



Institut Supérieur
des Technologies de l'Information
et de la Communication

République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Carthage
Institut Supérieur des Technologies de
l'Information et de la Communication



Mini-Projet-Datawarehouse

Rapport De Projet

Mise en œuvre d'un Data Mart en utilisant Pentaho Data Integration et MySQL

Membres

Fedi Bellakhel

Tutor: Mme Zeyneb Trabelsi

Date: 04/03/2024

Table des matière

Introduction Générale	I
Chapter 1: Préparation de la base de données en production	2
1.1 Introduction	2
1.2 Creation du base de donnés	3
Chapter 2: II. Conception du Datamart pour l'analyse du sujet ventes.	7
2.0.1 schéma en étoile	7
Chapter 3: Intégration des données	9
3.0.1 introduction	9
3.0.2 Intégration des données	9
3.0.3 Alimenter la datamart « VentesDM »	10
3.0.4 alimenter la table client	15
3.0.5 Alimenter la table dimension Temps	18
3.0.6 Alimenter la table fact Ventes	21
Chapter 4: Conclusion Générale	29

Introduction Générale

La création et la préparation d'une base de données en production ainsi que la conception d'un Datamart pour l'analyse des ventes constituent des étapes cruciales dans le processus d'analyse de données. L'intégration des données provenant de différentes sources est également une phase essentielle pour garantir la qualité et la cohérence des informations disponibles dans le Datamart. Dans ce document, nous avons abordé plusieurs aspects, notamment :

- **Préparation de la base de données en production :**

Cette section décrit les étapes nécessaires à la création et à la configuration d'une base de données adaptée à la production.

- **Conception du Datamart pour l'analyse du sujet des ventes :**

Cette partie présente le schéma en étoile utilisé pour organiser les données de manière optimale en vue de leur analyse.

- **Intégration des données :**

Cette étape comprend plusieurs sous-sections détaillant le processus d'intégration des données dans le Datamart, y compris l'alimentation des tables de dimension et de fait.

Chapter 1

Préparation de la base de données en production

1.1 Introduction

Soit le schéma de la base de données opérationnelle « Prod » suivant :

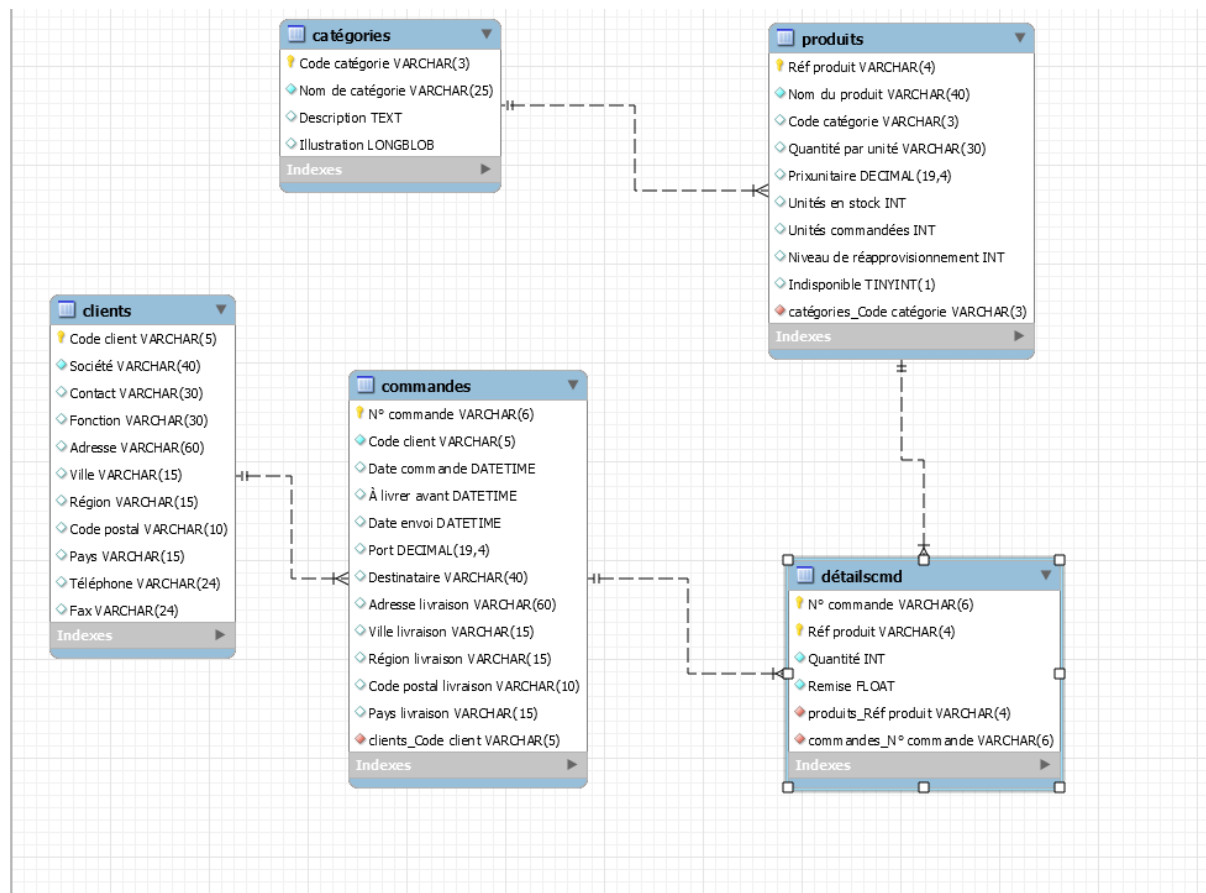


Figure 1.1: Diag

On veut à partir de cette base de données opérationnelle, mettre en œuvre un DW pour

étudier l'évolution de la quantité et le montant des Ventes selon plusieurs dimensions : Temps, Clients et Produits. L'objectif est de pouvoir analyser les ventes par jour et par mois. Les produits peuvent être regroupés en fonction de leurs catégories. Les clients peuvent être regroupés en fonction de leurs pays , mais il faut d'abord créer la base de données dans l'outil mysql workbench

1.2 Creation du base de donnés

Pour ce faire nous allons suivre les étapes suivantes :

- ouvrir MYSQL workbench

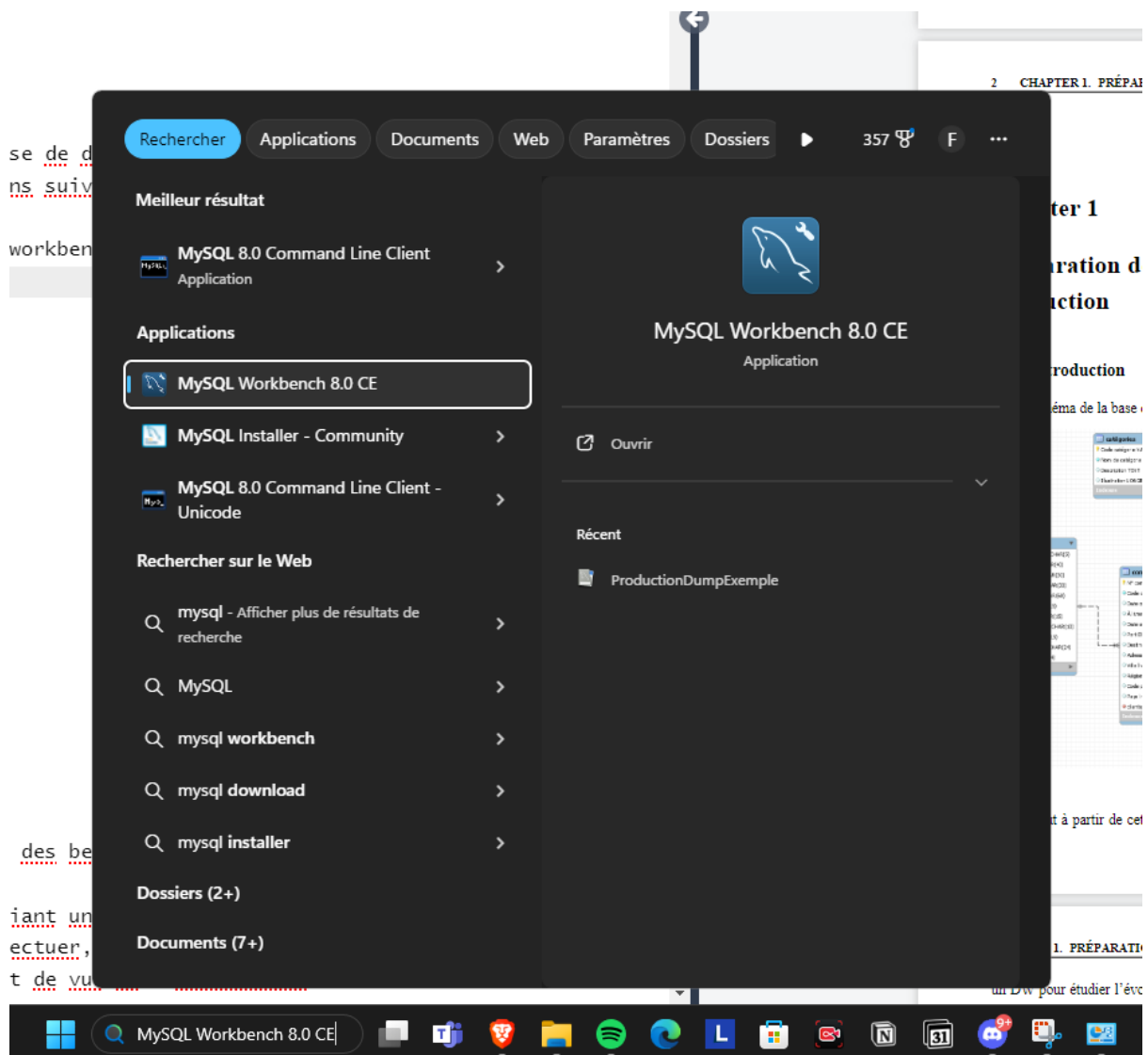


Figure 1.2: MYSQL workbench

- se connecter au serveur mysql

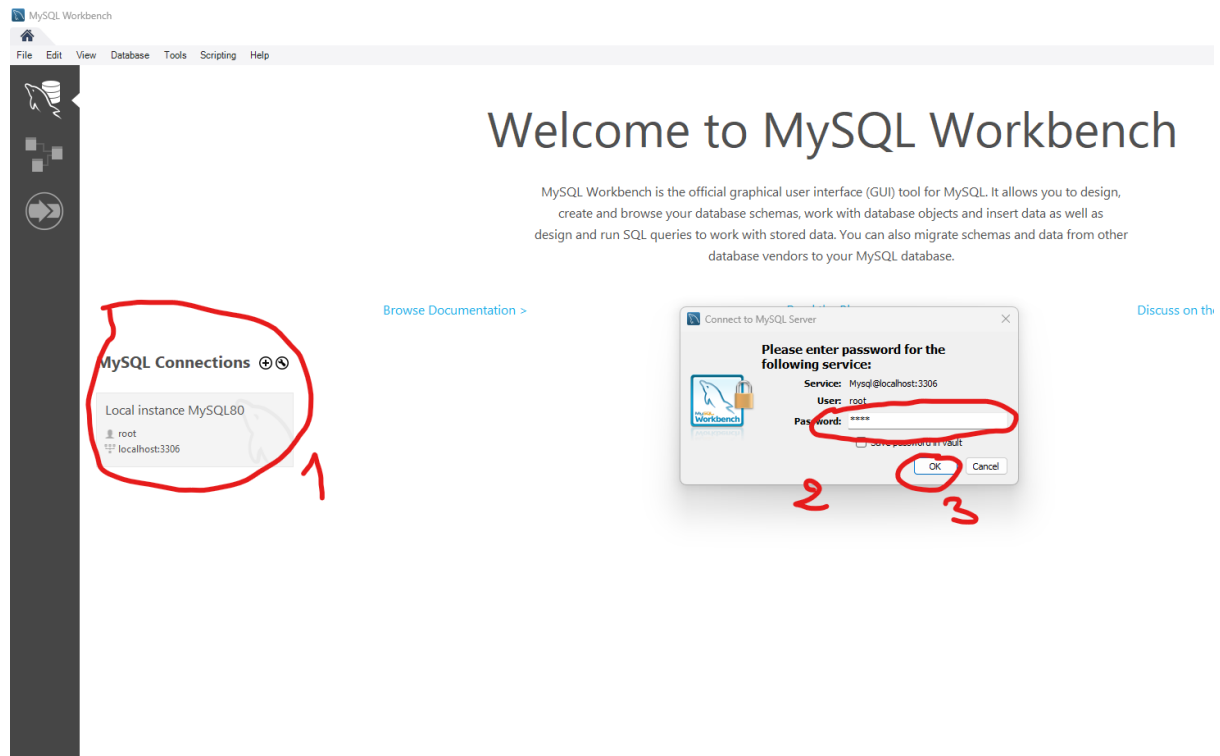


Figure 1.3: Notez bien : il faut utiliser les coordonnées du serveur créées lors de l'installation de MySQL.

- On crée la source de données à partir du fichier SQL : <ProductionDumpExemple.sql>. Appuyez sur "exécuter".
- le chemin de datasources sera afficher comme ca :

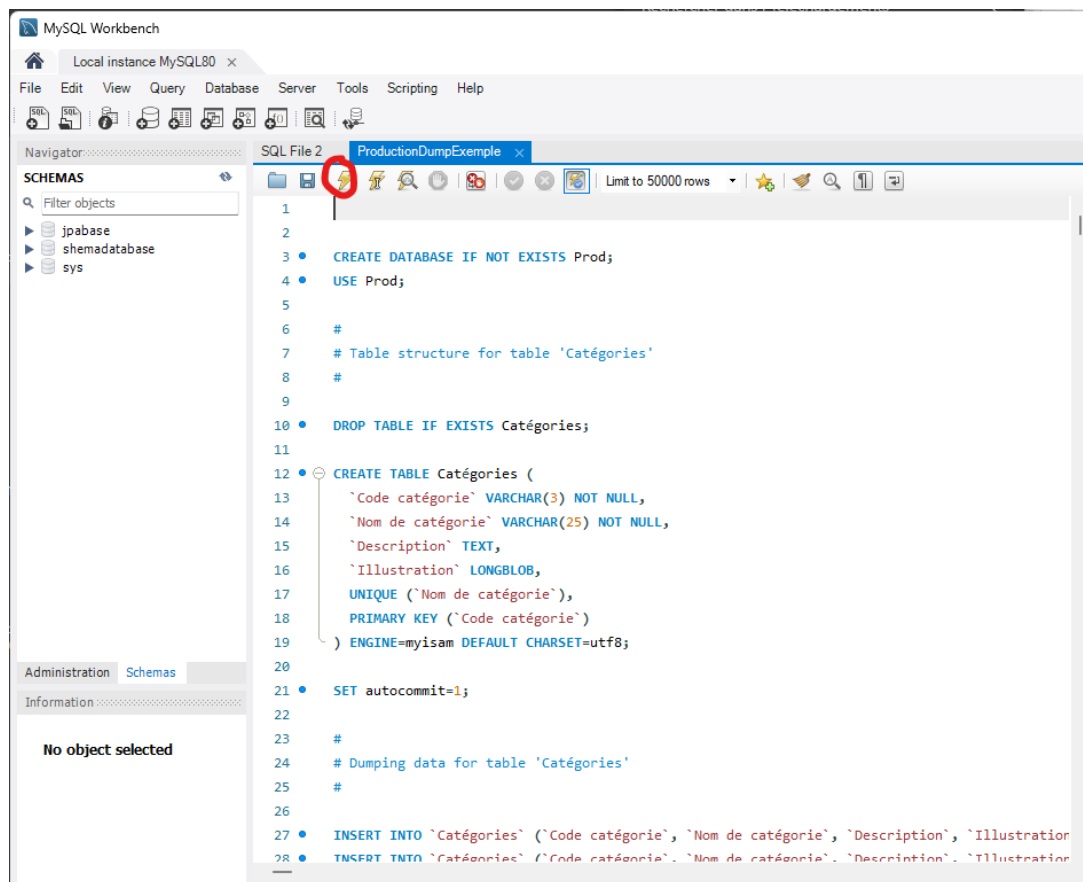


Figure 1.4: MYSQL workbench

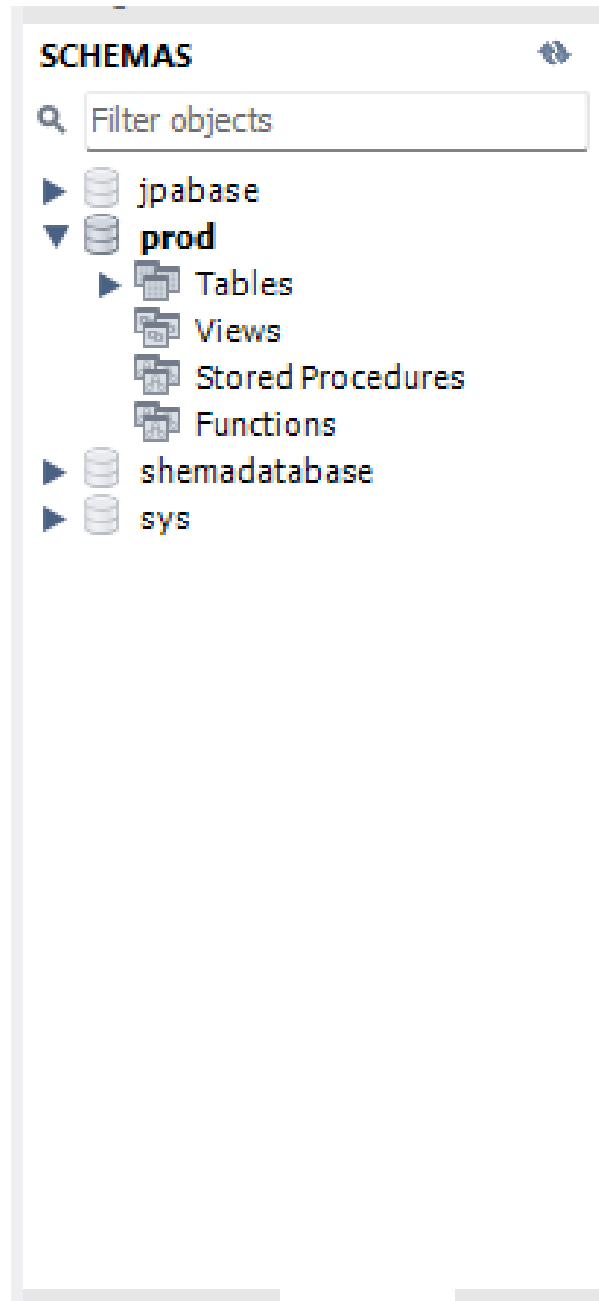


Figure 1.5: MYSQL workbench

Chapter 2

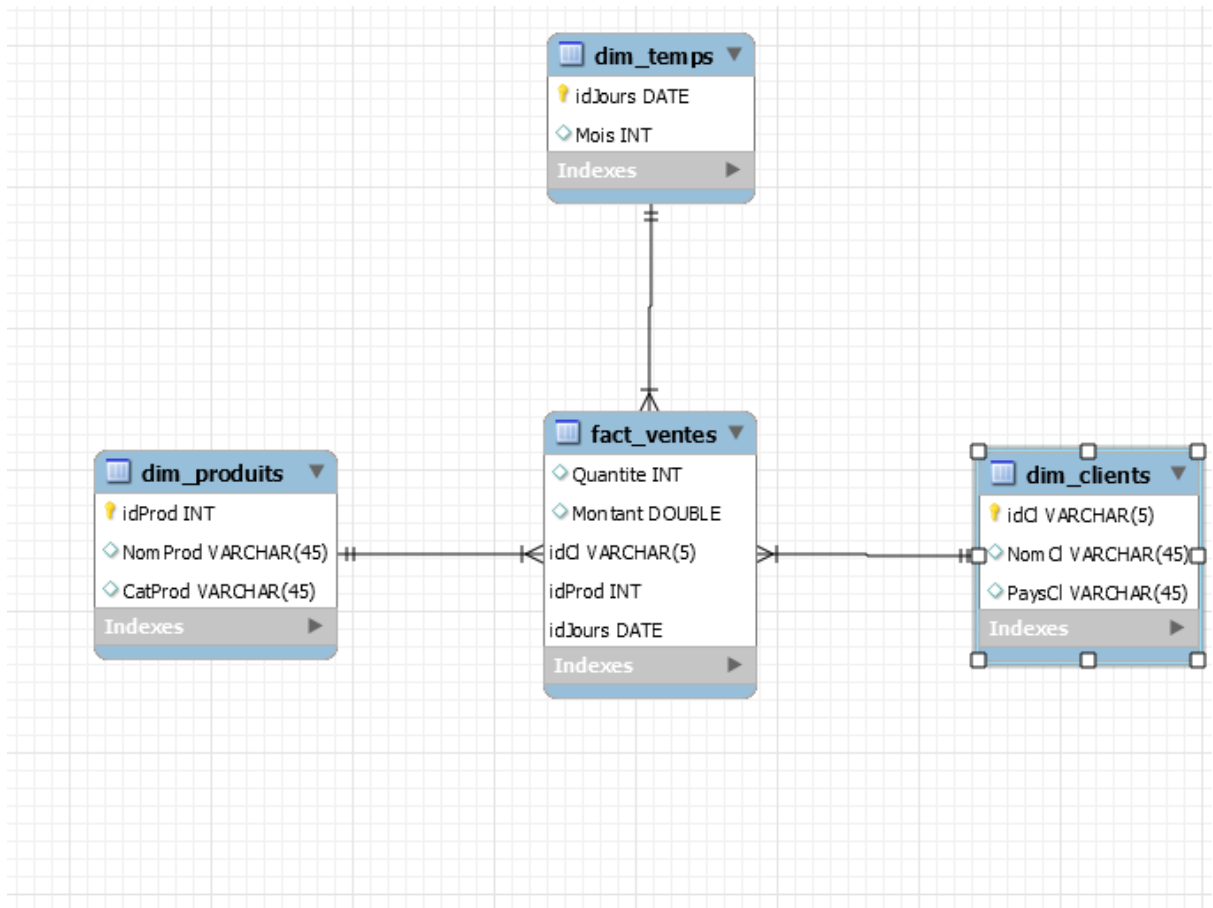
II. Conception du Datamart pour l'analyse du sujet ventes.

2.0.1 schéma en étoile

dans cette section on vas Proposer un schéma en étoile du datamart « ventesDM » pour étudier les faits de ventes.

Pour ce faire nous allons suivre les étapes suivantes :

- file » New Model
- copier et utiliser le code dans le fichier dump dans un script
- générer le script dump (Database » Forward engineer):



Chapter 3

Intégration des données

3.0.1 introduction

Dans cette section, nous allons alimenter ce datamart à travers le processus ETL en utilisant l'outil Pentaho Data Integration. Cependant, nous aurons besoin de quelques programmes supplémentaires pour que cela fonctionne.

3.0.2 Intégration des données

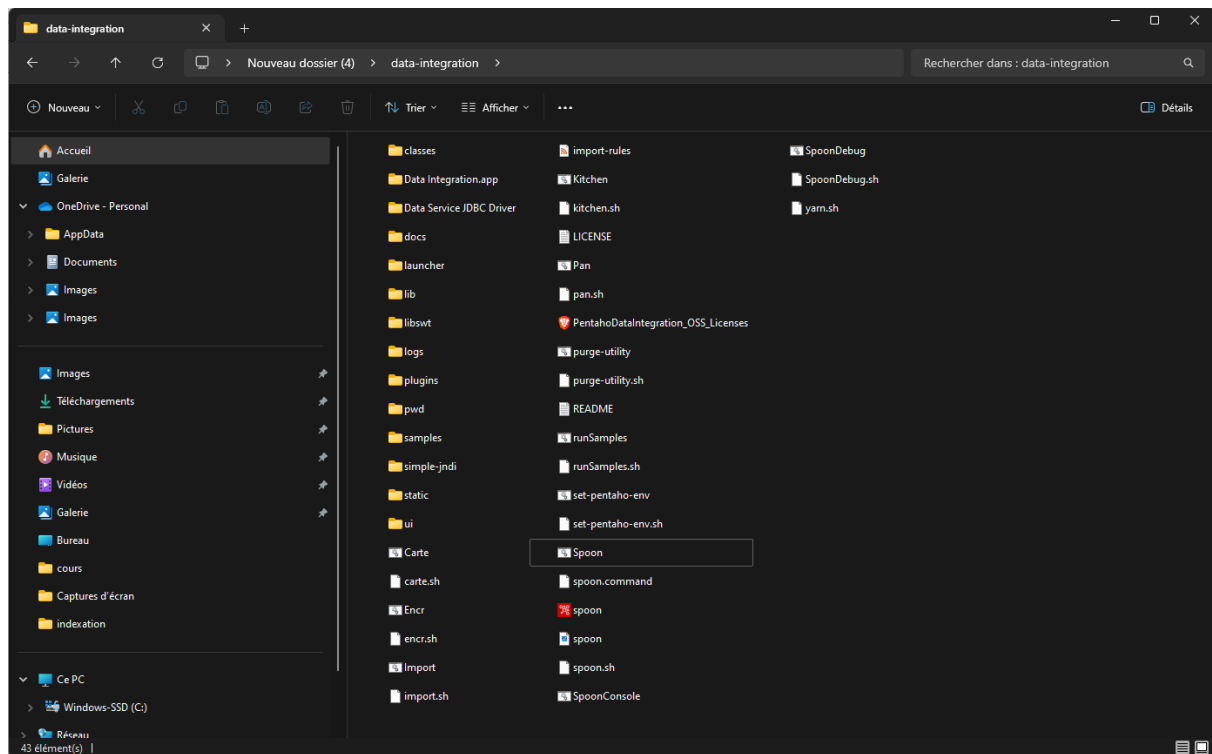
pour que cela fonctionne on a besoin de :

- Pentaho Data Integration (pdi): Télécharger la dernière version de Pentaho Data Integration – Community Edition (9.4) depuis le lien suivant : <http://sourceforge.net/projects/pentaho> Décompresser le fichier téléchargé « pdi-ce-9.4.0.0-343.zip ».

- MYSQL jdbc driver: Après avoir téléchargé le connecteur adéquat (Exemple : « mysql-connector-j-8.3.0.jar »,

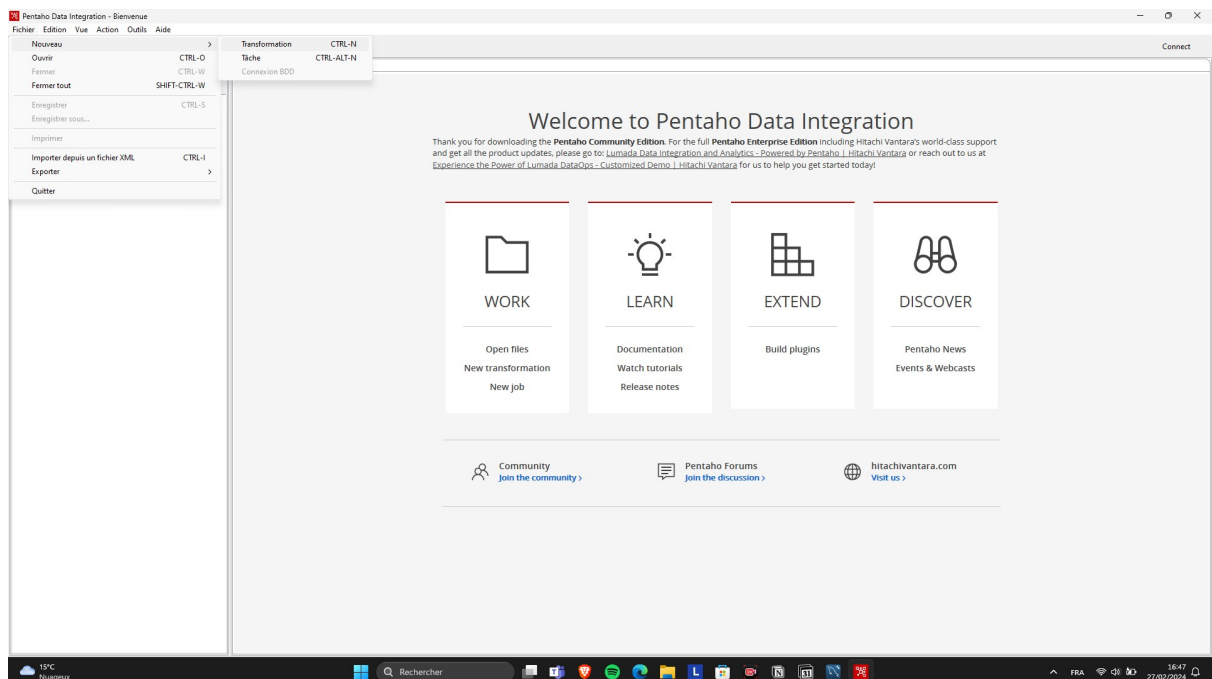
placer le fichier obtenu (*.jar) dans le répertoire C:Files- integration

Après avoir pris en compte toutes les conditions nécessaires , Lancer pentaho data Integration :

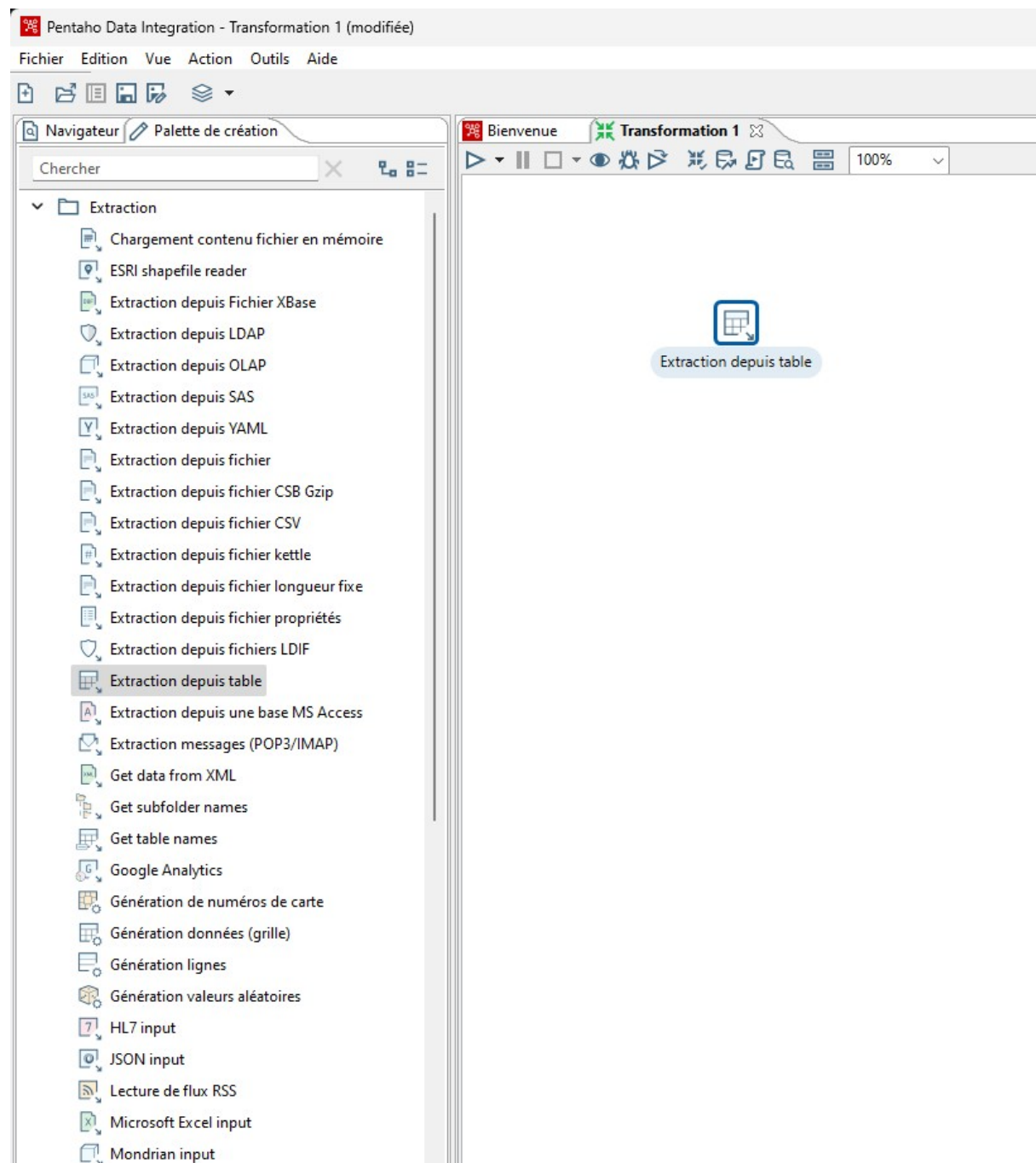


3.0.3 Alimenter la datamart « VentesDM »

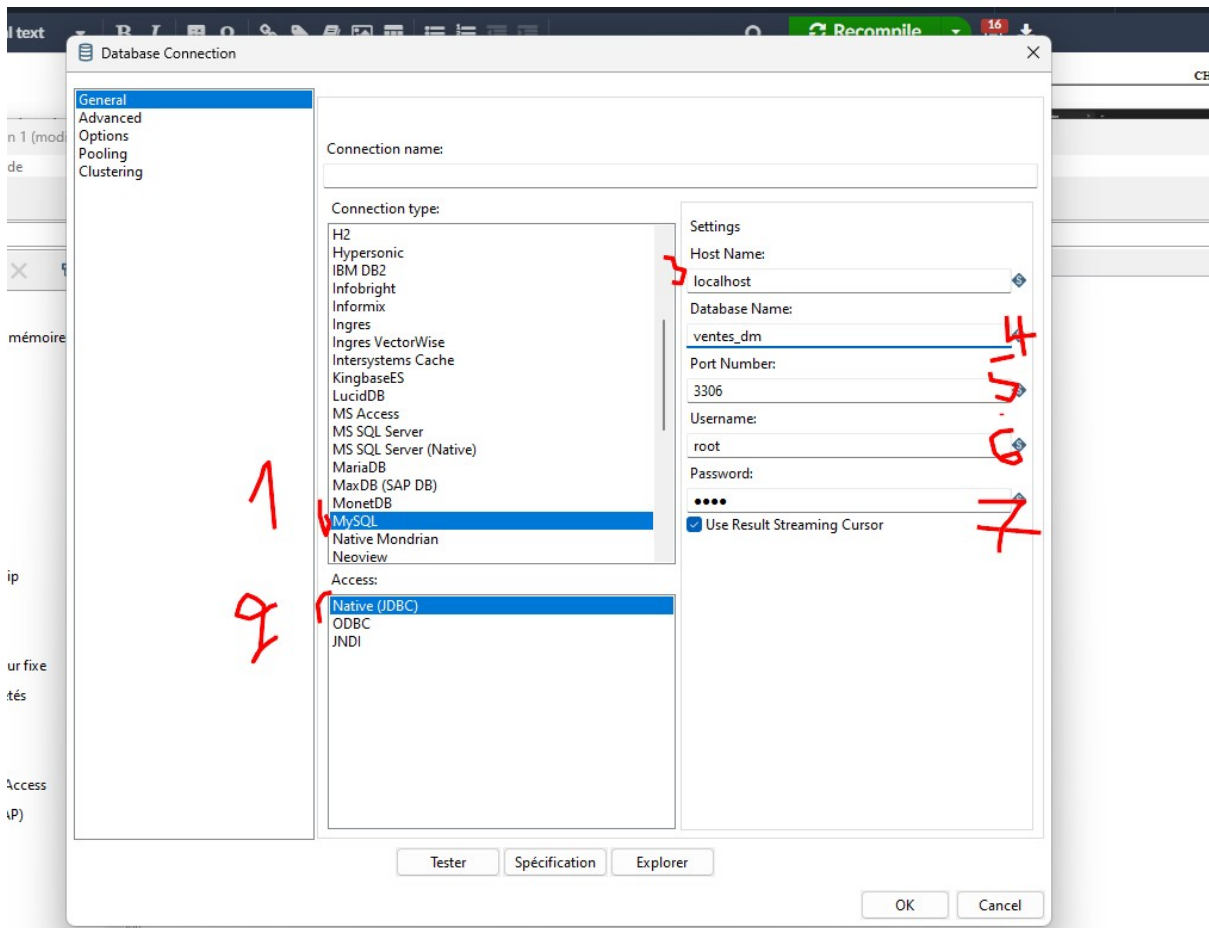
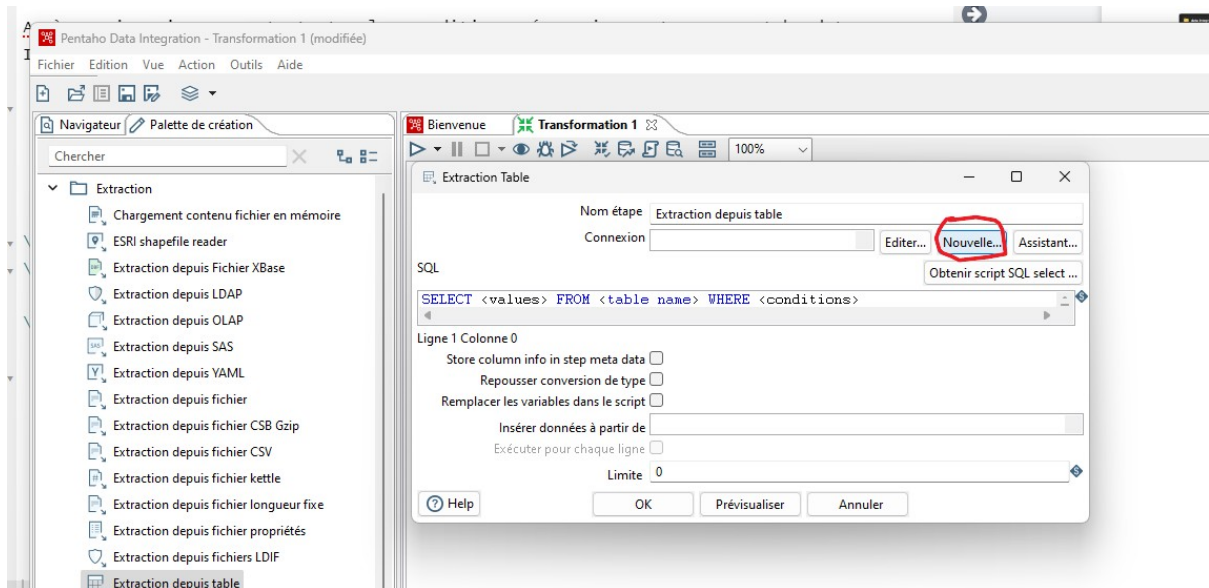
- Créer une nouvelle transformation : Fichier » Nouveau » Transformation.



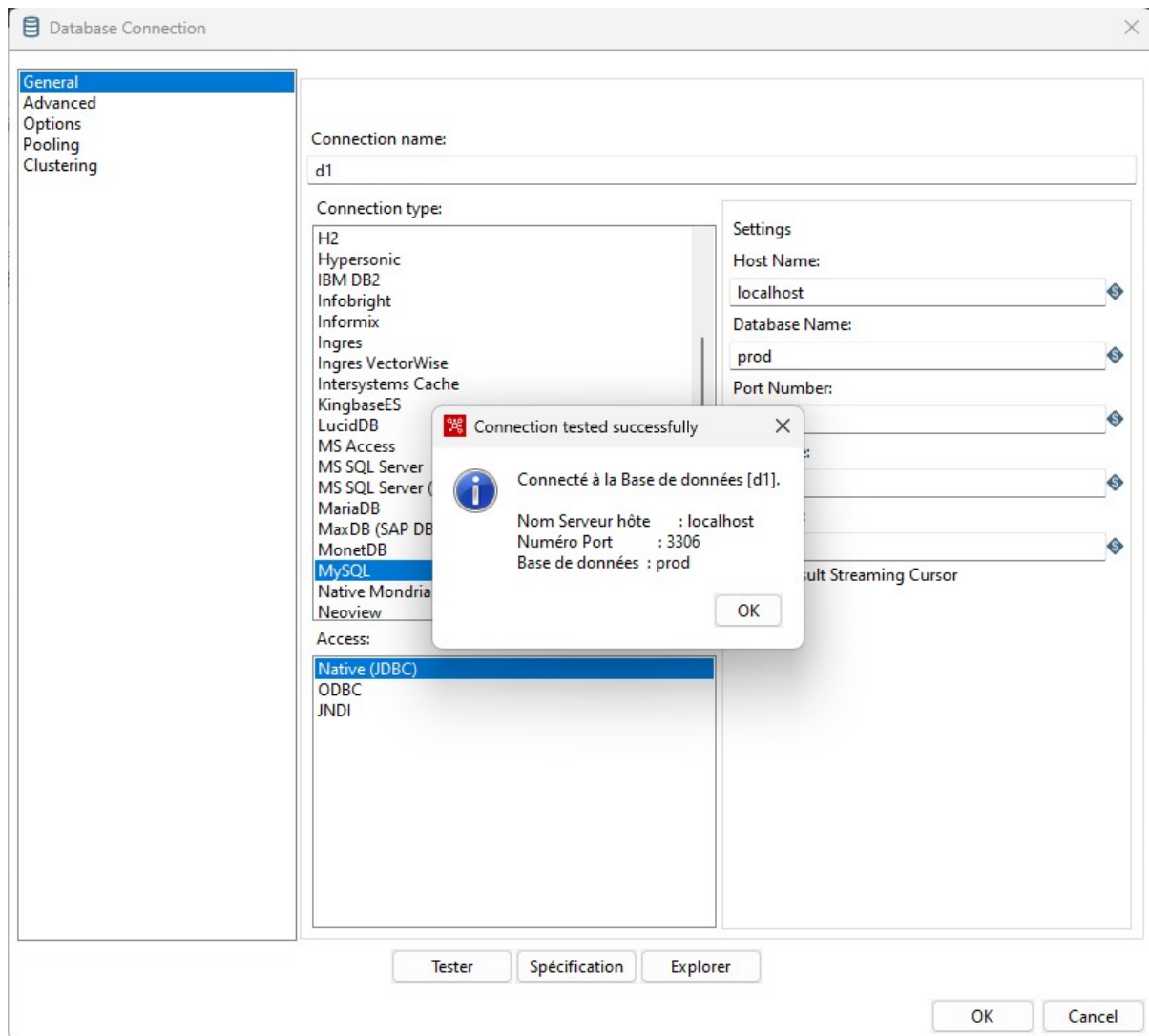
- Ajouter une fonction d'extraction : « Extraction depuis Table ».



- Renommer la fonction d'extraction : « ExtractProduits ».
- Créer une nouvelle connexion à la base de données « prod:connectprod » : remplir les champs convenablement :



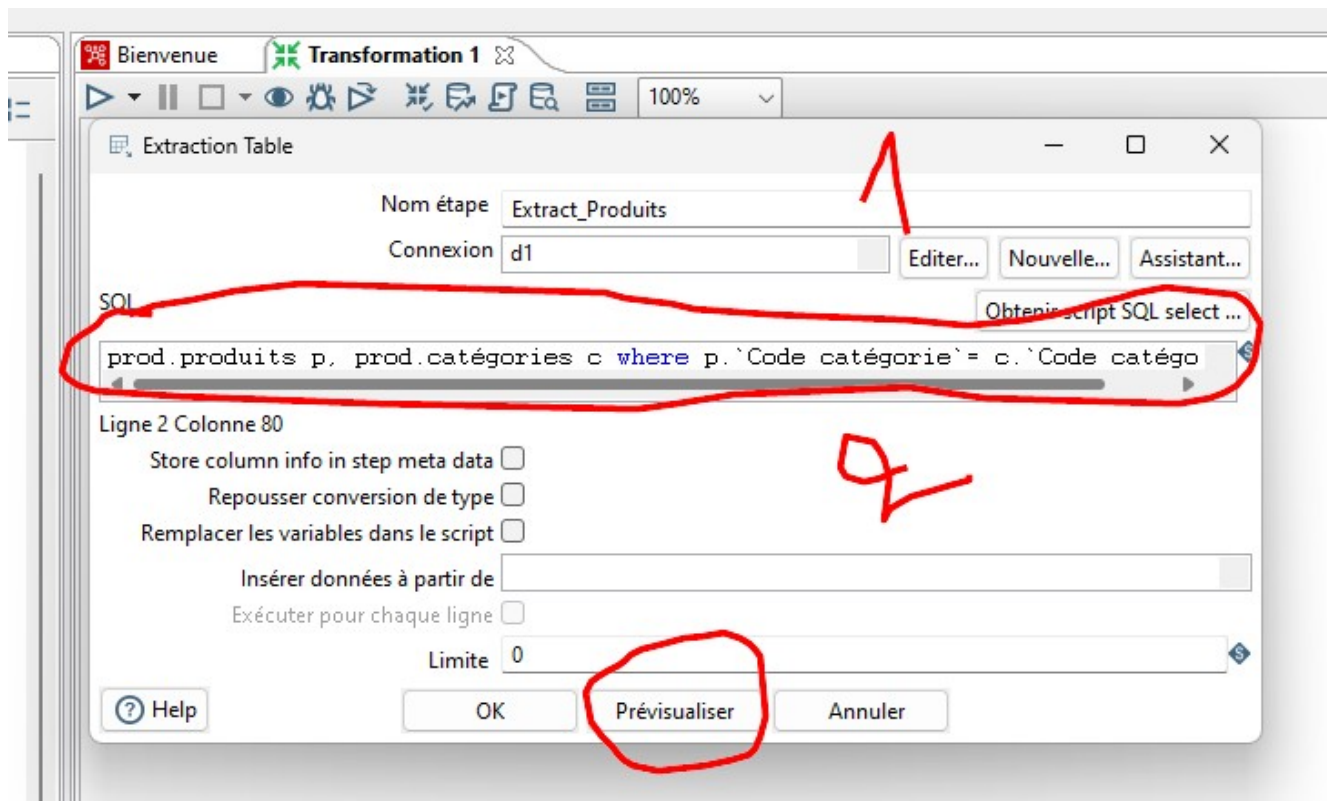
- Tester la connexion à la base de données « prod » :



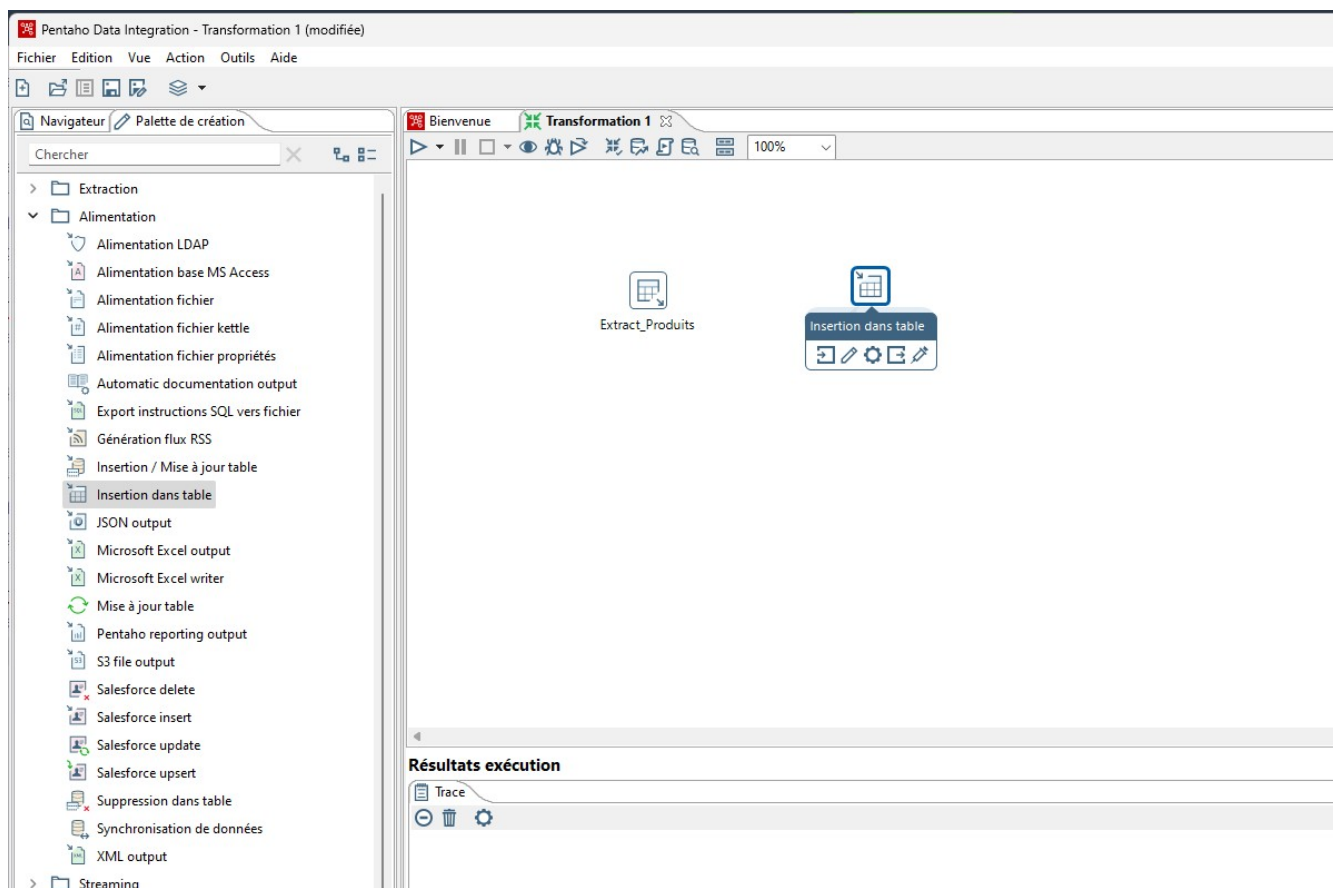
- Sélectionner les données à extraire pour alimenter la dimension « Produits » en utilisant la requête SQL suivante :

```
1 • SELECT p.`Réf produit` , p.`Nom du produit` , c.`Nom de catégorie` FROM
2   prod.produits p, prod.catégories c where p.`Code catégorie`= c.`Code catégorie`;
```

- Cliquer sur « Prévisualiser » pour afficher le dataset obtenu :



- Ajouter une fonction d'alimentation : « Insertion dans table »



- lier entre les deux evenements
- Renommer la fonction d'alimentation: « AlimenterDimProduits »
- Créer une nouvelle connexion au datamart « connectdatamart:ventesDM »

Insertion dans table

Nom étape Alimenter_DimProduits

Connexion d1 Editer... Nouvelle... Assistant...

Schéma cible Parcourir...

Table cible Parcourir...

Valider transaction toutes les 1000

Tronquer la table ☐

Ignorer les erreur d'insertion ☐

Sélectionner champs ☐

Général Champs table

Tables avec données partitionnées ☐

Champ partitionnement

Partitionnement des données par ☒

Partitionnement données par jour ☐

Activer insertion groupée ☒

Nom table défini dans un champ ☐

Champ contenant le nom de la table

Insérer champ nom table dans table ☒

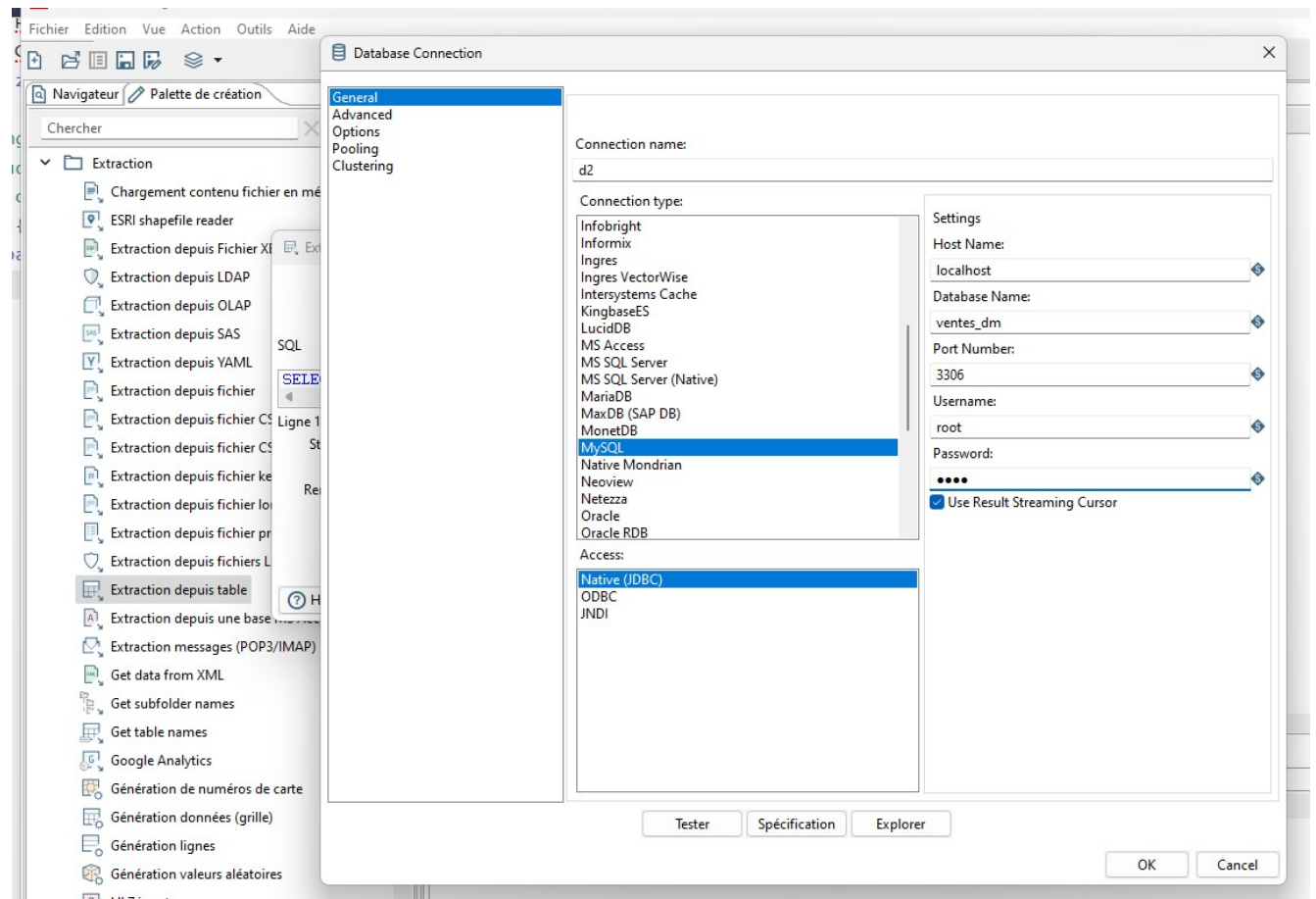
Retourner une clé auto-générée ☐

Nom de la clé auto-générée

Help OK Annuler SQL

3.0.4 alimenter la table client

- Refaire toutes les étapes de la phase 1 en utilisant les paramètres suivants



Requête SQL pour Extraire les données :

```
1      SELECT c.'Code client', c.'Société', c.'Pays' FROM prod.clients c;
```

Insertion dans table

Nom étape : Alimenter_Dimclients

Connexion : d2 [Modifier...] [Nouvelle...] [Assistant...]

Schéma cible : ventes_dm [Parcourir...]

Table cible : dim_produits [Parcourir...]

Valider transaction toutes les : 1000 [S]

Tronquer la table : ☐

Ignorer les erreurs d'insertion : ☐

Sélectionner champs : ☒ [S]

Général | Champs table

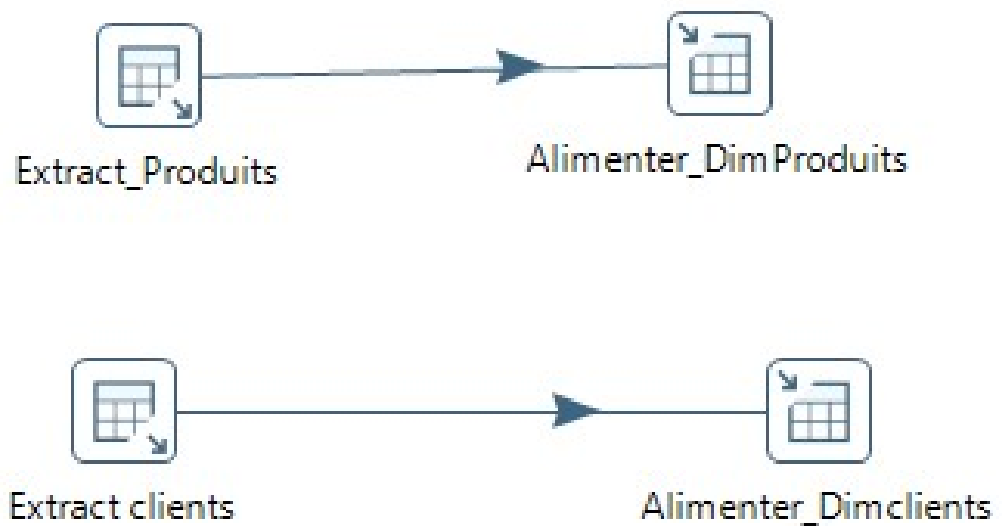
Champs:

#	Champ table	Champ flux	
1	idCli	code client	
2	NomCli	societe	
3	PaysCl	Pays	

[Capturer champs] [Editer schéma]

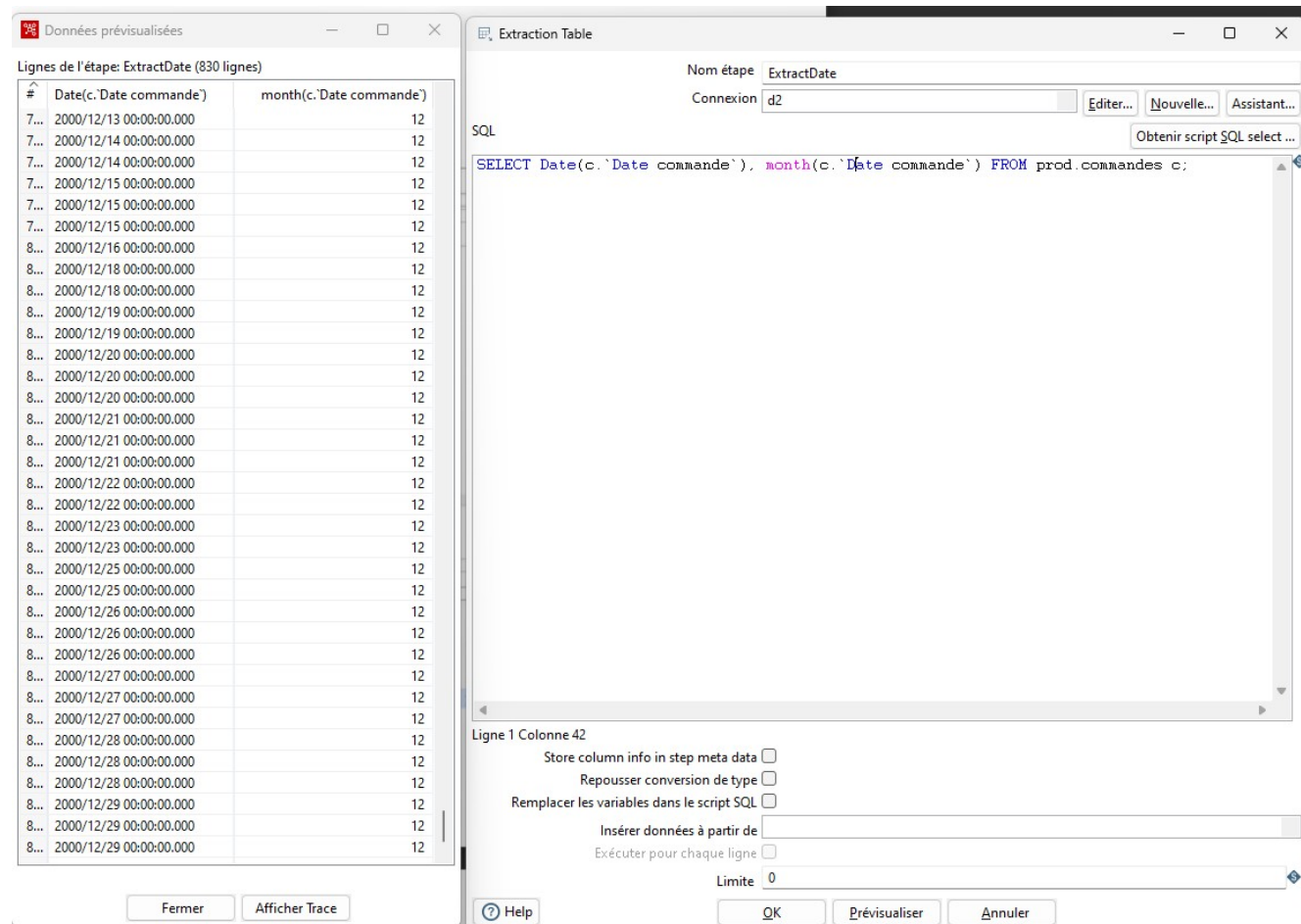
? Help OK Annuler SQL

Resultas :



3.0.5 Alimenter la table dimension Temps

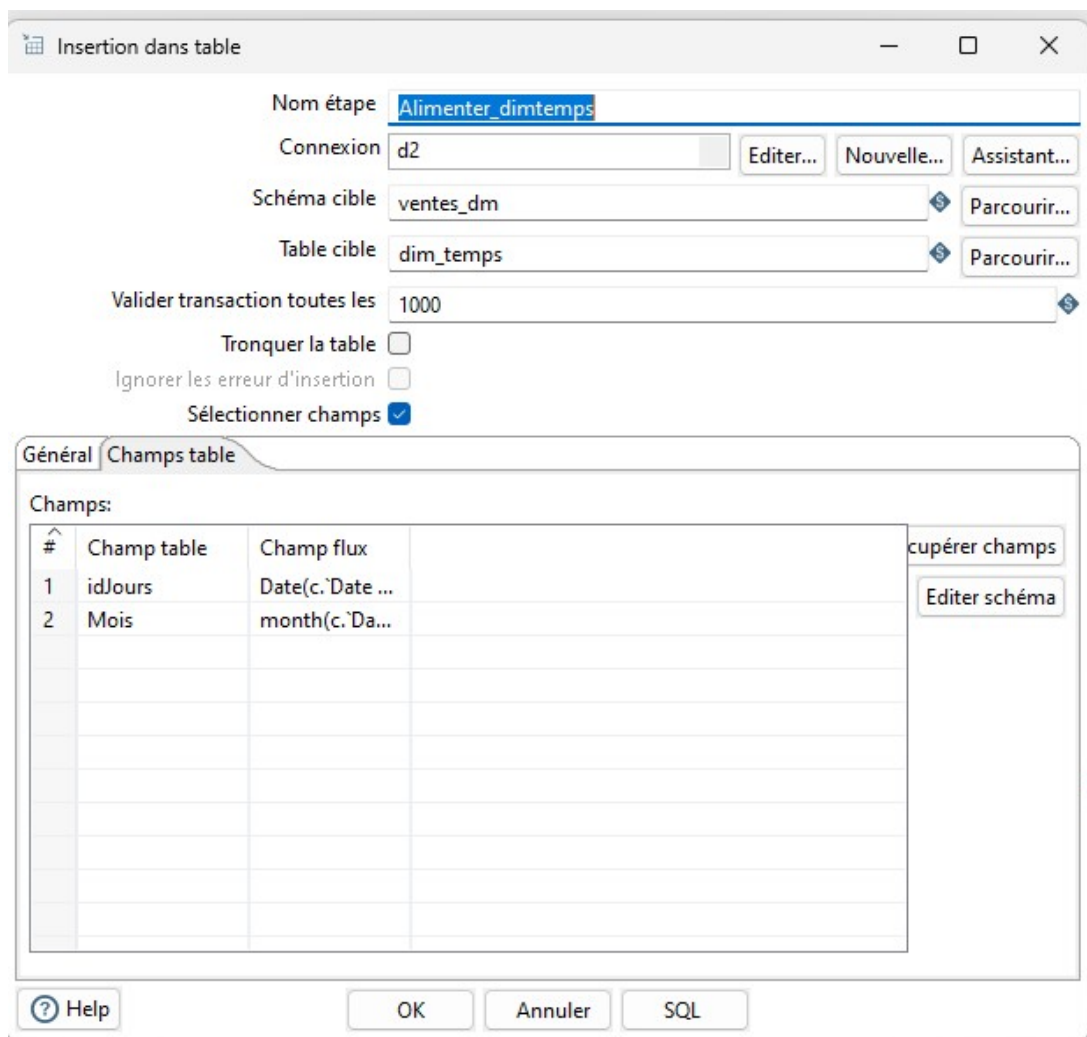
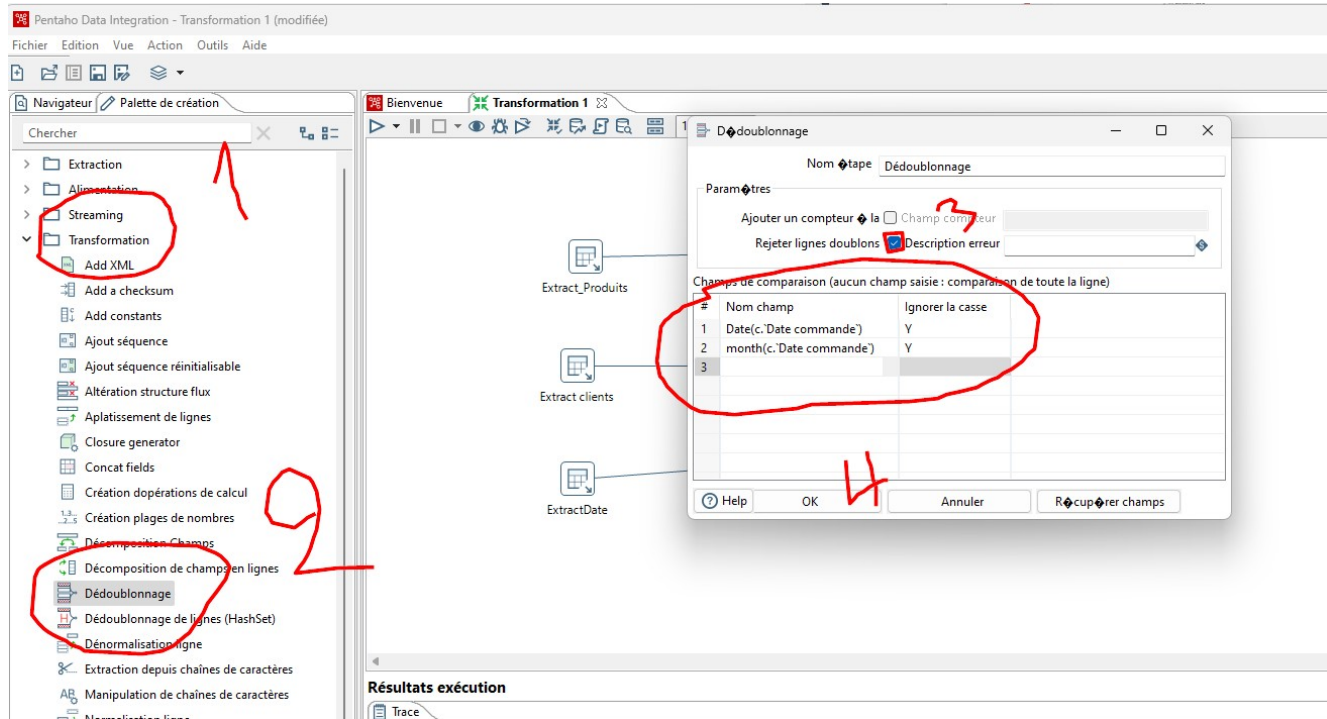
- Refaire toutes les étapes de la phase IV.1 en utilisant les paramètres suivants :
-



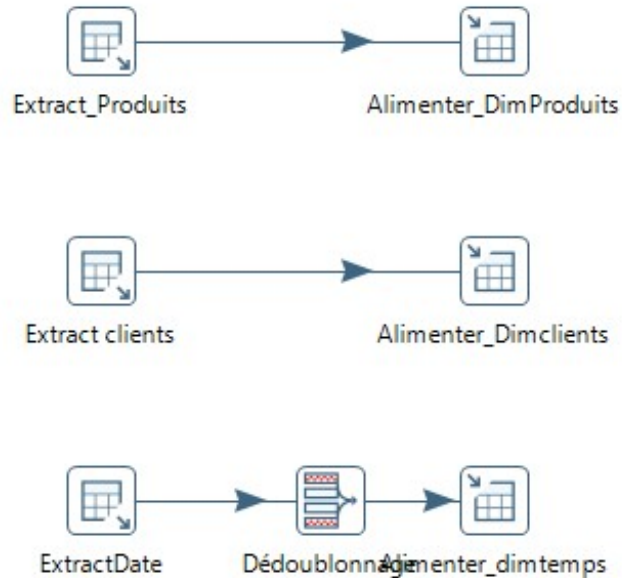
Requête SQL pour Extraire les données :

```
1 SELECT Date(c.'Date commande'), month(c.'Date commande') FROM
2 prod.commandes c;
```

- Après l'Extraction, effectuer une opération de Transformation (dédoublonage) pour supprimer les doublons :



- **resultas :**



3.0.6 Alimenter la table fact Ventes

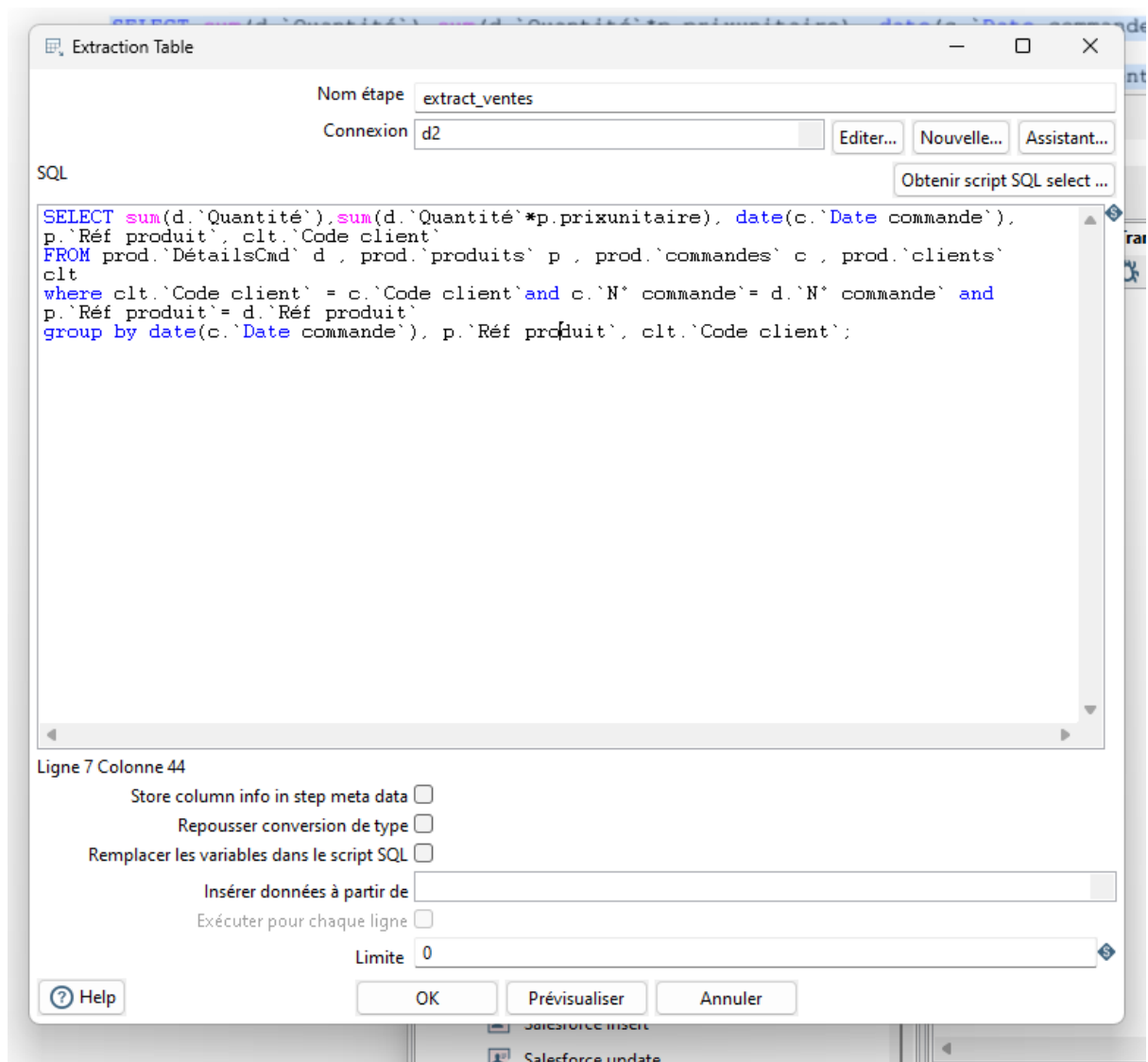
- Refaire toutes les étapes de la phase 1

Requête SQL pour extraire les données et calculer les mesures :

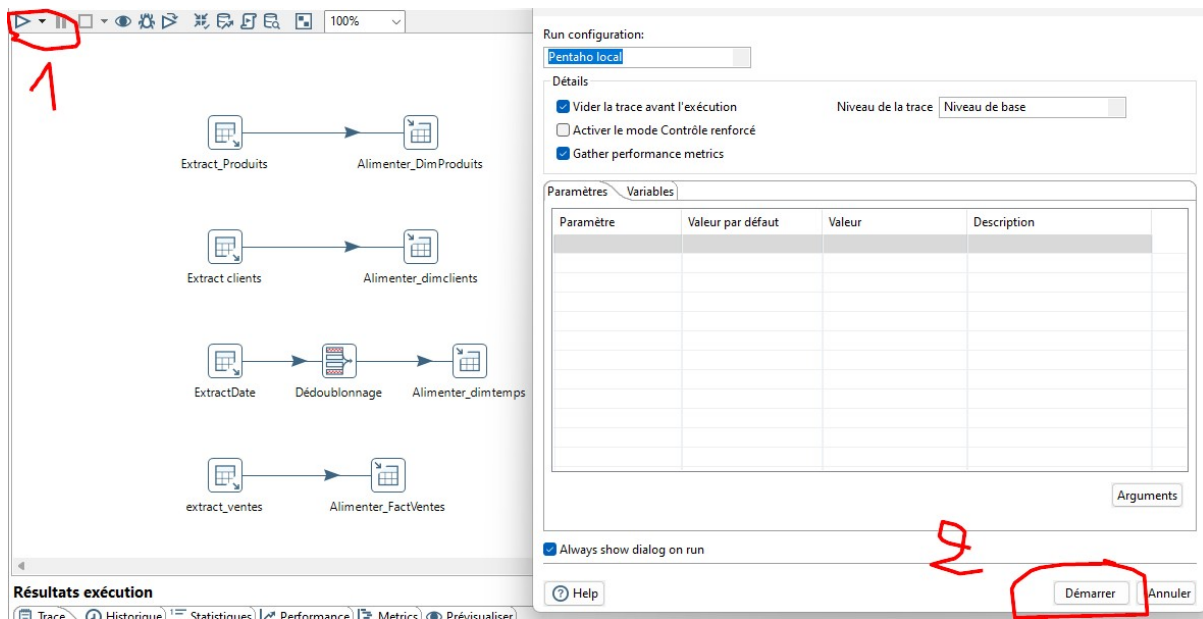
```

Administration - Server Status  SQL File 3
Limit to 50000 rows
1 • SELECT sum(d.`Quantité`),sum(d.`Quantité`*p.prixunitaire), date(c.`Date commande`),
2   p.`Réf produit`, clt.`Code client`
3 FROM prod.`DétailsCmd` d , prod.`produits` p , prod.`commandes` c , prod.`clients`
4   clt
5 where clt.`Code client` = c.`Code client`and c.`N° commande`= d.`N° commande` and
6   p.`Réf produit`= d.`Réf produit`
7 group by date(c.`Date commande`), p.`Réf produit`, clt.`Code client`;

```

- Exécuter la Transformation (tout le workflow) pour alimenter le datamart



Utiliser MySQL Workbench pour vérifier que le datamart a été alimenté :

Table « ventesDM .dimclients »

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a search filter 'Filter objects'. The 'shemadatabase' is expanded, showing various database objects. The 'Administration' tab is selected, showing connection details for 'Local instance MySQL80'.

The main window shows a query editor with the following SQL query:

```
1 Select * FROM ventes_dm.dim_clients;
```

The query is executed, and the results are displayed in the 'Result Grid' tab. The results show a list of clients with columns 'idCl', 'NomCl', and 'PaysCl'.

idCl	NomCl	PaysCl
ALFKI	Alfreds Futterkiste	Allemagne
ANATR	Ana Trujillo Emparedados y helados	Mexique
ANTON	Antonio Moreno Taquería	Mexique
AROUT	Around the Horn	Royaume-Uni
BERGS	Berglunds snabbköp	Suède
BLAUS	Blauer See Delikatessen	Allemagne
BLONP	Blondel père et fils	France
BOLID	Bólido Comidas preparadas	Espagne
BONAP	Bon app'	France
BOTTM	Bottom-Dollar Markets	Canada
BSBEV	B's Beverages	Royaume-Uni
CACTU	Cactus Comidas para llevar	Argentine
CENTC	Centro comercial Moctezuma	Mexique
CHOPS	Chop-suey Chinese	Suisse
COMMI	Comércio Mineiro	Brésil
CONSH	Consolidated Holdings	Royaume-Uni
DRACD	Drachenblut Delikatessen	Allemagne
DUMON	Du monde entier	France
FASTC	Eastern Connection	Royaume-Uni

The 'Output' tab shows the execution log with the following entries:

#	Time	Action	Message
34	23:40:52	SELECT * FROM ventes_dm.dim_produits LIMIT 0, 50000	77 row(s) returned
35	23:40:52	idCl PaysCl NomCl	Error Code: 1064. You have an error in your SQL syntax; check the man...
36	23:41:57	Select * FROM ventes_dm.dim_clients LIMIT 0, 50000	91 row(s) returned

The 'Query Completed' status is shown at the bottom.

Table « ventesDM.dim.produits »

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `Select * FROM ventes_dm.dim_produits;`. The Result Grid displays 19 rows of data with columns `idProd`, `NomProd`, and `CatProd`. The Action Output pane at the bottom shows the execution of the query, indicating that 77 rows were returned.

#	Time	Action	Message
35	23:40:52	idClPaysCINomCl	Error Code: 1064. You have an error in your SQL syntax; check the man...
36	23:41:57	Select * FROM ventes_dm.dim_clients LIMIT 0, 50000	91 row(s) returned
37	23:50:20	Select * FROM ventes_dm.dim_produits LIMIT 0, 50000	77 row(s) returned

Table « ventesDM.dim.temps »

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'ventes_dm' database expanded, showing tables 'dim_clients', 'dim_produits', and 'dim_temps'. The 'dim_produits' table is selected, showing its columns: 'idProd' (int PK), 'NomProd' (varchar(45)), and 'CatProd' (varchar(45)).

The main window shows a SQL query in 'Query 1':

```
1 • select * from ventes_dm.dim_temps
```

The 'Result Grid' displays the query results for 'dim_temps'. The table has two columns: 'idJours' and 'Mois'. The results show dates from 2000-01-01 to 2000-01-23, all with a value of 1 in the 'Mois' column.

idJours	Mois
2000-01-01	1
2000-01-02	1
2000-01-03	1
2000-01-04	1
2000-01-05	1
2000-01-06	1
2000-01-08	1
2000-01-09	1
2000-01-10	1
2000-01-11	1
2000-01-12	1
2000-01-13	1
2000-01-15	1
2000-01-16	1
2000-01-17	1
2000-01-18	1
2000-01-19	1
2000-01-20	1
2000-01-22	1
2000-01-23	1

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message
✓ 34	00:36:07	select * from ventes_dm.fact_ventes LIMIT 0, 50000	70 row(s) returned
✓ 35	00:39:00	select * from ventes_dm.dim_clients LIMIT 0, 50000	90 row(s) returned
✓ 36	00:40:42	select * from ventes_dm.dim_temps LIMIT 0, 50000	320 row(s) returned

Table « ventesDM.fact.ventes »

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'Local instance MySQL80', with the 'fact_ventes' table selected under the 'ventesDM' database. The main window shows the SQL query 'select * from ventesDM.fact_ventes' and the resulting data grid. The data grid contains 30 rows of sales data with columns: Quantite, Montant, idCl, idProd, and idJours. The bottom panel shows the 'Output' tab with the 'Action Output' table, which lists the execution of three queries: a select from dim_clients (90 rows), a select from dim_temps (320 rows), and the current select from fact_ventes (70 rows).

Table: dim_produits

Columns:

- idProd: int PK
- NomProd: varchar(45)
- CatProd: varchar(45)

Quantite	Montant	idCl	idProd	idJours
30	5850	BLONP	17	2000-08-25
20	1500	BLONP	70	2000-08-25
10	500	CENTC	21	2000-08-18
1	130	CENTC	37	2000-08-18
15	337.5	CHOPS	24	2000-08-11
21	2520	CHOPS	55	2000-08-11
21	1050	CHOPS	74	2000-08-11
50	4750	ERNSH	2	2000-08-17
65	6938.75	ERNSH	5	2000-08-17
60	5235	ERNSH	16	2000-08-23
28	630	ERNSH	24	2000-08-23
60	7767	ERNSH	30	2000-08-23
6	960	ERNSH	32	2000-08-17
36	1800	ERNSH	74	2000-08-23
35	3325	FOLKO	2	2000-08-24
25	1206.25	FOLKO	41	2000-08-24
50	4600	FRANK	40	2000-08-29
70	19250	FRANK	59	2000-08-29
15	1350	FRANK	76	2000-08-29
10	6189.5	GROSR	29	2000-08-30

#	Time	Action	Message	Duration / Fetch
35	00:39:00	select * from ventesDM.dim_clients LIMIT 0, 50000	90 row(s) returned	0.000 sec / 0.000 sec
36	00:40:42	select * from ventesDM.dim_temps LIMIT 0, 50000	320 row(s) returned	0.000 sec / 0.000 sec
37	00:41:46	select * from ventesDM.fact_ventes LIMIT 0, 50000	70 row(s) returned	0.000 sec / 0.000 sec

Chapter 4

Conclusion Générale

En conclusion, ce document offre un aperçu complet du processus de création, de conception et d'intégration des données pour faciliter l'analyse du sujet des ventes. Il met en évidence l'importance de chaque étape dans la création d'un environnement de données robuste et prêt à être exploité pour des analyses approfondies et des prises de décision éclairées d'une entrepôt de données.