



UNIVERSITE DE CORSE
Master Informatique
parcours DFS et DE

1^{ère} année
2025-2026

BD partie 3 CH3 – Processus ETL

Evelyne VITTORI
vittori_e@univ-corse.fr



Plan du cours



CH1 – Principes des Datawarehouse

- Objectifs
- Différences avec une BD
- Architectures DW, DL et DLH



CH2 – Modélisation dimensionnelle

- Concepts de modélisation dimensionnelle
- Schémas en étoile et en flocon



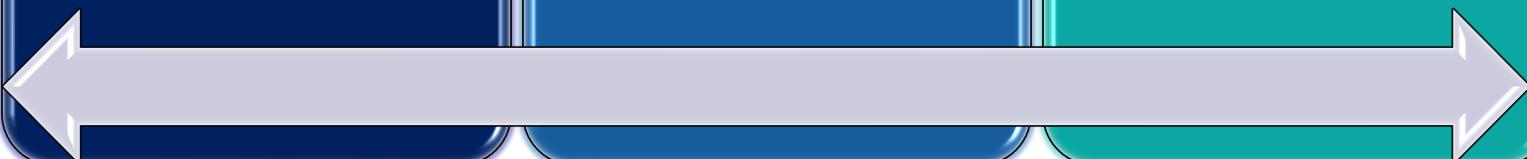
CH3 – Processus ETL

- Définition et rôle d'un processus ETL
- Principaux outils
- Mise en pratique avec PentahoDI

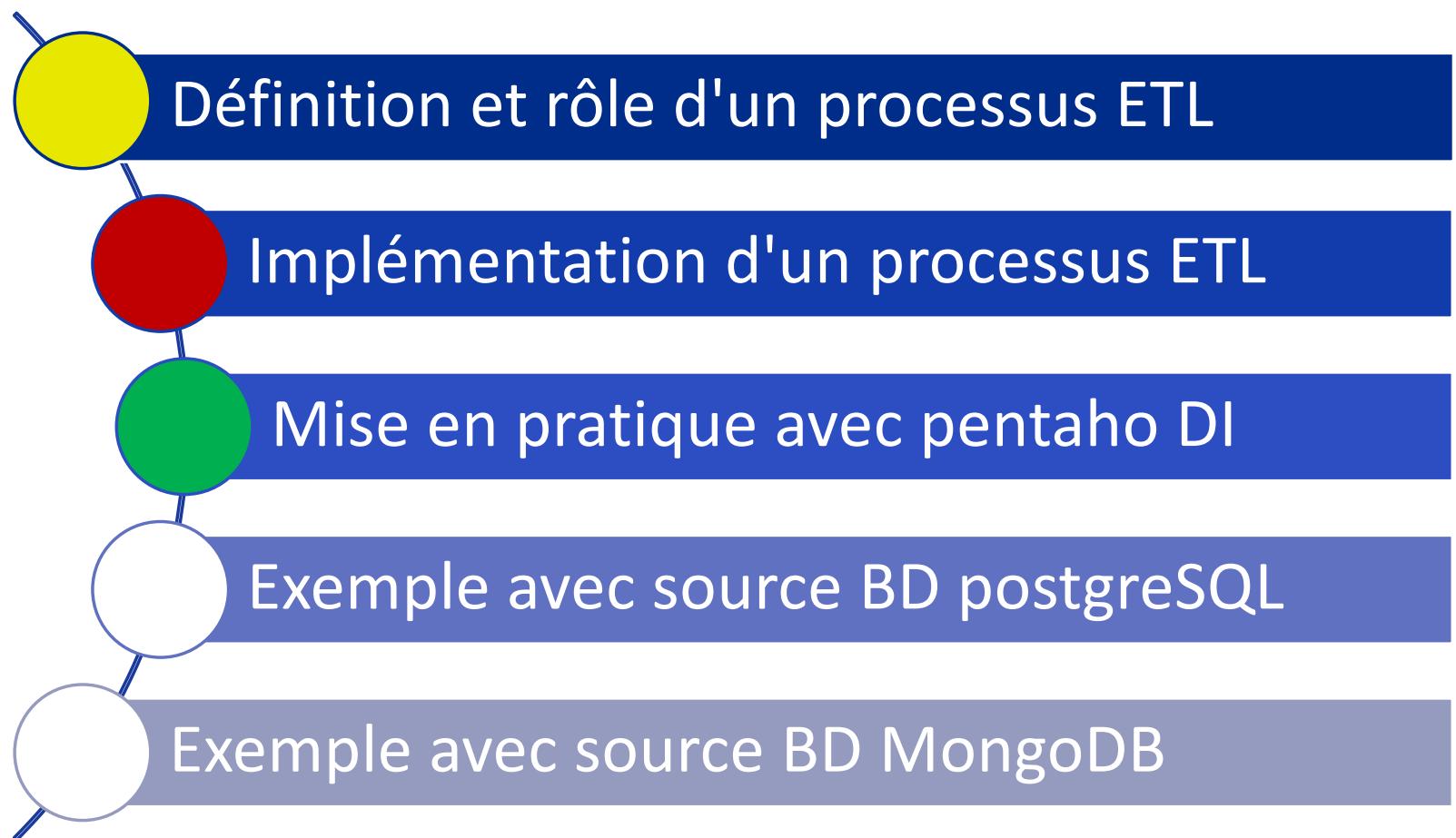


CH4 – Exploitation d'un DW

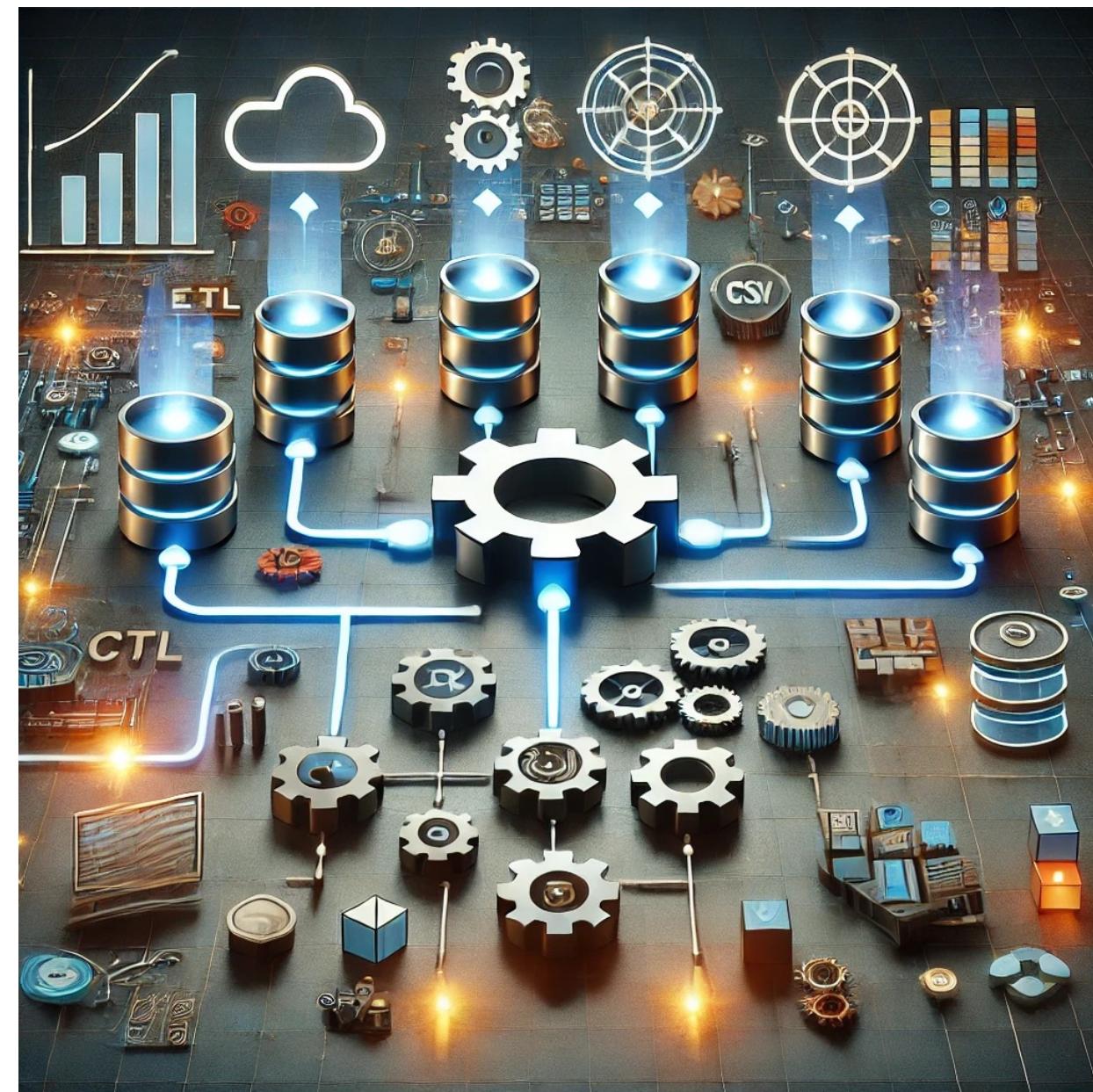
- Principes OLAP
- Notion d'hypercube OLAP
- Langage MDX
- Mise en pratique avec IcCube



CH3 – Procesus ETL



1 – Qu'est-ce qu'un processus ETL ?



Processus ETL

Extraction



Transformation



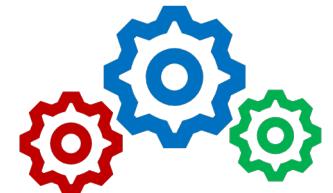
Changement



Qu'est-ce qu'un processus ETL?

Flux **automatisé** qui consiste à

- Extraire des données de diverses sources (**Extract**)
- les Transformer pour les rendre cohérentes et adaptées à l'analyse (**Transform**)
- puis les Charger dans un entrepôt de données ou une autre destination cible (**Load**)



You get the data out of its original source location (E), you do something to it (T), and then you load it (L) into a final set of tables for the users to query.

(Kimball et al., 2008, p369)

Programmation
ou Utilisation
d'un outil
spécifique

Extraction

- Collecte des données à partir de **sources hétérogènes** :
 - BD relationnelles
 - fichiers plats (CSV, JSON),
 - BD NoSQL
- Extraction sans modification et sous divers formats



Transformation

Règles de transformation

- Nettoyage
 - élimination des doublons, gestion des valeurs manquantes
- Agrégation de données
 - regroupement, calcul de moyennes, etc.
- Conversion de format
 - format standard adapté aux tables de l'entrepôt



OBJECTIF

Rendre cohérentes des données issues de différentes sources

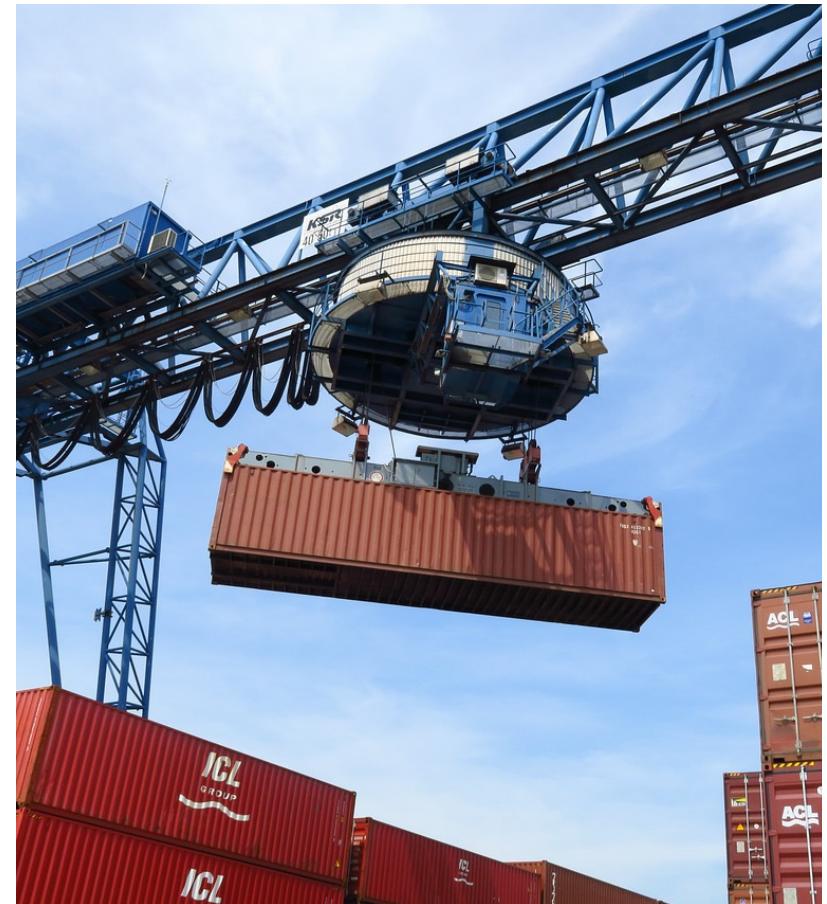
Types de Transformations

- Unifier les données
 - Ex. dates : MM/JJ/AA -> JJ/MM/AA
 - Ex. noms : D-Naiss, Naissance, Date-N -> « Date-Naissance »
- Trier, Nettoyer, Préparer
 - Eliminer les doubles
 - Jointures, projection, agrégation (SUM, AVG, ...)
 - Gestion des valeurs manquantes (NULL) (ignorer ou corriger ?)
 - Gestion des valeurs erronées ou inconsistantes (détection et correction)
 - Vérification des contraintes d'intégrité (pas de violation)
- Inspection manuelle possible



Chargement

- Insertion des données transformées dans un DW ou un datamart
- Chargement
 - **complet**
toutes les données à chaque cycle
 - ou **incrémental**
seules les nouvelles données



Une alternative pour les Datalakes : ELT ≠ ETL

■ Extraction (Extract)

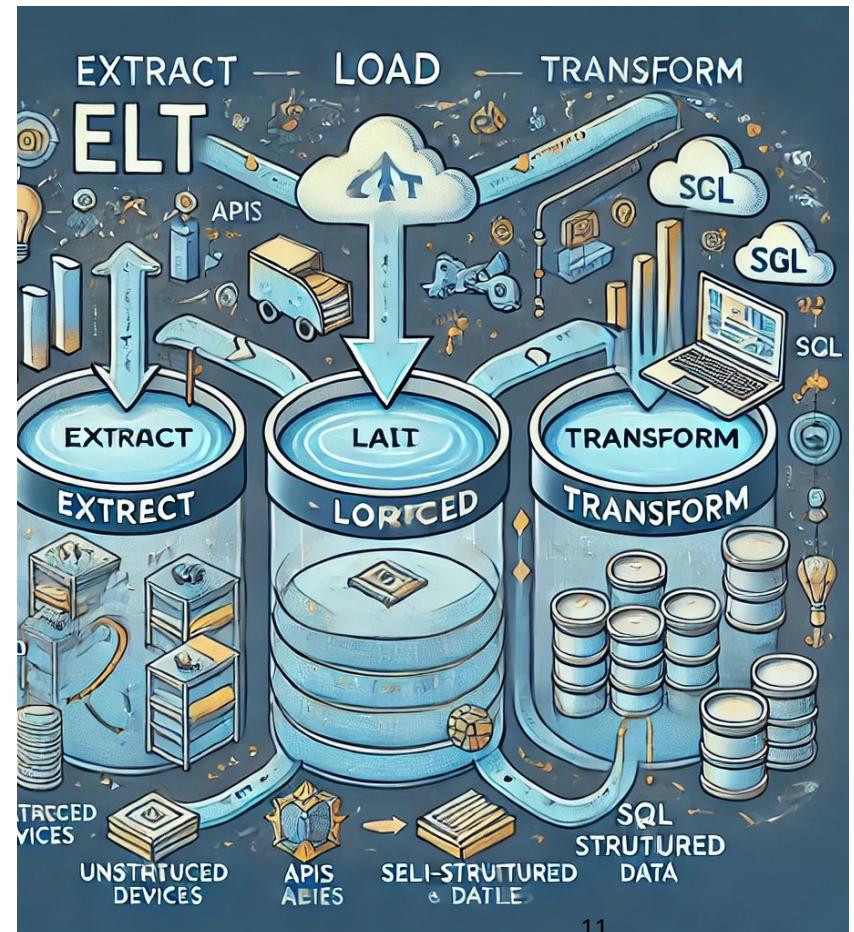
- Collecte des données brutes à partir des sources

■ Chargement (Load)

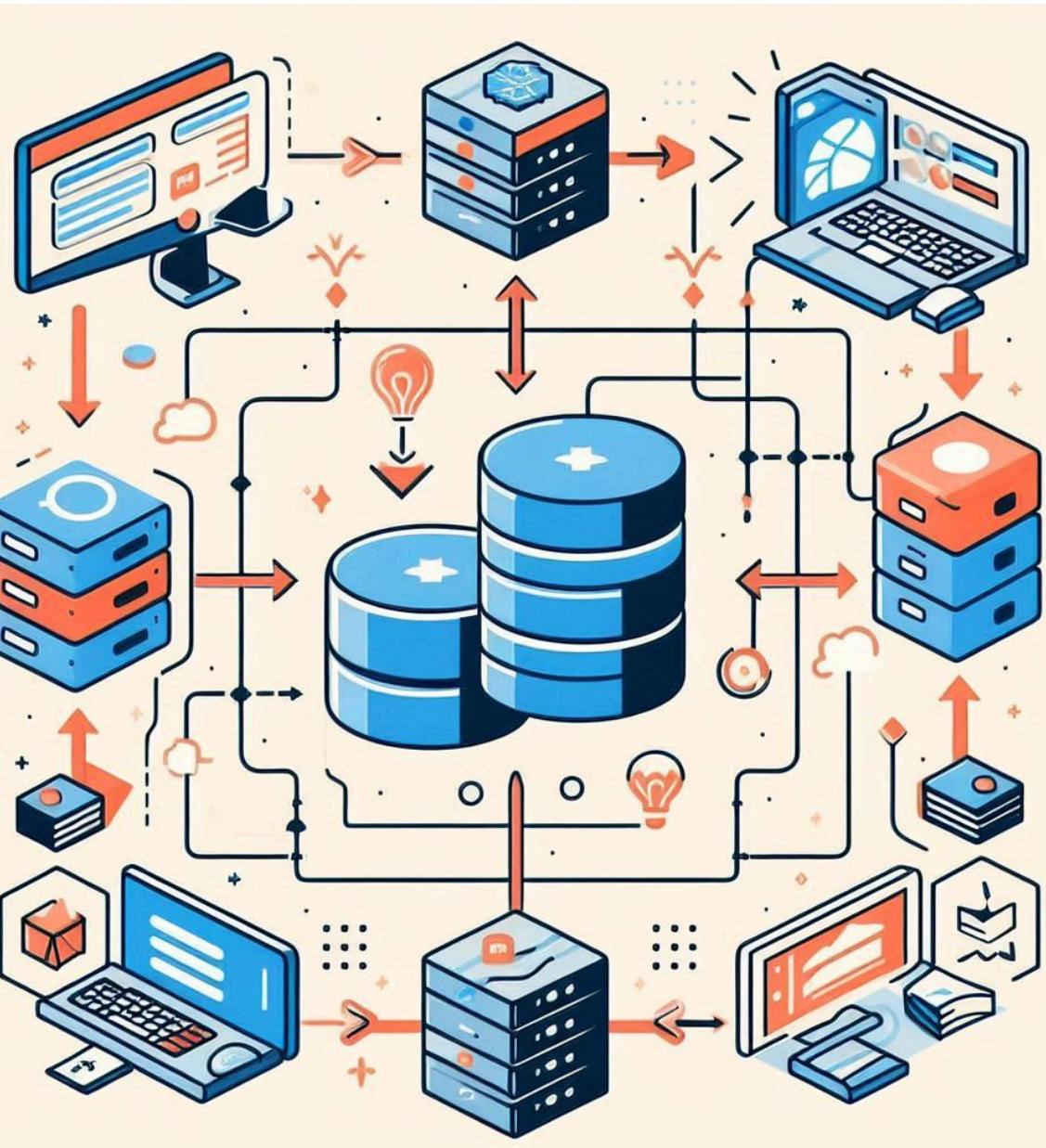
- Stockage des **données non transformées** dans un Data Lake ou un entrepôt

■ Transformation (Transform)

- Nettoyage, structuration, et traitement des données directement dans le moteur cible
- Les requêtes de transformation doivent respecter la syntaxe spécifique du SGBD cible



2 – Implémentation d'un processus ETL



Comment implémenter un processus ETL ?

Programmation manuelle

■ Avantages

- Flexibilité totale : on fait ce que l'on veut !!
- On reste indépendant

Exemple : scripts python

■ Inconvénients

- Temps de développement long
- Maintenance complexe
- Suivi et reporting fastidieux
- Moins adapté aux volumes massifs : complexité accrue pour l'optimisation

A choisir si :

- Scénarios très simples ou ponctuels
- Besoin d'une personnalisation extrême difficile à réaliser dans un outil

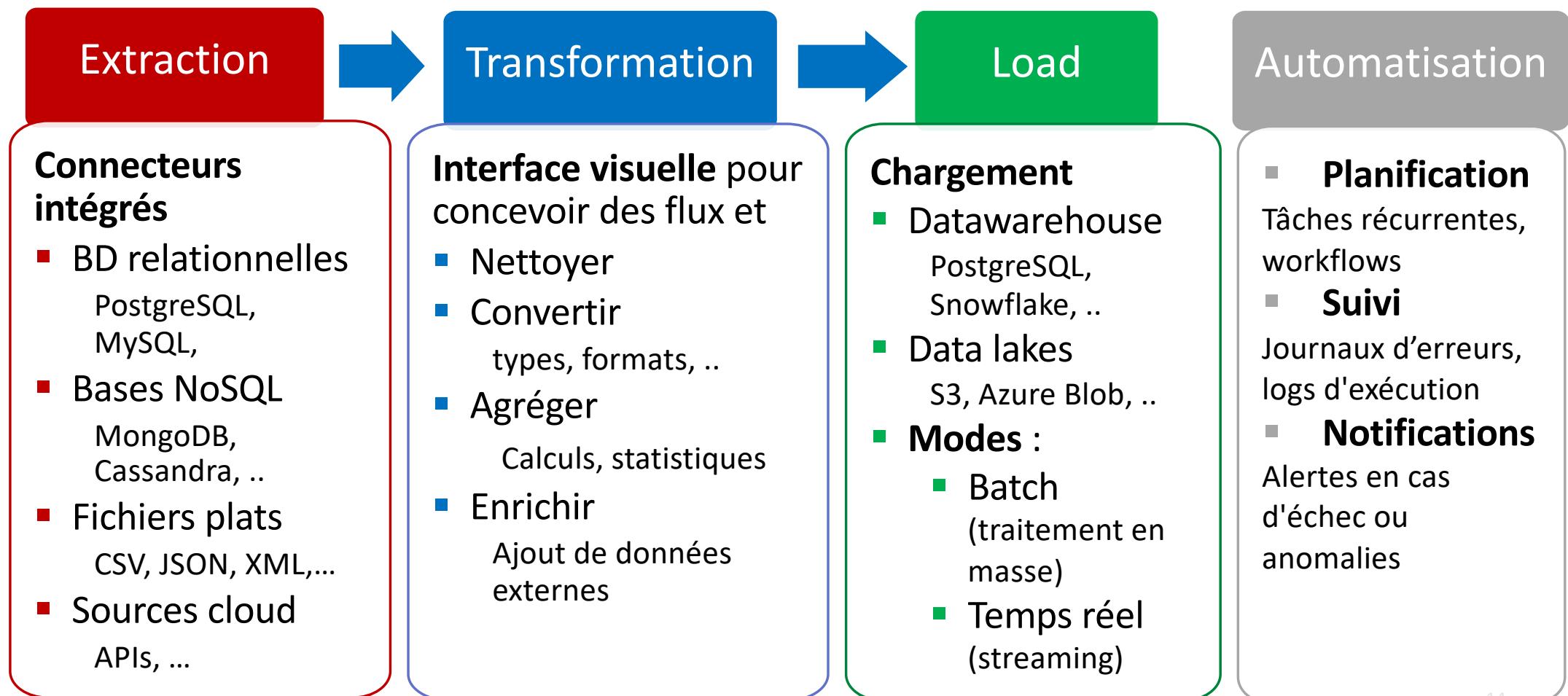
Utilisation d'un outil ETL

■ Avantages

- Interface visuelle (glisser-déposer)
- Temps de développement réduit
- Maintenance et documentation facilitée
- Connecteurs pré-intégrés avec de nombreuses sources de données
- Optimisation intégrée
- Adapté aux volumes massifs

Solution clé pour automatiser et accélérer le traitement des données massives

Caractéristiques des outils ETL



Quels outils pour l'ETL ?

Open-source/gratuits	Propriétaires / Payants	Cloud
<ul style="list-style-type: none">■ Pentaho Data Integration (PDI)■ Apache NiFi■ Hevo Data	<ul style="list-style-type: none">■ Talend Open Studio■ Informatica PowerCenter■ Microsoft SQL Server Integration Services (SSIS)■ Oracle Data Integrator (ODI)	<ul style="list-style-type: none">■ AWS Glue■ Google Dataflow■ Azure Data Factory

Version community Edition : Notre choix dans ce cours

3 - Définition de processus ETL avec PentahoDI

Pentaho

Platform ▾ Solutions ▾ Pricing Services Resources ▾ Blog Get a Demo

Low-code Environment for Data Preparation

80% saved in data operations costs

7x faster display of knowledge graphs

55% savings of data scientist's time finding & evaluating data

Streamline Hybrid Data Estates with Advanced Data Orchestration

 pentaho® Data Integration



Pourquoi choisir PentahoDI (PDI)?



- Outil ETL/ELT open-source et gratuit
 - Version Community Edition
- Ancien nom (kettle)
- Composant essentiel de la suite Pentaho

Business Intelligence

■ Atouts

- Interface conviviale et visuelle
- Connectivité étendue
- Communauté et écosystème
- Adapté à une variété de cas d'usage

A diagram showing the components of Pentaho Business Intelligence. At the top, the Pentaho logo is displayed with the text "open source business intelligence™". Below the logo, five icons represent different tools: "Data Integration" (yellow cylinder), "Reporting" (document with a pencil), "Data Mining" (network graph), "Dashboards" (grid of four squares), and "Analysis" (blue cube). To the right of these icons, their respective names are listed: Data Integration, Reporting, Data Mining, Dashboards, and Analysis. A vertical line connects the "Dashboards" and "Analysis" icons.

- Pentaho DI
- Pentaho Server
- Pentaho Report Designer : rapports interactifs, statiques ou dynamiques
- Pentaho Dashboard Designer : tableaux de bord visuels
- Pentaho Schema Workbench : Outil graphique pour concevoir des schémas OLAP
- Mondrian OLAP : Moteur OLAP

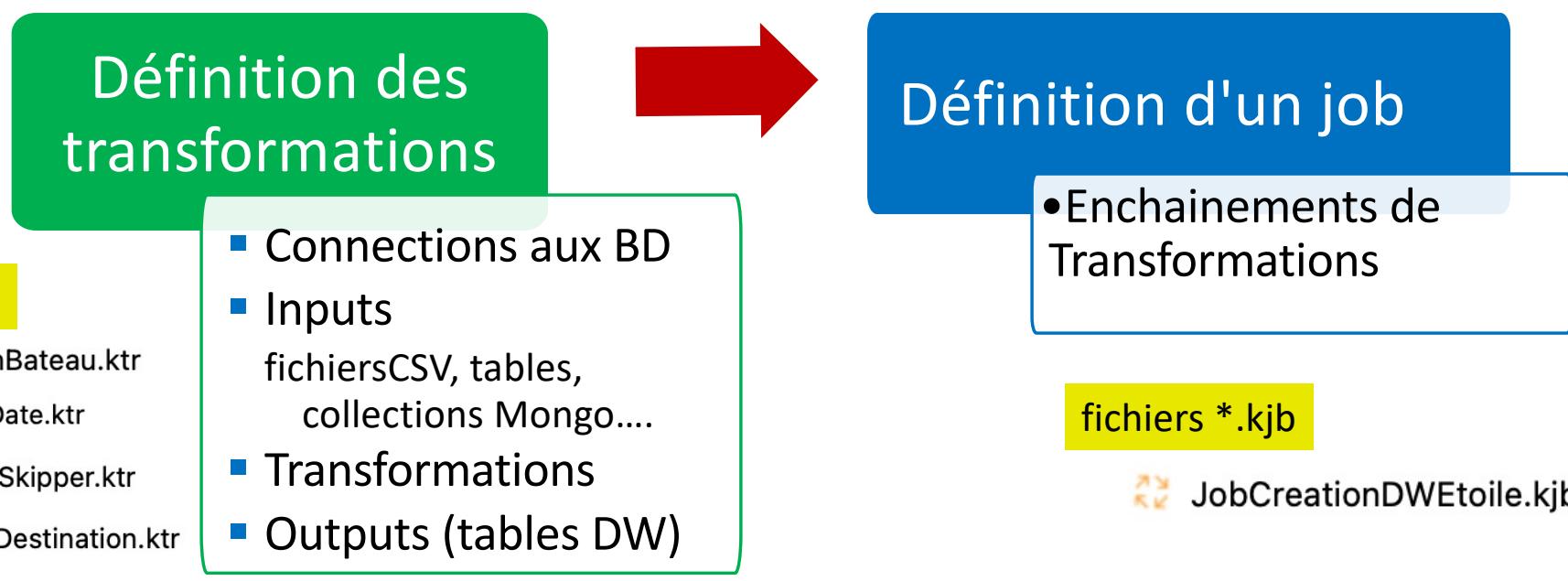
Installation de Pentaho-DI (version desktop)

- Télécharger pentaho Data Integration community Edition version 9.4
 - <https://github.com/ambientelivre/legacy-pentaho-ce?tab=readme-ov-file>
 - Fichier pdi-ce-9.4.0.0-343.zip
 - Dézipper le fichier dans le repertoire de votre choix.
 - Le fichier à exécuter est : spoon.bat (windows) ou spoon.sh (macOS).
- Modifier le driver jdbc postgres utilisé par pentaho pour assurer la connection avec votre serveur postgres :
 - Télécharger le driver jdbc postgresql-42.7.4.jar sur le site officiel postgres
 - <https://jdbc.postgresql.org/download/postgresql-42.7.4.jar>
 - Dans le dossier pentaho/data-integration/lib/ de votre installation :
 - supprimer le fichier postgresql-42.2.23.jar et
 - remplacer le par le driver postgresql-42.7.4.jar que vous avez téléchargé.



Définition d'un processus ETL sous pentaho-DI

- Un processus est appelé **JOB**
- Un job est constitué d'une suite de **TRANSFORMATIONS**



Définition d'une transformation

The screenshot shows the Spoon interface for defining a transformation. The top menu bar includes 'Data Integration', 'File', 'Edit', 'View', 'Action', 'Tools', and 'Help'. The 'File' menu has 'New' (highlighted in blue), 'Open...', 'Open Recent', 'Transformation Job', and 'Database Connection'. The 'Tools' menu has 'Transformation', 'Job', and 'Database Connection'. The 'Design' tab is selected in the top-left corner of the main window.

The main window displays 'Transformation 1' with various toolbars and a central canvas. A red arrow points from the 'New' option in the 'File' menu to the 'choix des sources' (choice of sources) box. A green arrow points from the 'Input' option in the sidebar to the same box. A blue arrow points from the 'Output' option in the sidebar to the 'choix des sorties' (choice of outputs) box. A red arrow points from the 'Transform' option in the sidebar to the 'choix des transformations intermédiaires' (choice of intermediate transformations) box.

choix des sources

choix des sorties

choix des transformations intermédiaires

Design

- > Input
- > Output
- > Streaming
- > Transform
- > Utility
- > Flow
- > Scripting
- > Pentaho Server
- > Lookup
- > Joins
- > Data Warehouse
- > Validation
- > Statistics
- > Big Data
- > Cryptography
- > Job
- > Mapping
- > Bulk loading
- > Inline
- > Experimental
- > Deprecated
- > History

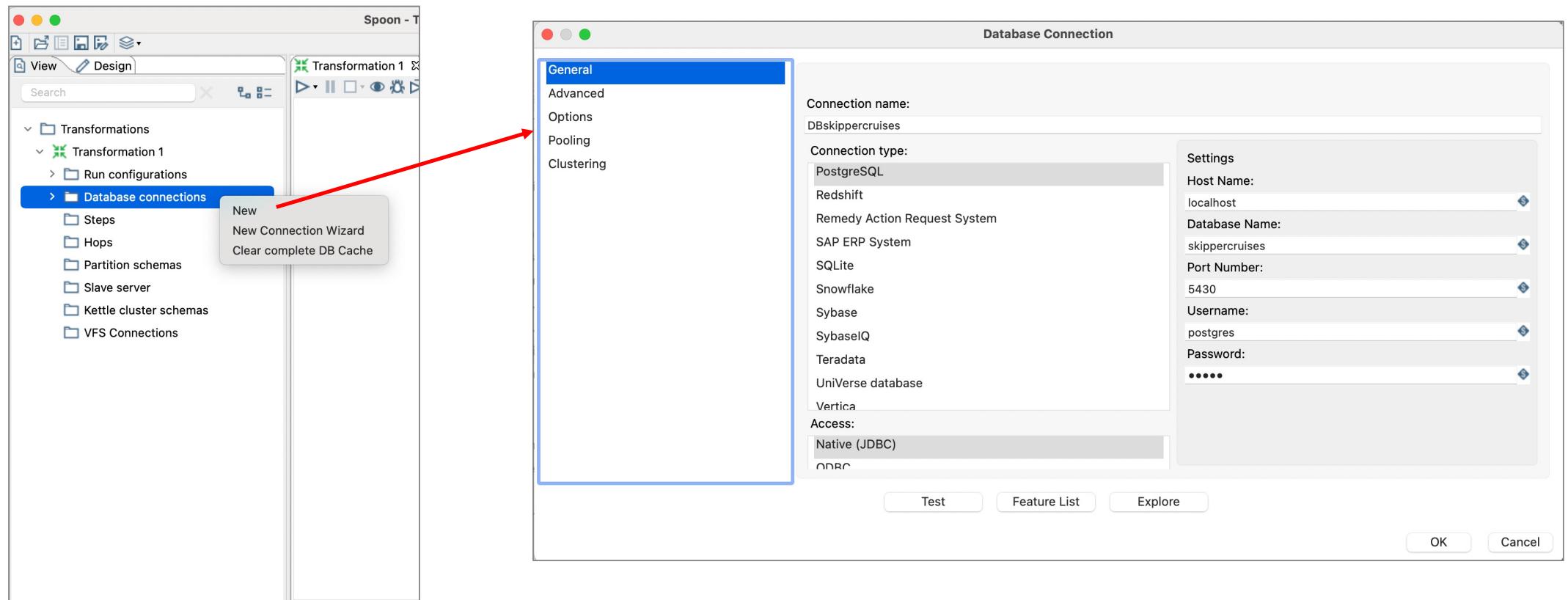
Transformation 1

100%

Drag & Drop a Step
Also try shift + double-click

- La transformation est définie par drag-drop des outils choisis dans l'onglet Design

Définition des connexions aux BD



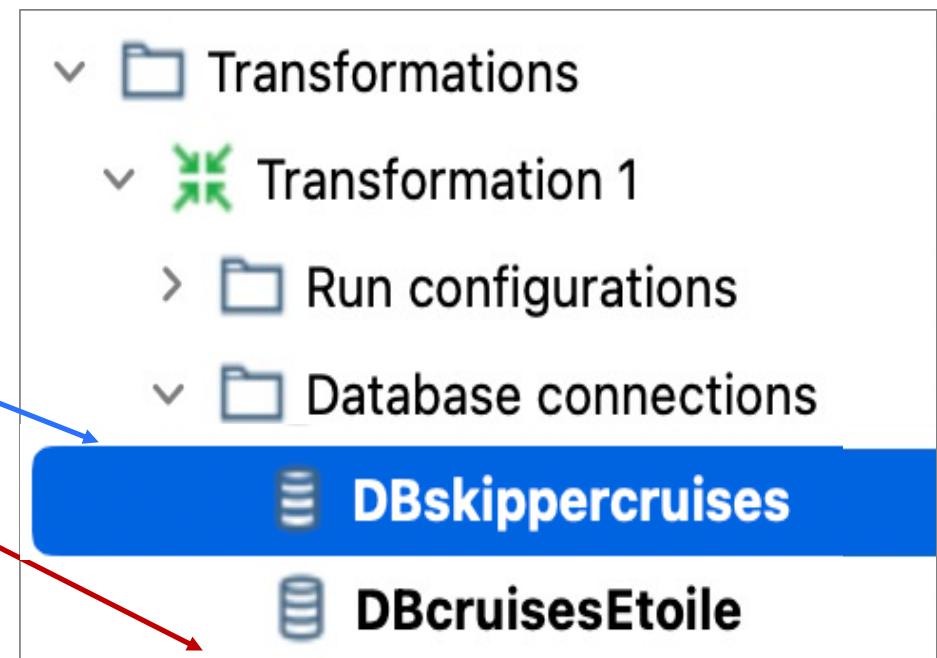
Définition des connexions aux BD

- Définition de plusieurs connexions :

- **BD source**
- **DW destination**

- Pour chaque connexion

- Choix "partage"(share)
 - pour accéder à la source dans toutes les transformations



Exemple de Transformation fichier CSV->Table postGres

Définition
input

The screenshot shows the Pentaho Spoon interface for defining a transformation named "Importation Vente...". On the left, the "Input" node is selected. A red arrow points from the "CSV file input" icon in the center to the configuration dialog on the right. The configuration dialog is titled "CSV file input" and contains the following details:

- Step name: Extraction depuis fichier CSV
- Filename: \${Internal.Entry.Current.Directory}/ventes.csv
- Delimiter: ;
- Enclosure: "
- NIO buffer size: 50000
- Lazy conversion?: checked
- Header row present?: checked
- Add filename to result?: checked
- The row number field name (optional):
- Running in parallel?: unchecked
- New line possible in fields?: checked
- Format: mixed
- File encoding: UTF-8

Below the configuration are four columns of field definitions:

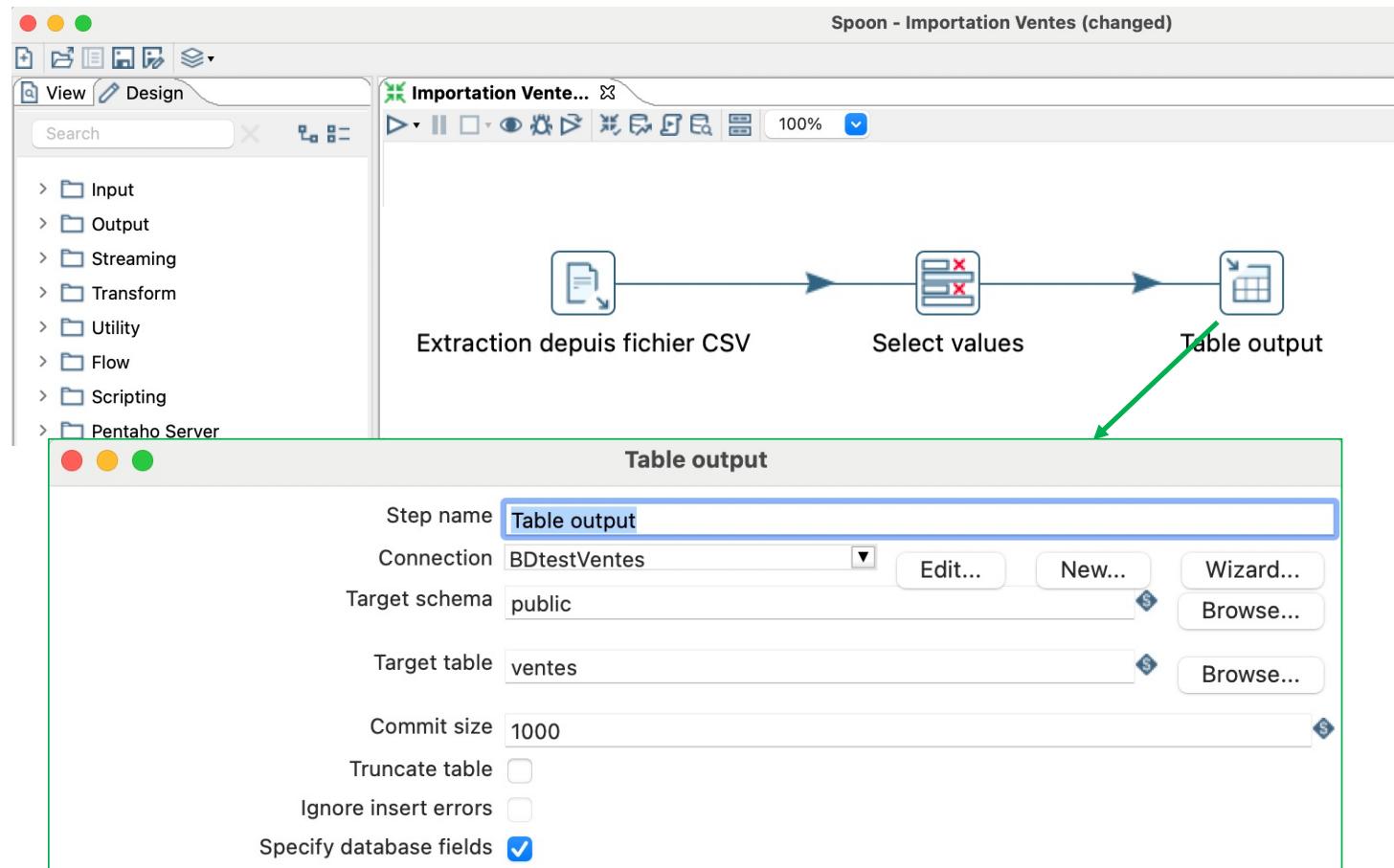
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Date	Date	yyyy-MM-...			\$,		none
2	Produit	String		10		\$,		none
3	Region	String		16		\$,		none
4	Ventes	Integer	#	15	0	\$,		none

At the bottom of the dialog are buttons for Help, OK, Get Fields, Preview, and Cancel.

On the far left, a small window titled "ventes.csv" displays the following CSV data:

```
1 Date;Produit;Region;Ventes
2 2024-01-01;Ordinateur;Europe;500
3 2024-01-01;Smartphone;Amérique du Nord;300
4 2024-01-02;Ordinateur;Asie;400
5 2024-01-03;Smartphone;Europe;200
6 2024-01-03;Tablette;Europe;150
7 2024-01-03;Ordinateur;Amérique du Nord;350
8 2024-01-04;Tablette;Asie;220
9 2024-01-04;Smartphone;Europe;180
```

Exemple de Transformation fichier CSV->Table postGres



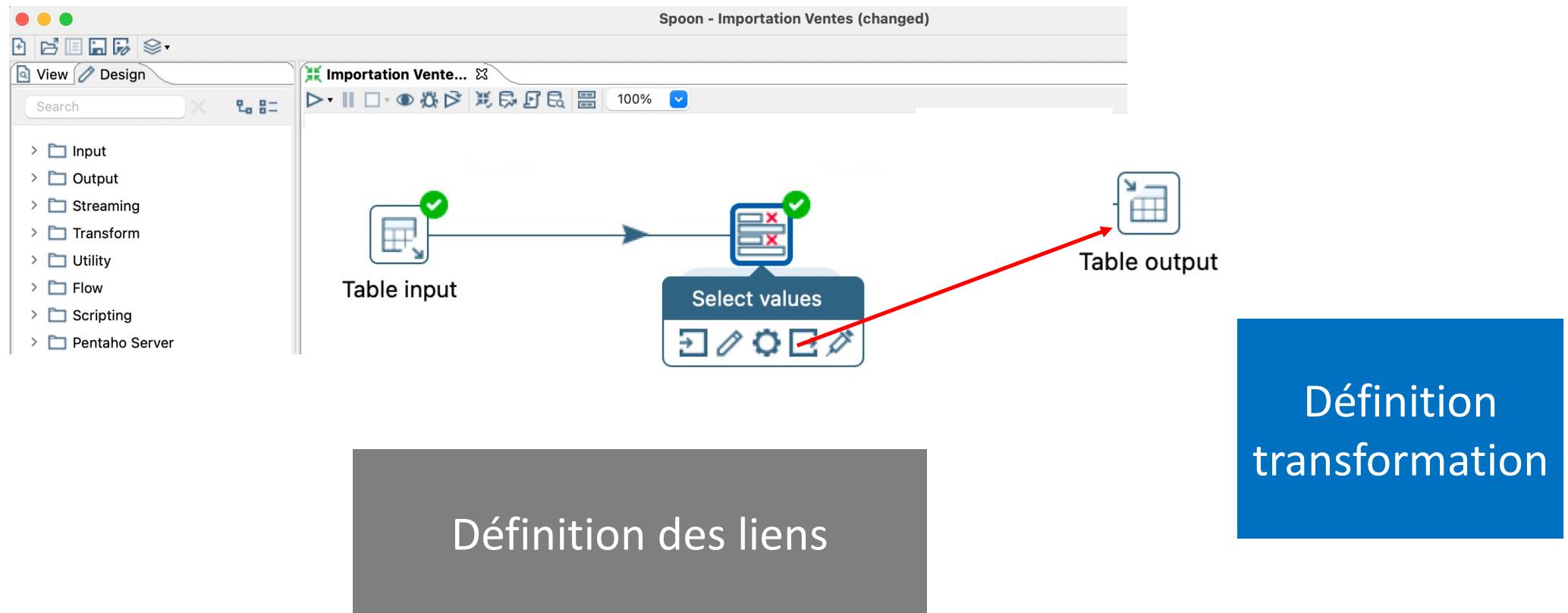
Définition
output

table vente
postgreSQL

Data Output Messages Notifications

	produit	region	ventes	datev
1	Ordinateur	Europe	500	2024-01-01
2	Smartphone	Amérique du Nord	300	2024-01-01
3	Ordinateur	Asie	400	2024-01-02
4	Smartphone	Europe	200	2024-01-03
5	Tablette	Europe	150	2024-01-03
6	Ordinateur	Amérique du Nord	350	2024-01-03
7	Tablette	Asie	220	2024-01-04
8	Smartphone	Europe	180	2024-01-04

Exemple de Transformation fichier CSV->Table postGres



Exemple de Transformation fichier CSV->Table postGres

Spoon - Importation Ventes (changed)

The screenshot shows a transformation named "Importation Vente...". The flow consists of three steps: "Extraction depuis fichier CSV" (CSV Input), "Select values" (highlighted with a blue arrow), and "Table output" (Postgres Output). The "Select values" step is currently selected, as indicated by the blue border around its name in the top bar and the blue step name in the dialog below.

Select values

Step name **Select values**

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	Produit	produit		
2	Region	region		
3	Ventes	ventes		
4	Date	datev		

Get fields to select

Edit Mapping

Source fields:

Target fields:

Auto target selection? Hide assigned source fields?

Auto source selection? Hide assigned target fields?

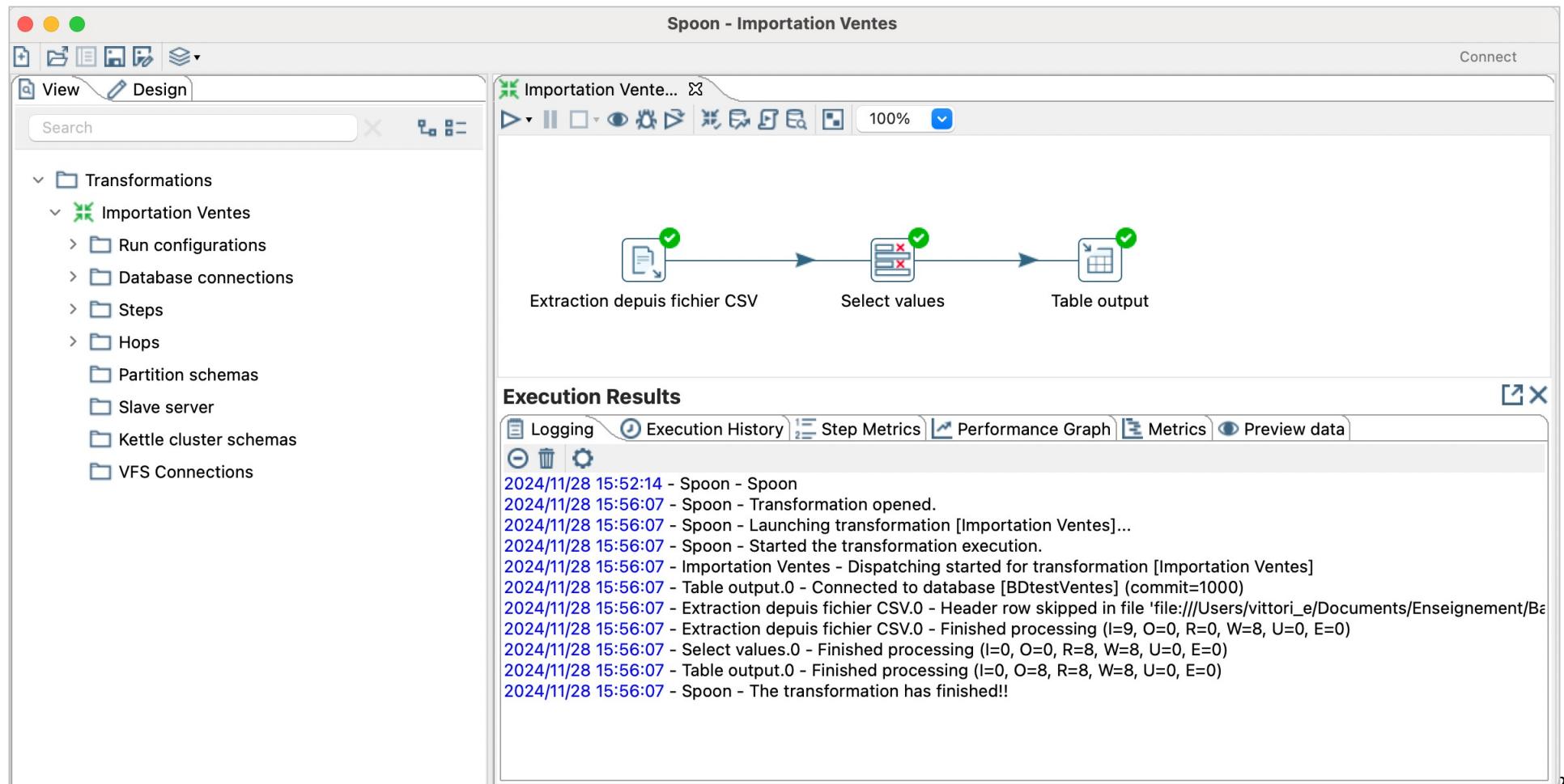
Add Delete OK Guess Cancel

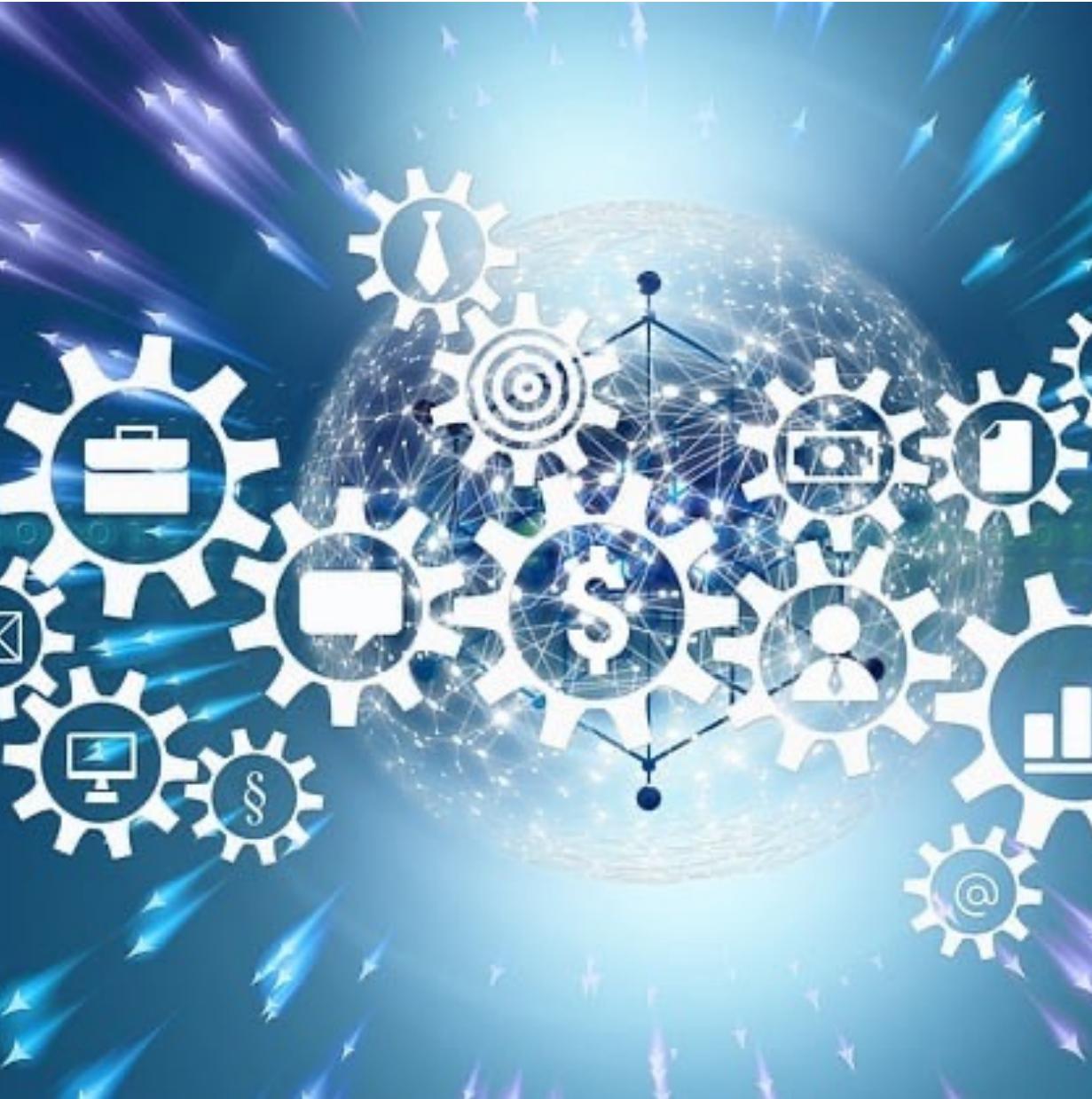
Mappings:

Produit	(Extraction depuis fichier CSV) --> produit
Region	(Extraction depuis fichier CSV) --> region
Ventes	(Extraction depuis fichier CSV) --> ventes
Date	(Extraction depuis fichier CSV) --> datev

Définition transformation

Exécution d'une transformation





4 – Processus ETL avec source BD PostgreSQL

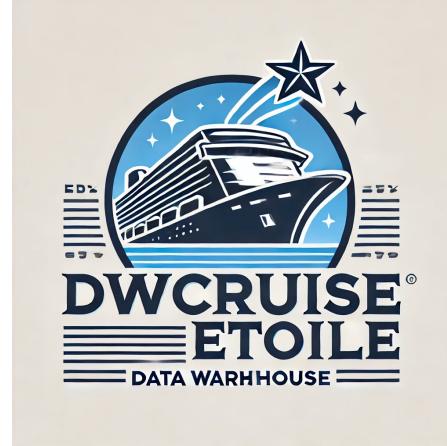


Exemple : DWCrusesEtoile

- scripts SQL à récupérer sur l'ENT :
 - BD source (schéma et données) : *creationBDSkippercruises.sql*
 - DW destination (schéma uniquement) : *creationDWETOILE.sql*



skippercruises



DWCruisesEtoile

BD source : BDskipperCruises



table	name
primary key	id_paiement serial
foreign key	id_reservation integer
column	montant_paiement numeric(10,2)
column	date_paiement date
column	mode_paiement character varying(50)

table	name
primary key	id_client serial
column	nom character varying(100)
column	age integer
column	nationalite character varying(50)

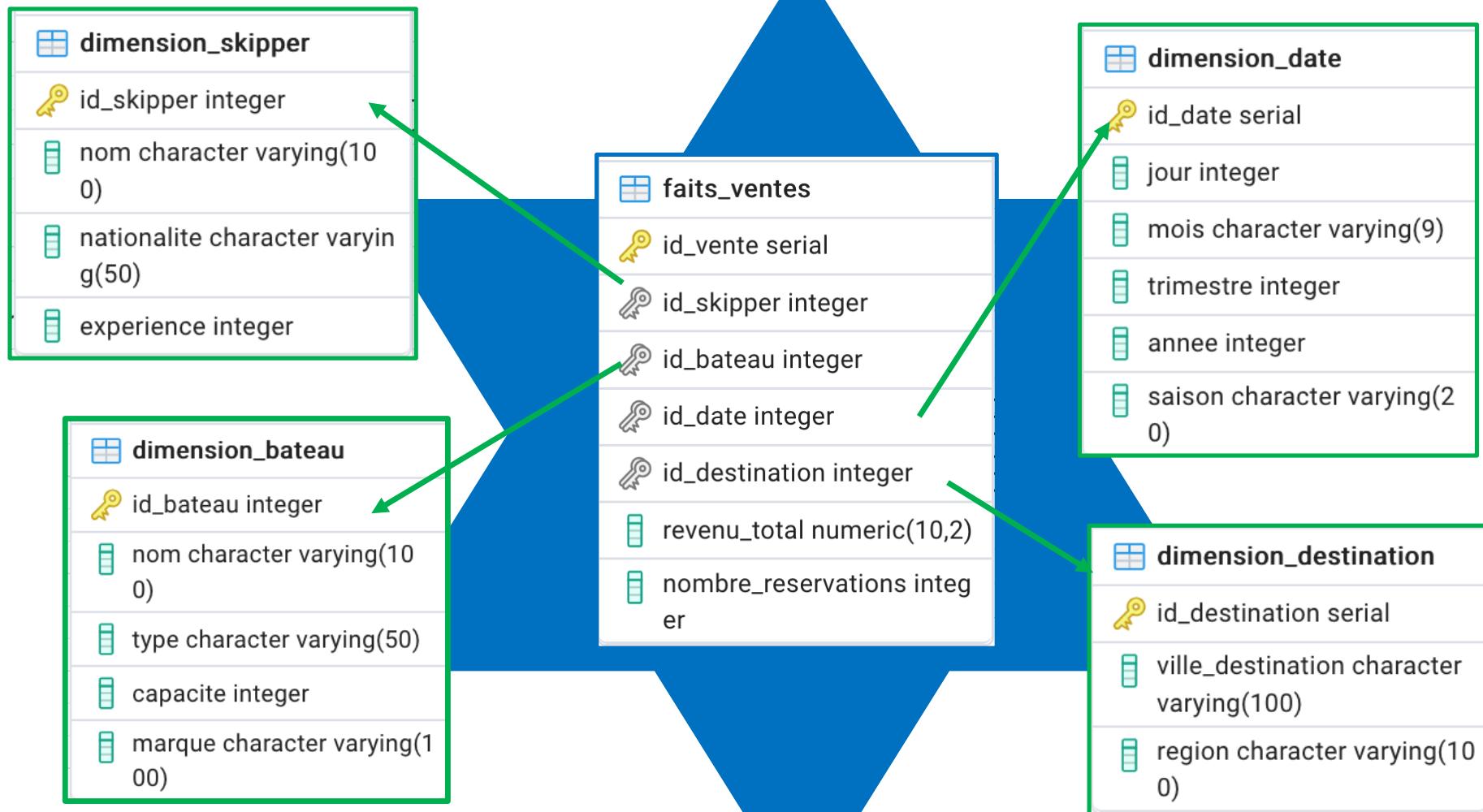
table	name
primary key	id_reservation serial
foreign key	id_client integer
foreign key	id_croisiere integer
column	montant_options numeric(10,2)
column	date_reservation date

table	name
primary key	id_croisiere serial
column	date_depart date
column	date_arrivee date
column	id_skipper integer
column	id_bateau integer
column	ville_depart character varying(100)
column	ville_destination character varying(100)
column	prix_base numeric(10,2)

table	name
primary key	id_bateau serial
column	nom character varying(100)
column	type character varying(50)
column	marque character varying(100)

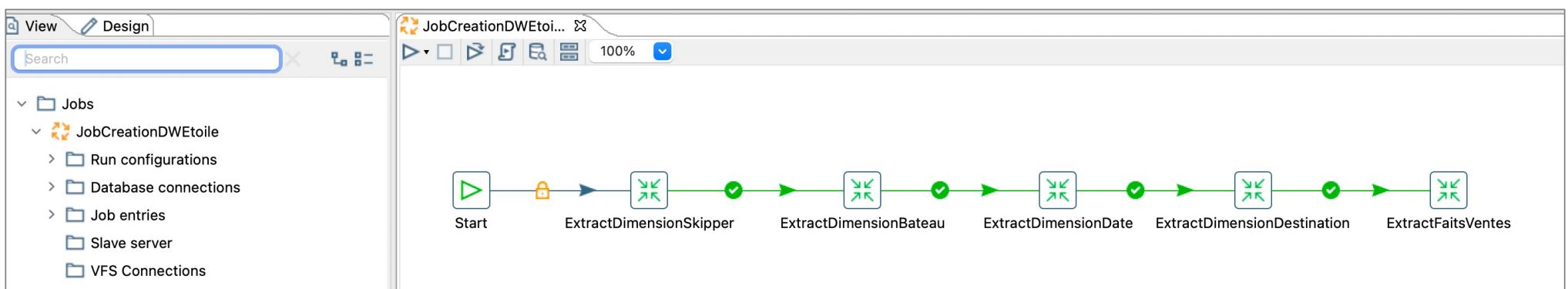
table	name
primary key	id_skipper serial
column	nom character varying(100)
column	experience integer
column	nationalite character varying(50)

DW Destination : DWCrusesEtoile

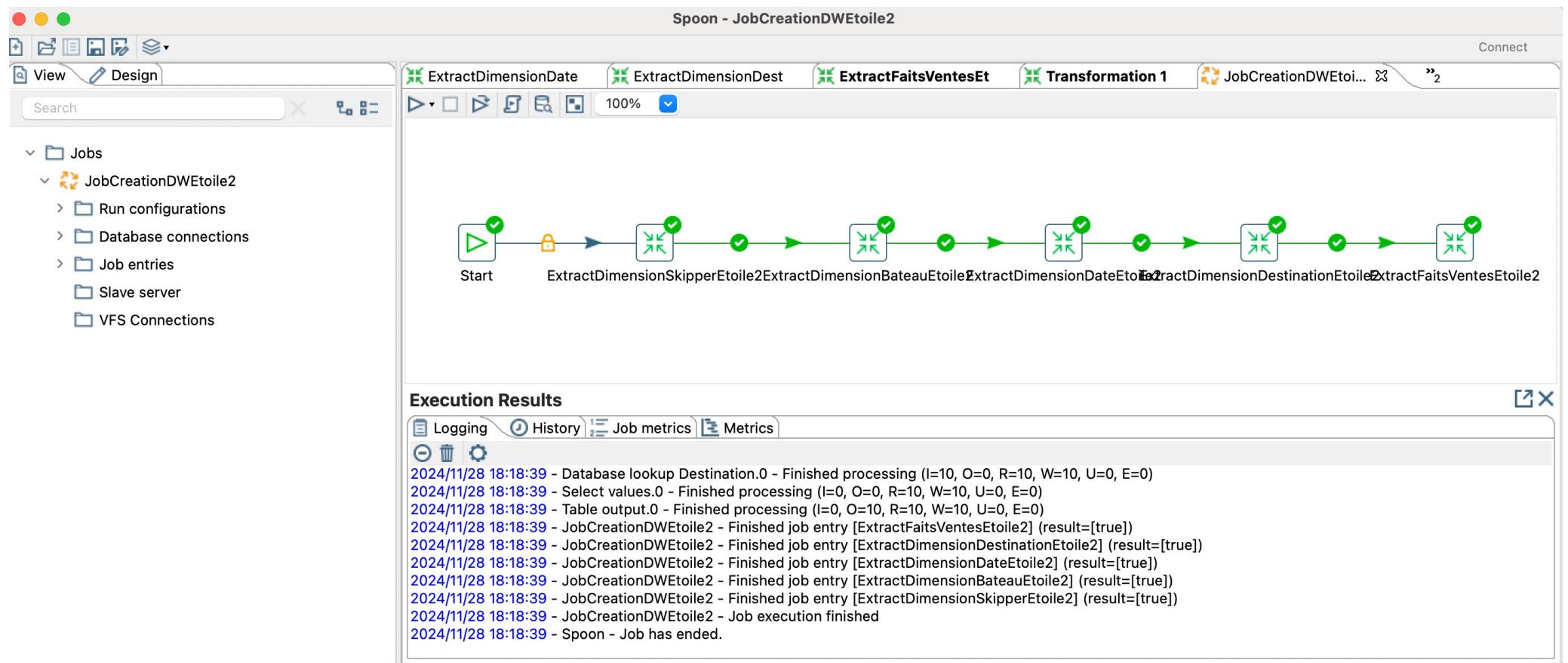


Processus ETL complet : Création d'un job

- 5 transformations + 1 job à définir :
 - ExtractDimensionSkipper
 - ExtractDimensionBateau
 - ExtractDimensionSkipper
 - ExtractDimensionDate
 - ExtractFaitsVentes



Exécution du processus (job)



ExtractDimensionBateau – Table Input

The image shows the Spoon interface for Apache Nifi. At the top, there's a toolbar with various icons. Below it is a tab bar with three tabs: "ExtractDimensionSkip", "ExtractDimensionDate", and "ExtractDimensionDateEtoile2" (the latter is selected). The main workspace displays a flow diagram with three components: "Table input", "Select values", and "Table output", connected sequentially by arrows. A red arrow points from the "Table input" component to a detailed configuration window below.

Table input

Step name: **Table input**

Connection: DBskippercruises

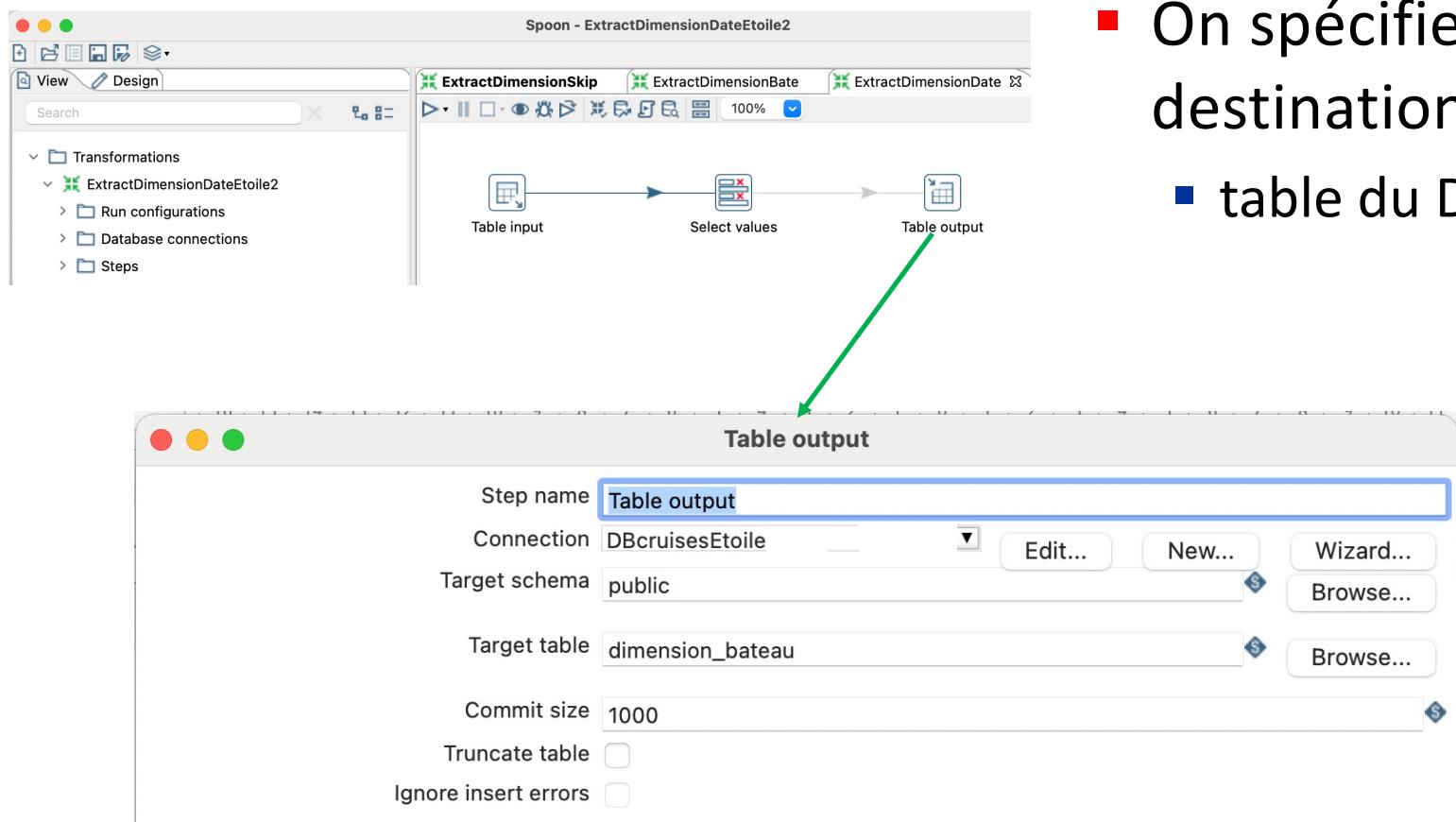
SQL:

```
SELECT id_bateau, nom, type, capacite, marque FROM bateaux
```

On utilise une requête SQL pour récupérer les colonnes nécessaires

-

ExtractDimensionBateau – Table Output



- On spécifie la table destination
 - table du DWcruisesEtoile

dimension_bateau	
	id_bateau integer
	nom character varying(100)
	type character varying(50)
	capacite integer
	marque character varying(100)

ExtractDimensionBateau – Select Values

The screenshot shows the Apache Nifi Spoon interface. On the left, the file tree displays a transformation named 'ExtractDimensionDateEtoile2' containing a single 'Select values' step. The main canvas shows a flow from 'Table input' to 'Select values' to 'Table output'. A blue arrow points from the 'Select values' step on the canvas to a detailed configuration dialog window titled 'Select values'.

Select values Dialog:

- Step name: Select values
- Fields:

#	Fieldname	Rename to	Length	Precision
1	id_bateau			
2	nom			
3	type			
4	capacite			
5	marque			

- Buttons: Get fields to select, Edit Mapping
- Bottom buttons: OK, Cancel

Mapping Dialog (Visible at the Bottom):

- Source fields: (Empty)
- Target fields: (Empty)
- Mappings:

Fieldname	Description
id_bateau	(Table input) --> id_bateau
nom	(Table input) --> nom
type	(Table input) --> type
capacite	(Table input) --> capacite
marque	(Table input) --> marque

- Buttons: Add, Delete, OK, Guess, Cancel
- Checkboxes: Auto target selection? (checked), Hide assigned source fields? (checked), Auto source selection? (unchecked), Hide assigned target fields? (checked)

ExtractDimensionDate – Table Input

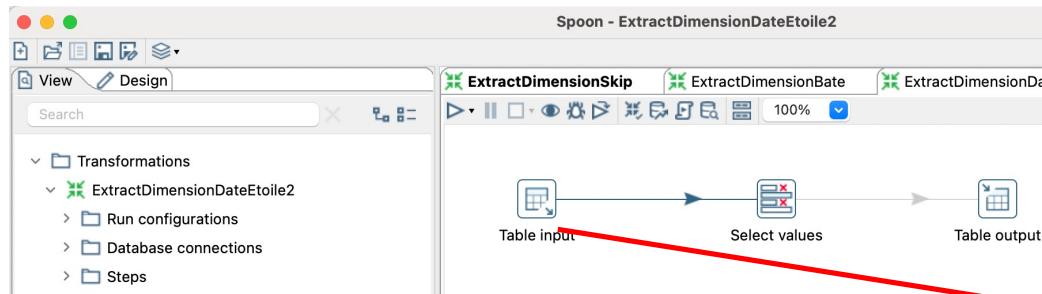


Table input

Step name: Table input
Connection: DBskippercruises

```

SELECT DISTINCT
    EXTRACT(DAY FROM date_depart) AS jour,
    EXTRACT(YEAR FROM date_depart) AS annee,
    EXTRACT(QUARTER FROM date_depart) AS trimestre,
    CASE
        WHEN EXTRACT(MONTH FROM date_depart) IN (12, 1, 2) THEN 'Hiver'
        WHEN EXTRACT(MONTH FROM date_depart) IN (3, 4, 5) THEN 'Printemps'
        WHEN EXTRACT(MONTH FROM date_depart) IN (6, 7, 8) THEN 'Eté'
        WHEN EXTRACT(MONTH FROM date_depart) IN (9, 10, 11) THEN 'Automne'
        ELSE 'Inconnu'
    END AS saison,
    CASE
        WHEN EXTRACT(MONTH FROM date_depart) = 1 THEN 'janvier'
        WHEN EXTRACT(MONTH FROM date_depart) = 2 THEN 'février'
        WHEN EXTRACT(MONTH FROM date_depart) = 3 THEN 'mars'
        WHEN EXTRACT(MONTH FROM date_depart) = 4 THEN 'avril'
        WHEN EXTRACT(MONTH FROM date_depart) = 5 THEN 'mai'
        WHEN EXTRACT(MONTH FROM date_depart) = 6 THEN 'juin'
        WHEN EXTRACT(MONTH FROM date_depart) = 7 THEN 'juillet'
        WHEN EXTRACT(MONTH FROM date_depart) = 8 THEN 'août'
        WHEN EXTRACT(MONTH FROM date_depart) = 9 THEN 'septembre'
        WHEN EXTRACT(MONTH FROM date_depart) = 10 THEN 'octobre'
        WHEN EXTRACT(MONTH FROM date_depart) = 11 THEN 'novembre'
        WHEN EXTRACT(MONTH FROM date_depart) = 12 THEN 'décembre'
        ELSE 'Inconnu'
    END AS mois_lettres
FROM croisieres;

```

Line 28 Column 40

Store column info in step meta data Enable lazy conversion Replace variables in script? Insert data from step Execute for each row? Limit size: 0

OK Preview Cancel

BDSkipperCruises

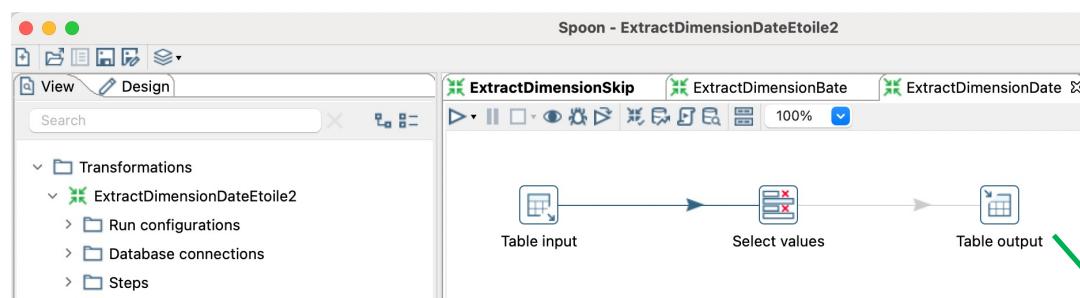
- croisieres
- id_croisiere serial
- date_depart date
- date_arrivee date
- id_skipper integer
- id_bateau integer
- ville_depart character varying(100)
- ville_destination character varying(100)
- prix_base numeric(10,2)

DWcruisesEtoile

- dimension_date
- id_date serial
- jour integer
- mois character varying(9)
- trimestre integer
- annee integer
- saison character varying(20)

- On utilise une requête SQL pour
 - récupérer les colonnes nécessaires :
 - jour, année, trimestre
 - en créer d'autres
 - saison, mois_lettre

ExtractDimensionDate – Table Output



dimension_date
id_date serial
jour integer
mois character varying(9)
trimestre integer
annee integer
saison character varying(2 0)

- On spécifie la table destination
 - table du DWcruisesEtoile

The screenshot shows the 'Table output' configuration dialog. The 'Step name' field is set to 'Table output'. The 'Connection' dropdown is set to 'DBcruisesEtoile'. The 'Target schema' is 'public'. The 'Target table' is 'dimension_date'. Other settings include 'Commit size' at 1000, and 'Truncate table', 'Ignore insert errors', and 'Specify database fields' checkboxes.

Table output	
Step name	Table output
Connection	DBcruisesEtoile
Target schema	public
Target table	dimension_date
Commit size	1000
Truncate table	<input type="checkbox"/>
Ignore insert errors	<input type="checkbox"/>
Specify database fields	<input type="checkbox"/>

ExtractDimensionDate – Select values

Spoon - ExtractDimensionDateEtoile2

ExtractDimensionSkip ExtractDimensionDate ExtractDimensionDate Etoile2

Table input Select values Step name Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	jour			
2	annee			
3	trimestre			
4	saison			
5	mois_lettres			

Get fields to select Edit Mapping

Include unspecified fields, ordered by name

Help OK Cancel

Source fields: Target fields: id_date

Auto target selection? Auto source selection?
Hide assigned source fields? Hide assigned target fields?

Mappings:

jour	(Table input) --> jour
annee	(Table input) --> annee
trimestre	(Table input) --> trimestre
saison	(Table input) --> saison
mois_lettres	(Table input) --> mois

OK Guess Cancel

■ On utilise select value pour définir les correspondances entre l'élément Table Input et Table Output

ExtractFaitsVentes



Objectif : collecter les données nécessaires pour remplir la table FaitsVentes du DW

faits_ventes
id_vente serial
id_skipper integer
id_bateau integer
id_date integer
id_destination integer
revenu_total numeric(10,2)
nombre_reservations integer

Une ligne dans cette table correspond à une croisière

Somme calculée à partir des tables croisières et réservations

Comptage calculé à partir des tables croisières et réservations

ExtractFaitsVentes – Table Input



Requête sur BD skippercruises

- id_croisiere
- id_skipper
- id_bateau
- ville_destination
- année de la date_départ
- jour de la date_départ
- mois de la date_départ en lettres
- calcul du revenu_total(prix_base + montant_options)
- comptage du nombre de réservations

- Jointure avec la table dimension_destination du DW pour récupérer la clé id_dimension correspondant à année,mois, jour

- Jointure avec la table dimension_date du DW pour récupérer la clé id_date correspondant à année,mois, jour

ExtractFaitsVentes – Table Input

Table input

Step name **Table input**

Connection DBskippercruises

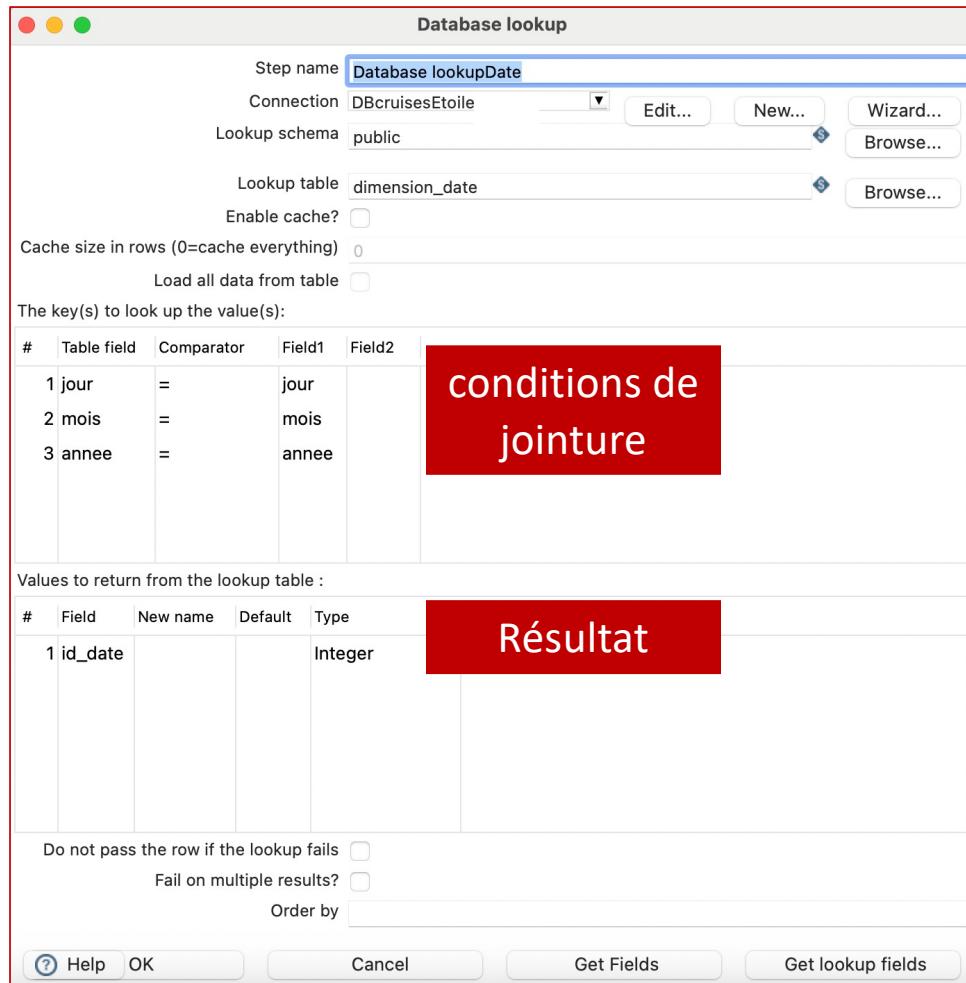
SQL

```
SELECT
    c.id_croisiere,
    c.id_skipper,
    c.id_bateau,
    c.ville_destination,
    EXTRACT(DAY FROM date_depart) AS jour,
    CASE
        WHEN EXTRACT(MONTH FROM date_depart) = 1 THEN 'janvier'
        WHEN EXTRACT(MONTH FROM date_depart) = 2 THEN 'février'
        WHEN EXTRACT(MONTH FROM date_depart) = 3 THEN 'mars'
        WHEN EXTRACT(MONTH FROM date_depart) = 4 THEN 'avril'
        WHEN EXTRACT(MONTH FROM date_depart) = 5 THEN 'mai'
        WHEN EXTRACT(MONTH FROM date_depart) = 6 THEN 'juin'
        WHEN EXTRACT(MONTH FROM date_depart) = 7 THEN 'juillet'
        WHEN EXTRACT(MONTH FROM date_depart) = 8 THEN 'août'
        WHEN EXTRACT(MONTH FROM date_depart) = 9 THEN 'septembre'
        WHEN EXTRACT(MONTH FROM date_depart) = 10 THEN 'octobre'
        WHEN EXTRACT(MONTH FROM date_depart) = 11 THEN 'novembre'
        WHEN EXTRACT(MONTH FROM date_depart) = 12 THEN 'décembre'
        ELSE 'Inconnu'
    END AS mois,
    EXTRACT(YEAR FROM date_depart) AS annee,
    COUNT(r.id_reservation) AS nombre_reservations,
    SUM(r.montant_options + c.prix_base) AS revenu_total
FROM
    croisières c
JOIN
    réservations r ON c.id_croisiere = r.id_croisiere
GROUP BY
    c.id_croisiere, c.id_skipper, c.id_bateau, c.ville_destination, c.date_depart;
```

Résultat = 1 seule ligne par croisière

- id_croisiere
- id_skipper
- id_bateau
- ville_destination
- année de la date_départ
- jour de la date_départ
- mois de la date_départ en lettres
- calcul du revenu_total(prix_base + montant_options)
- comptage du nombre de réservations

ExtractFaitsVentes – Database lookup Date



- Pour chaque ligne décrivant une croisière (table résultat de la requête Table Input *req*)
 - On cherche la ligne de la table dimension_date avec jour=*req.jour*
mois=*req.mois*
annee= *req.année*
 - On obtient id_date (la clé dans la table dimension_date) :
 - une colonne id_date est ajoutée à la table *req*

ExtractFaitsVentes – Database lookup Dimension

Database lookup

Step name: Database lookup Destination

Connection: DBcruisesEtoile

Lookup schema: public

Lookup table: dimension_destination

Enable cache?

Cache size in rows (0=cache everything): 0

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	ville_destination	=	ville_destination	

condition de jointure

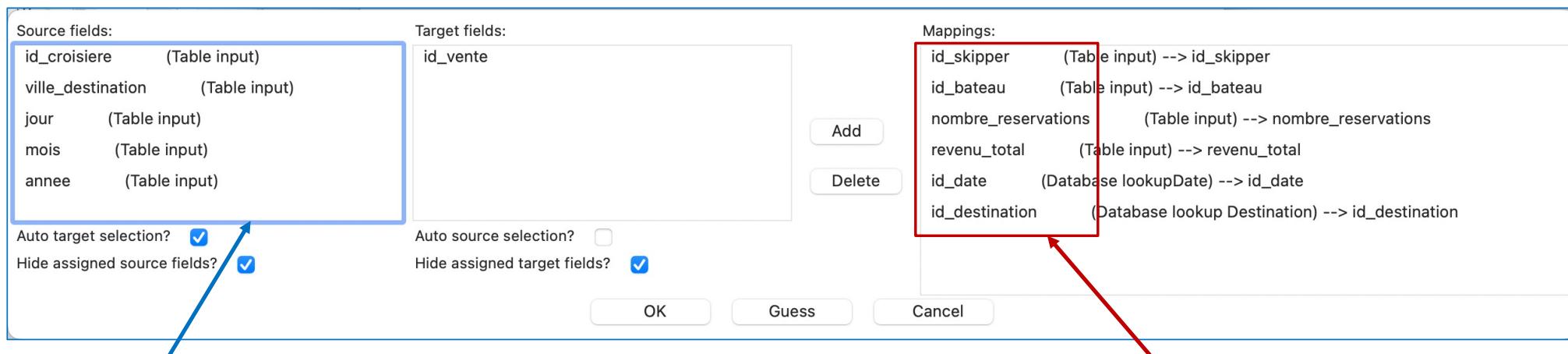
Values to return from the lookup table :

#	Field	New name	Default	Type
1	id_destination			Integer

Résultat

- Pour chaque ligne décrivant une croisière (résultat du lookup précédent)
 - On cherche la ligne de la table dimension_destination avec ville_destination=req.ville_destination
- On obtient id_destination (la clé dans la table dimension_destination)
 - une colonne id_dimension est ajoutée à la table req

ExtractFaitsVentes – select Values



colonnes de la source qui ne sont plus utilisées

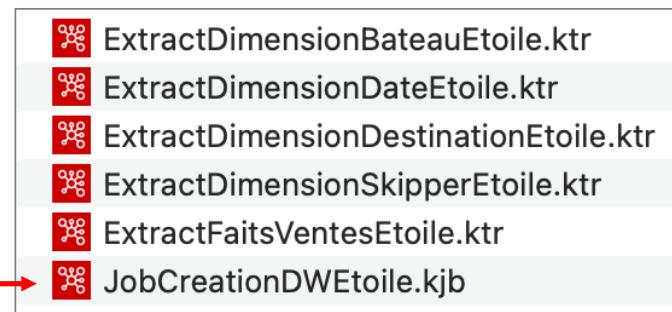
On ne garde que les colonnes nécessaires

colonnes de la source qui sont "mappées" vers la table output (FaitsVentes)

A vous de jouer (1) ...

Exécution Processus ETL DWcruisesEtoile

- Récupérez sur l'ENT les fichiers d'implémentation PDI du processus ETL associé à la BD DWcruisesEtoile
 - fichier *ProcessusDIDWcruisesEtoile.zip* contenant
 - les transformations PDI
 - et le job
- Modifiez les paramètres des sources de données afin de les mettre en conformité avec votre BD source skippercruises et votre DW DWcruisesEtoile
- Exécuter le job PDI pour alimenter votre DW



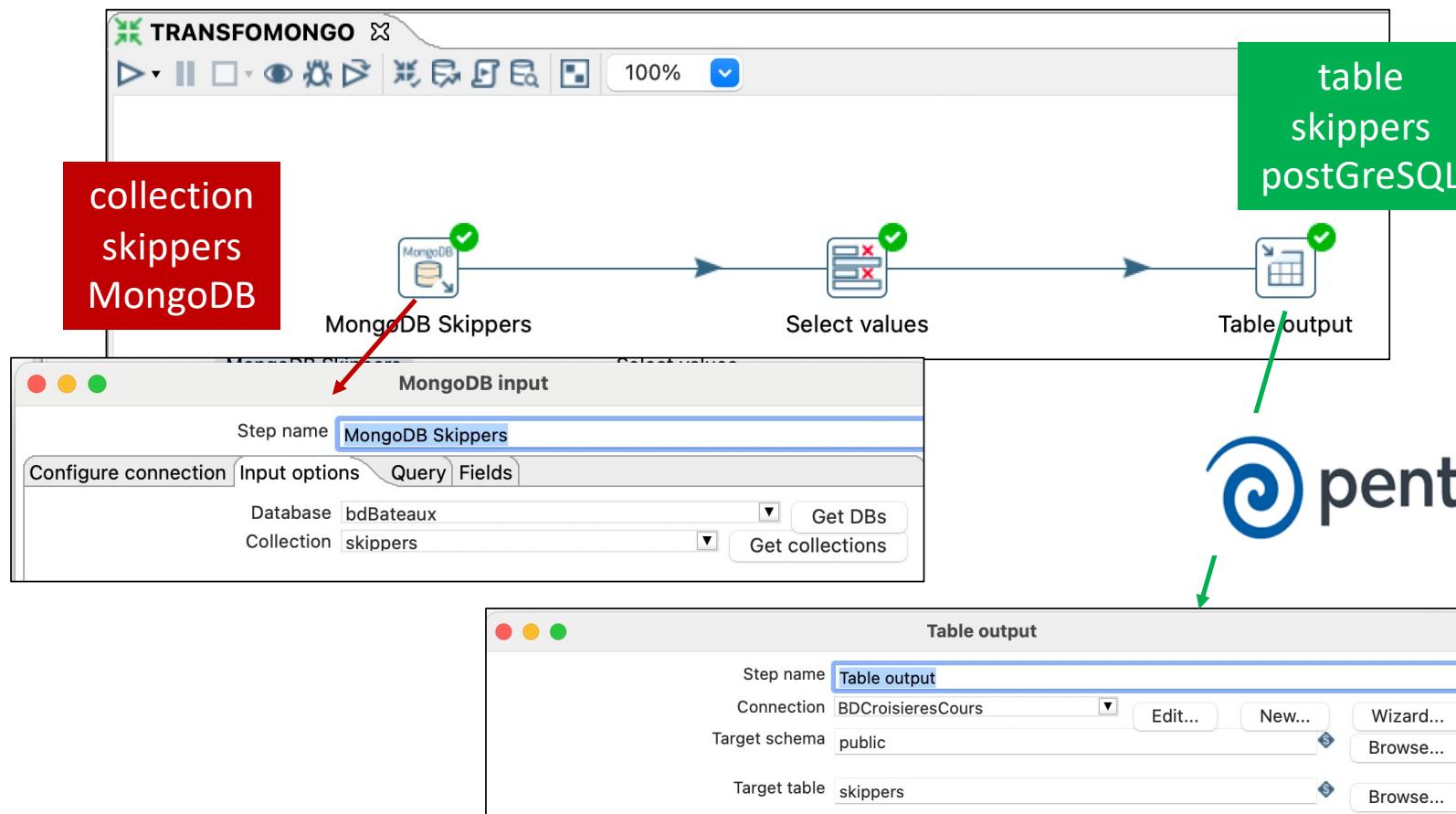
Le script SQL de creation de la BD skippercruises est disponible sur l'ENT



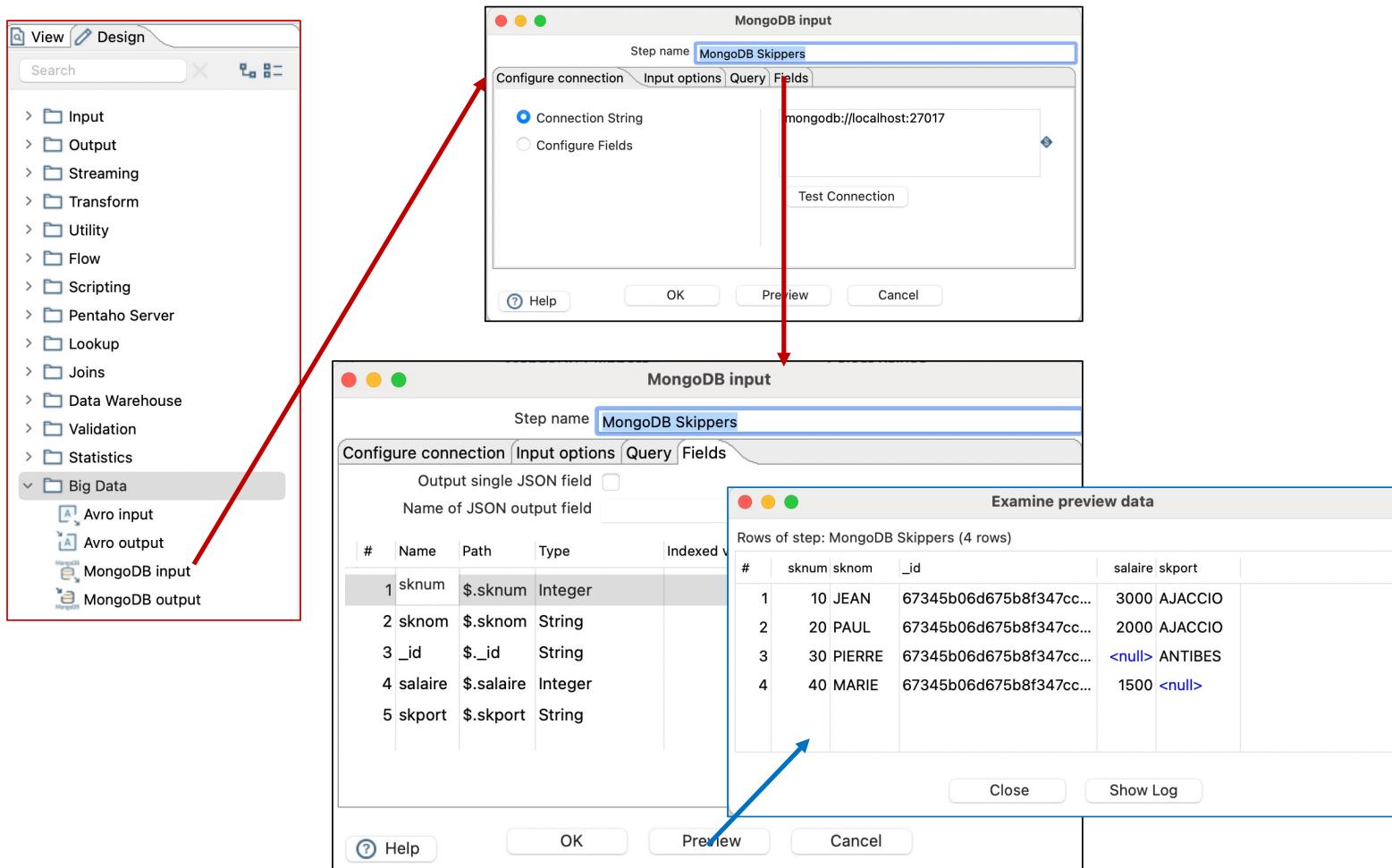
5 – Processus ETL avec source BD mongoDB



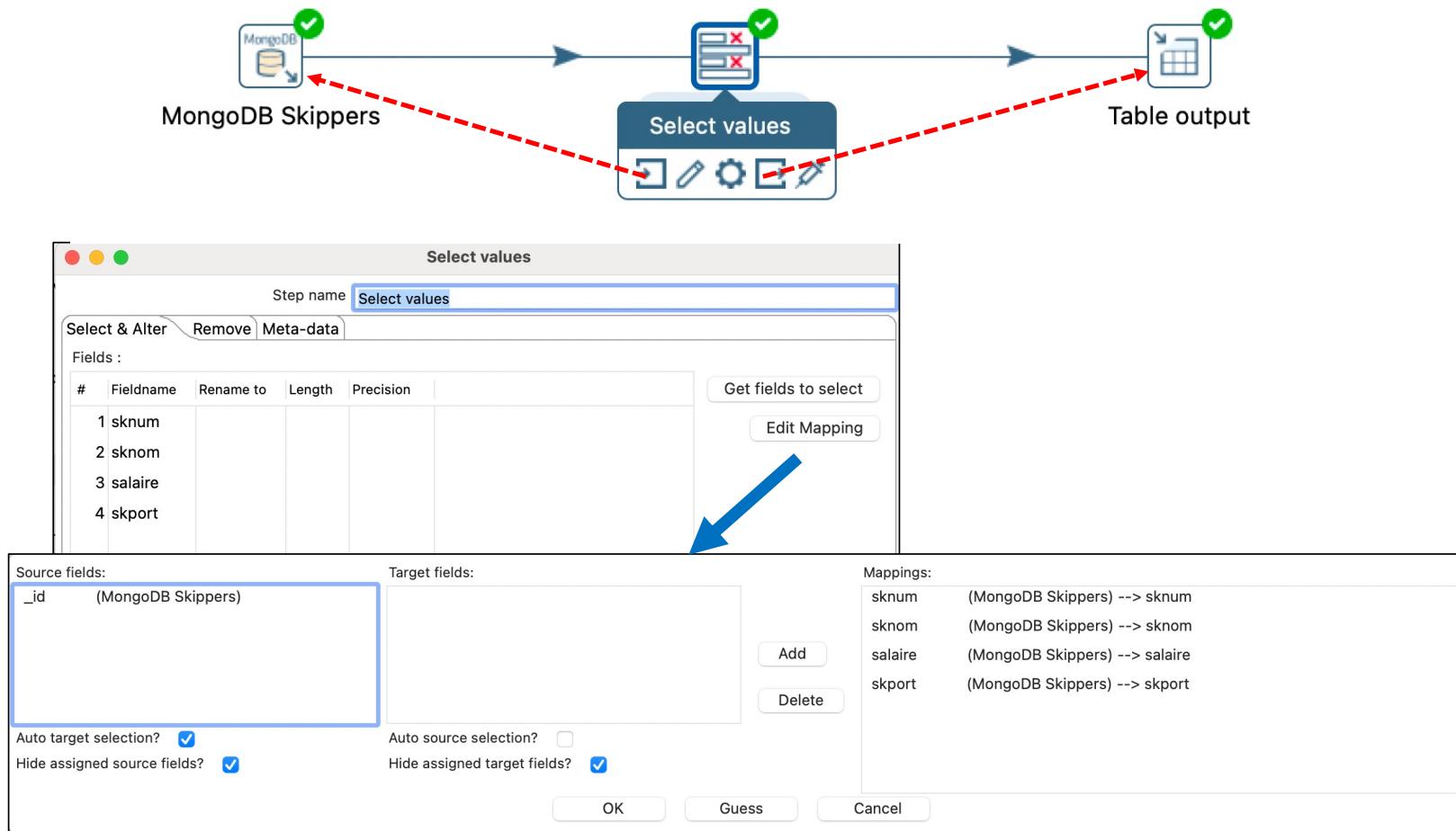
Exemple de transformation PDI : MongoDB -> PostgreSQL



Spécification MongoDB Input



Spécification Select Values



A vous de jouer (2) ...

Implémentation du processus ETL associé à DWCrusesPerf

- Implémentez le processus ETL d'alimentation du datawarehouse DWCrusesPerf défini en exercice dans le chapitre précédent.
 - Ce DW est alimenté par la BD postgresSQL skippercruises
- Vérifiez sous pgadmin que les tables de votre DW ont bien été remplies.



Sources et liens à consulter

- Des cours intéressants
 - <https://slideplayer.fr/slide/1159105/>
 - <https://fr.slideshare.net/slideshow/cours-data-warehouse/239246576>
- Des exemples de transformations sous Pentaho DI
(exercices corrigés proposés par Stéphane Crozat)
 - <https://stph.scenari-community.org/contribs/dwh/PentahoDI/co/ex01.html>
 - <https://stph.scenari-community.org/contribs/dwh/PentahoDI/co/uc03.html>