champ Expression.

Cliquez sur l'onglet Expression editor en bas de l'éditeur du tMap.

Dans l'Expression editor, appuyez sur Ctrl+Espace, double-cliquez sur customer.CustomerName dans la liste d'autocomplétion et saisissez l'expression Java `:.toUpperCase()`.

Appuyez sur Ctrl tout en sélectionnant la variable Namedans la table Var et glissez-la dans la colonne customer.CustomerName des tables demo1 et demo2.



Supprimez la table demo2, puisque nous n'allons pas utiliser plusieurs sorties pour le moment.

Dans l'éditeur du tMap :

Sélectionnez la table demo2.

Cliquez sur [x] dans la barre d'outils. Puis dans la boîte de dialogue Remove output table, cliquez sur OK.

Cliquez sur OK pour fermer l'éditeur du tMap.



Lorsque vous devez effectuer les transformations sur plusieurs sorties, utilisez les variables de la table **Var**.

Talend Data Quality Enterprise Edition MPX - tMap - tMap_1

customer

Column

- id
- CustomerName
- CustomerAddress
- idState
- id2
- RegTime
- RegisterTime
- Sum1
- Sum2

Var

Expres... T.. Variable

custo... S.. Name

demo1

Expression Column

- customer.id id
- customer.id2 + ... sumID
- Var.Name Cust...
- customer.Custo... Cust...
- customer.idState idState
- customer.RegTi... Regis...
- (customer.id2 <... flag

demo2

Expression Column

- customer.id id
- Var.Name Cust...

Schema editor | Expression editor

customer.CustomerName.toUpperCase()

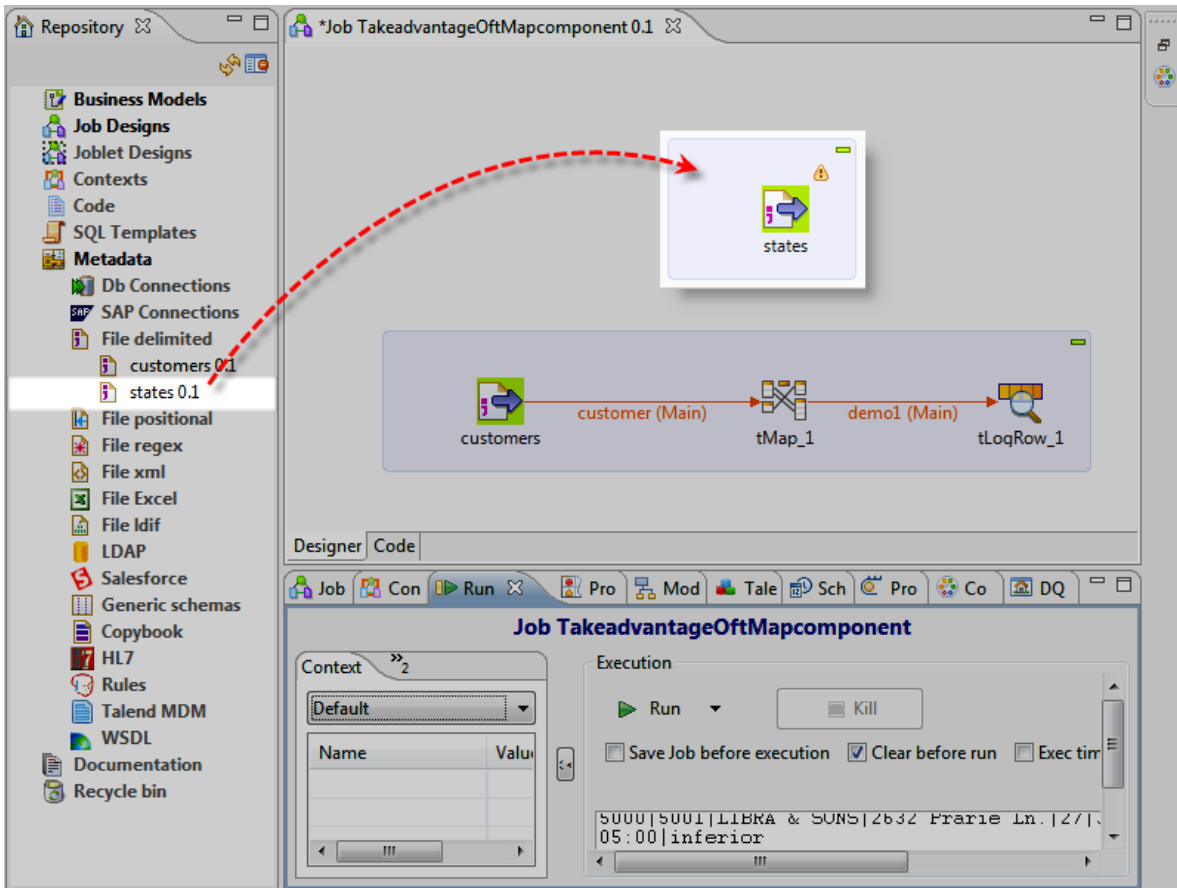
Apply Ok Cancel

2 Déposer un nouveau composant d'entrée et le relier au composant de transformation

Dans le Repository situé à gauche :

Pour ajouter un composant d'entrée, appuyez sur Ctrl+Shift tout en glissant-déposant la métadonnée `states` dans le Job Designer.

Le composant `tFileInputDelimited` est automatiquement créé dans le Job Designer.

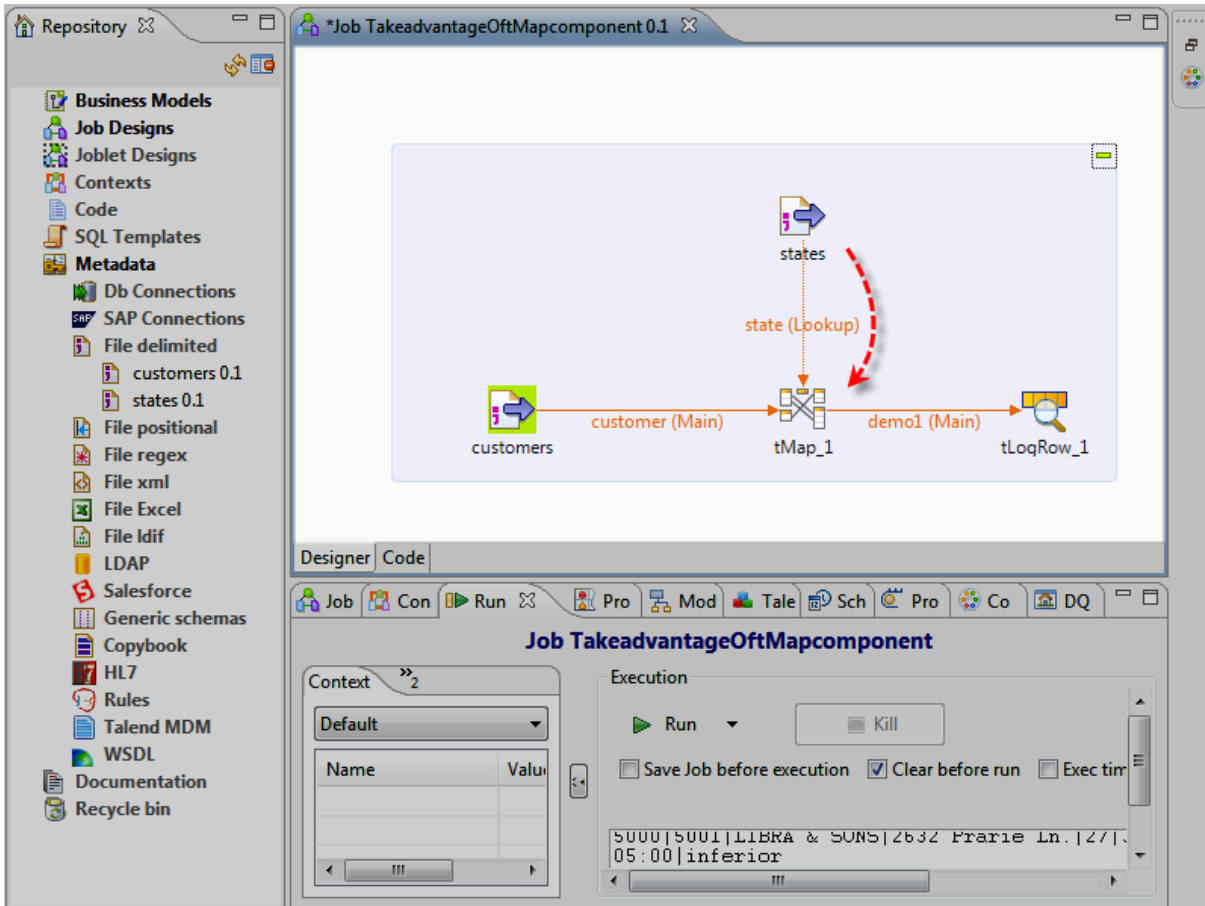


Dans le Job Designer :

Pour relier les composants, cliquez-droit sur le composant states et déplacez-vous jusqu'au tMap.

Double-cliquez sur le nom du lien (row1) et renommez-le state.

Double-cliquez sur le tMap pour ouvrir l'éditeur du tMap.



Dans l'éditeur du tMap :

Sélectionnez la colonne idState dans la table customer et glissez-la dans la colonne idState de la table state.

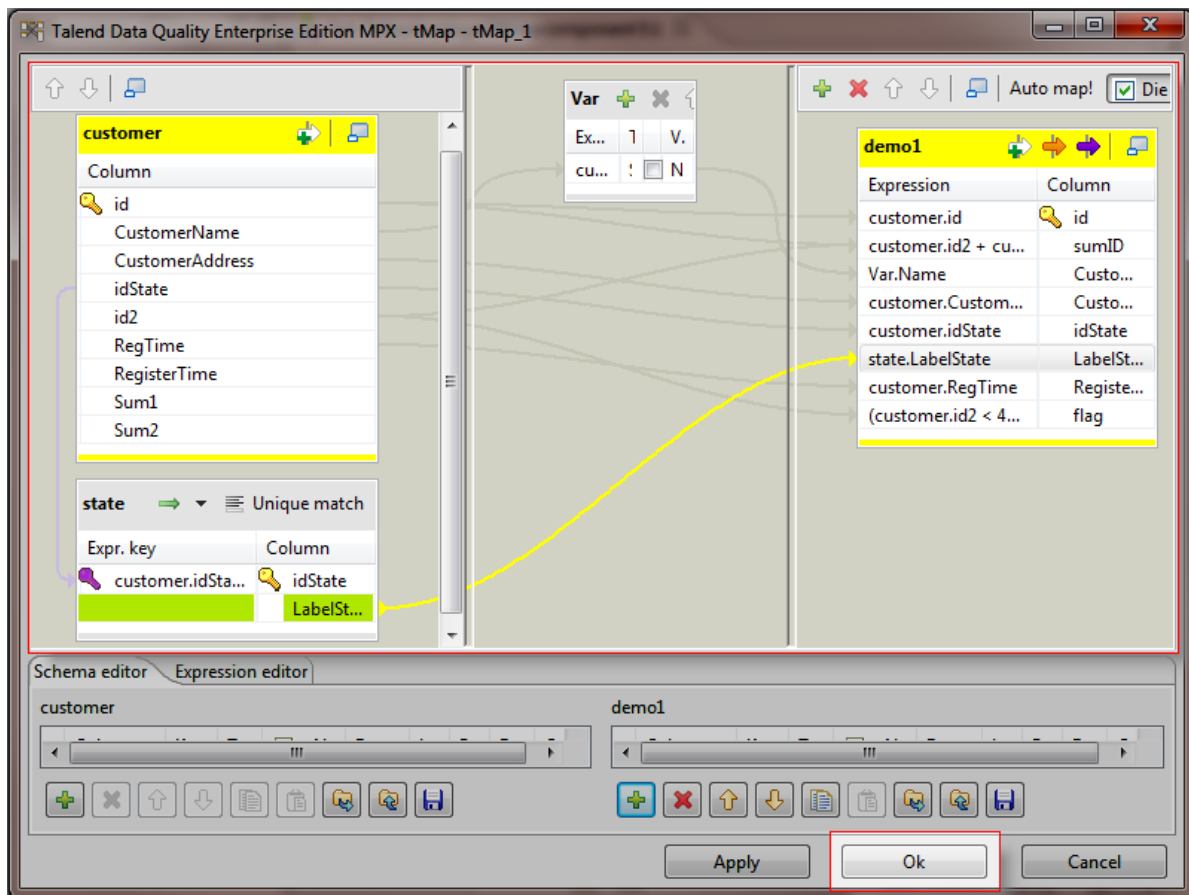
Dans la table state, assurez-vous que l'option Unique match (« Correspondance unique ») est sélectionnée.

Sélectionnez la colonne LabelState dans la table state et glissez-la à droite de la table demo1, sous la colonnecustomer.idState.

Cliquez sur OK.

La boîte de dialogue Propagate s'ouvre.

Cliquez sur Yes pour propager le schéma mis à jour au composant suivant.



3

Exécuter le Job

Dans le Job Designer :

Avant d'exécuter votre Job, enregistrez-le via Ctrl+S.

Cliquez sur la vue « Execution Debug ».

Cliquez sur le bouton « Debugage des Traces ».

Designer Code

Job(Ta) Context Run (Jo) Proble Module Talend Schedu Progres Compo DQ Rep

Job TakeadvantageOfMapcomponent

Context Target execution »1

Default

Name	Value

Execution

Pause Kill

Save Job before execution Clear before run Exec time

Stats & Traces

Statistics Traces

/1610B'S FURNITURE STORE|807 Old Trail Rd. |15|Iowa|27/06

Dans le Job Designer :

La table Traces affiche les données en sortie. Nous pouvons observer que l'identifiant de l'état (idState) apparaît avec son nom (LabelState).

The screenshot displays the Talend Job Designer interface for a job named "Job TakeadvantageOfMapcomponent 0.1". The job design includes a "states" input, a "state (Lookup)" component, a "customer (Main)" component, a "tMap_1" component, a "demo1 (Main)" component, and a "tLogRow_1" component. The "state (Lookup)" component shows a current row with "idState" 50 and "LabelState" Wyoming. The "customer (Main)" component shows a current row with "id" 8, "Customer..." "Toggle & Myer...", "Customer..." "618 Sheriden rd.", "idState" 9, "id2" 15, and "ReqTime" "24/03/2006 23:...". The "demo1 (Main)" component shows a current row with "id" 8, "sumID" 23, "Customer..." "TOGGLE & MYE...", "Customer..." "618 Sheriden rd.", "idState" 9, and "LabelState" Florida. The "tLogRow_1" component shows a log entry: "/116110UB'S FURNITURE STORE|80/ Old Ira11 Rd. |15|Iowa|2//08".

Below the job design, the "Job TakeadvantageOfMapcomponent" execution panel is visible. It includes a "Context" section with "Target execution" set to "1" and a "Default" context. The "Execution" section has a "Pause" button and a "Kill" button. There are checkboxes for "Save Job before execution", "Clear before run", and "Exec time". The "Stats & Traces" section has checkboxes for "Statistics" and "Traces", with "Traces" checked. The log output is visible in a scrollable area.

Ce Job est maintenant terminé !

Il agrège deux fichiers en entrée en créant une jointure et génère une table unique en sortie.

Maintenant, appliquons les filtres créés pour obtenir plusieurs sorties.