

Changez de vision et **prenez les bonnes décisions**



MIAGE - TP Qlikview - Script
Decideom - Jérôme Paul - Novembre 2016

Organisation du cours Qlikview

Partie cours (2h)

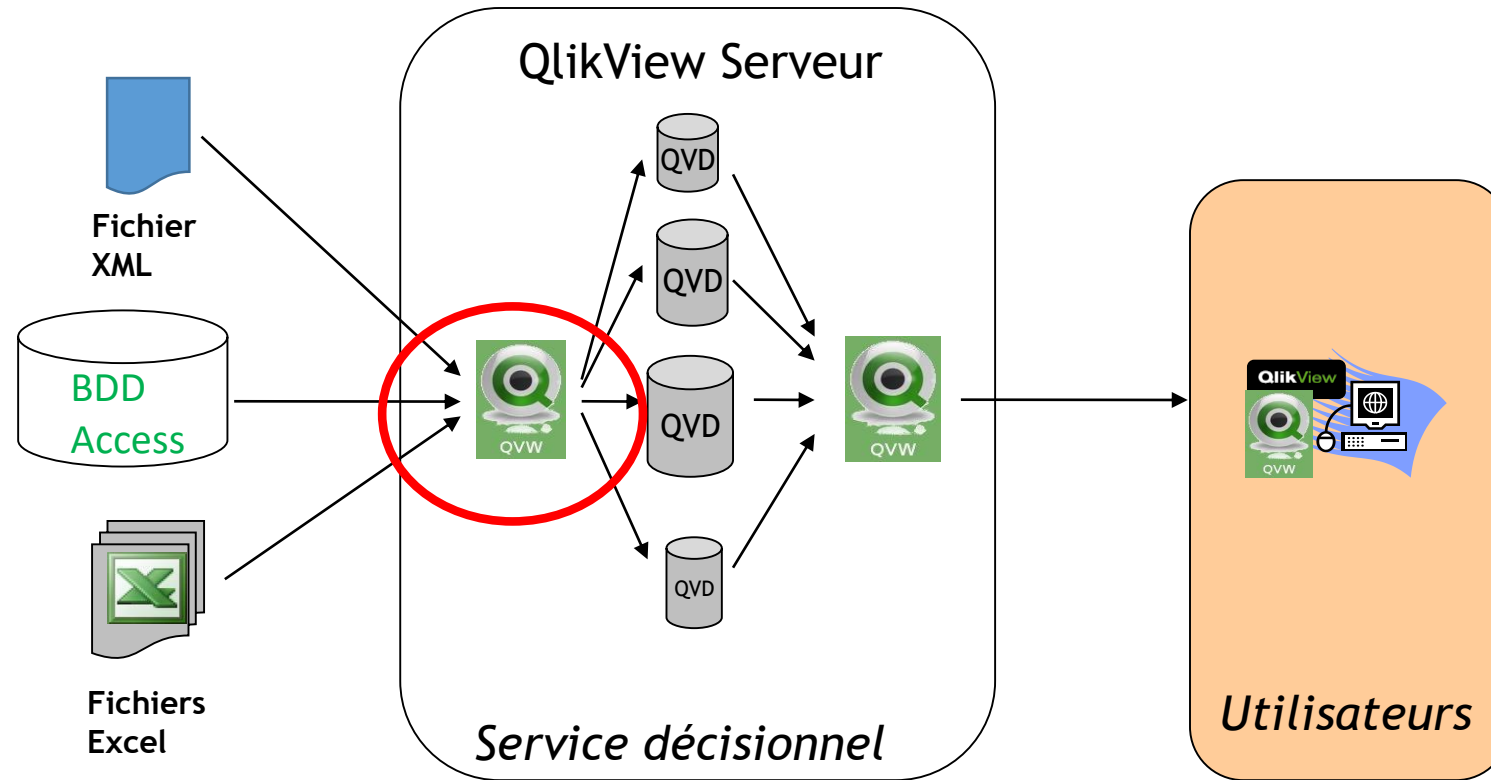
- *Positionnement de Qlik dans le monde de la BI*
- *Architecture décisionnelle avec Qlikview*
- *Partie Script (ETL / Modélisation)*
- *Partie Restitution (Les bonnes pratiques)*
- *Partie Serveur (Administration)*

Partie TP (4h)

- **Création d'une application d'extraction (2h)**
Alimentation à partir d'une base de données
Enrichissement à partir de fichiers
Champs calculés et règles de gestion
- *Création d'une application de restitution (2h)*



Objectif du TP – Créer une application d'extraction



un QVD = une table

DWH $\Leftrightarrow \Sigma$ QVDs

une application QVW

= un datamart



Plan de la partie Script

- Connexion à une base de données Access
- Récupération des tables
- Analyse des chargements
- Ajout de fichiers (XLS)
- Création de données calculées sous Qlikview
- Scripts avancés (Sum, Join, Concatenate, Exists)
- Fichiers de données QVD



Connexion à la base access

CTRL + E permet d'accéder au script de l'application

Positionner le curseur en dessous des SET

Choisir une connexion **OLE DB**

Cocher « **Forcer la version 32bits** »

Prendre le driver : **Microsoft Jet 4.0 OLE DB Provider**

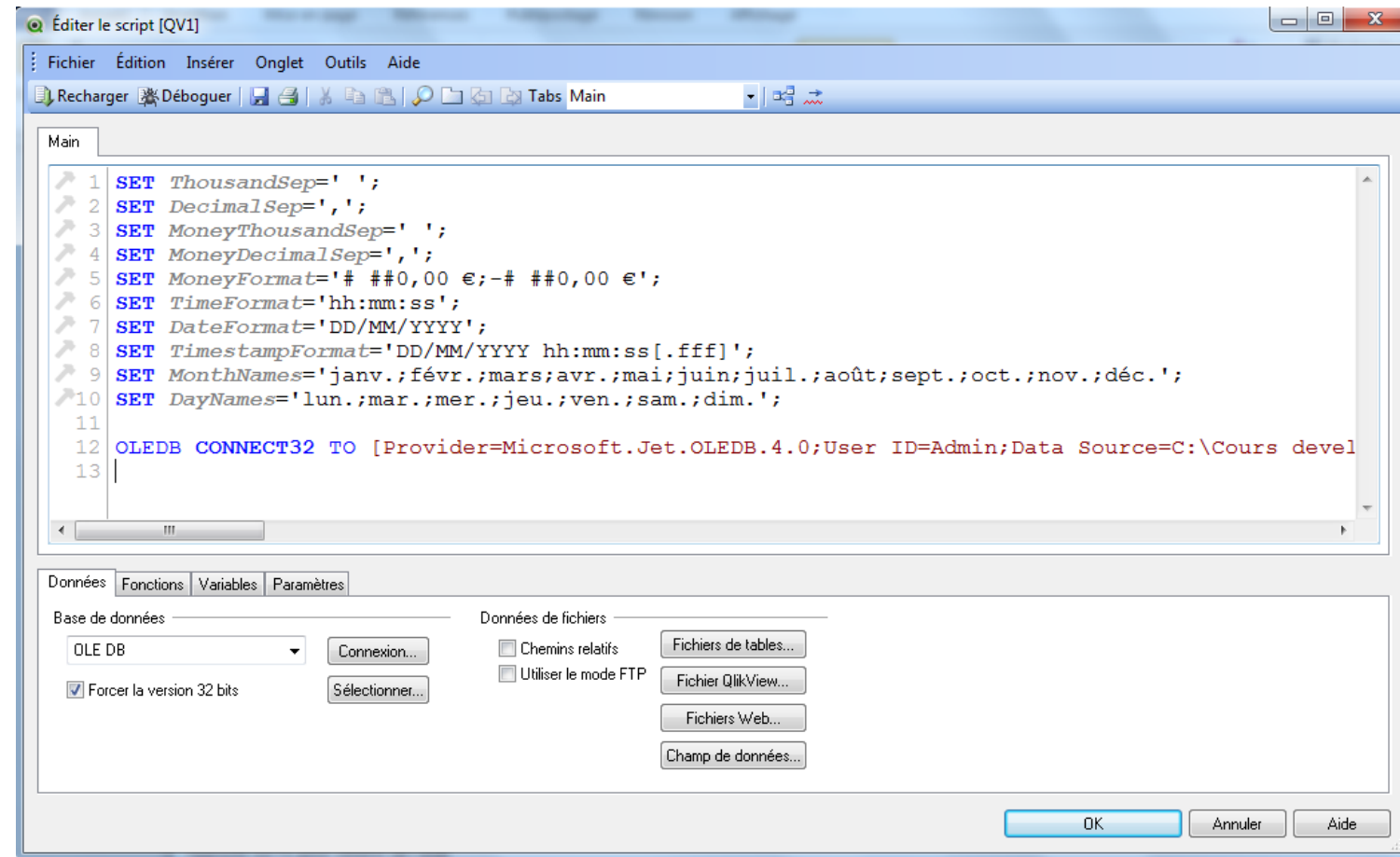
Chercher la base **QWT.mdb** dans le dossier **Datasources**

Prendre l'utilisateur **Admin**

Tester la connexion

Valider par « **OK** »

Le script de la connexion est généré automatiquement



Connexion à la base access – contenu de la base

Tables de base de données

Categories

Customers

Divisions

Order Details

Orders

Products

Shipments

Shippers

Categories contient des catégories des produits

Customers contient les clients

Divisions contient les divisions des clients

Order Details contient le détail des commandes

Orders contient les entêtes des commandes

Products contient les produits

Shipments contient le détail des expéditions

Shippers contient les expéditeurs

Information sur les jointures et la modélisation de la base de données Access

Les champs finissant par **ID** et par **No** sont des champs clefs, tous les autres ne sont pas des clefs.

La clef primaire des tables **Order Details** et **Shipments** est composée des champs **OrderID** et **LineNo** ; les autres champs ID de ces tables sont des doublons.

La clef primaire de la table **Orders** est le champ **OrderID**, les autres champs ID de cette table sont des clefs étrangères.



Récupération des tables – Utilisation de l'assistant

Récupérer les tables de dimensions :

Categories, prendre les 2 champs :
CategoryID
CategoryName

Customers : tous les champs

Divisions : tous les champs

Products : tous les champs

Shippers : tous les champs

Faire **Ajouter** entre chaque table puis cliquer sur **Ok**.

Remarque : Ajouter *Nom_table*: juste avant Load permet de nommer la table

Pour créer et charger les tables, il faut exécuter le script **CTRL + R**

La première fois il faut sauvegarder l'application.

Créer une instruction Select

Source de données: Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin; Connexion...
Base de données: Pilote...
Propriétaire: Support...

Tables de base de données: ☒ Tables, ☐ Vues, ☐ Synonymes, ☐ Tables système, ☐ Alias

Champs: Tri alphabétique

Champs sélectionnés: *
CategoryID
CategoryName
Description

☒ Afficher l'icône des champs clés

Script: Table, Colonnes, Aperçu, Blobs

```
LOAD CategoryID,  
CategoryName,  
Description;  
SQL SELECT *  
FROM Categories;
```

Options de génération du script:
☒ Colonne
☐ Ligne
☐ Structuré
☒ Avant Load
Ajouter

OK Annuler Aide

Choix de la table à intégrer

Choix des champs à sélectionner
Ne jamais utiliser *

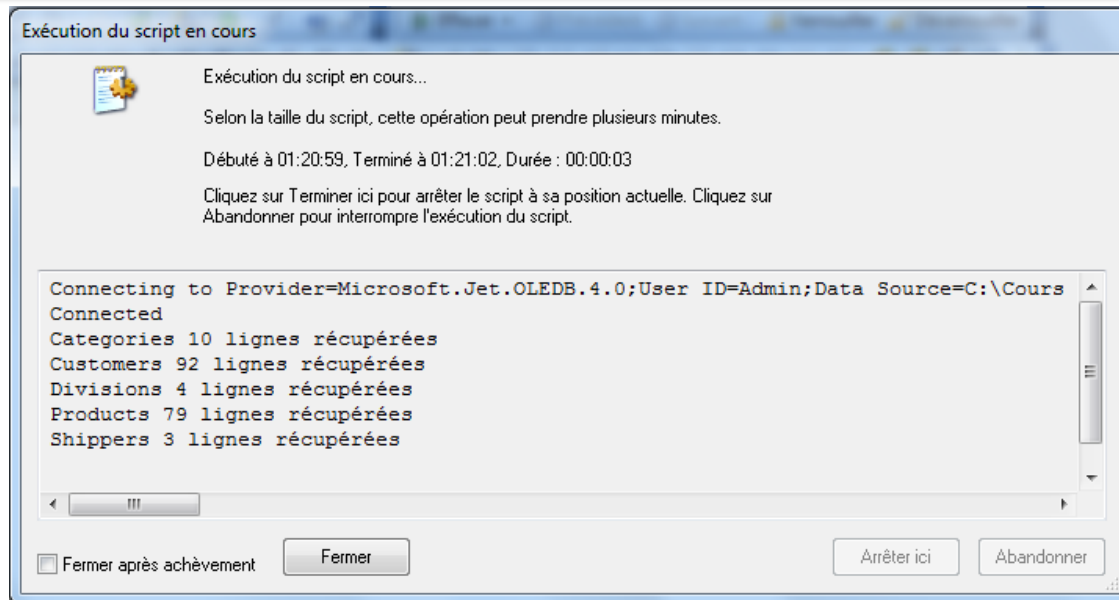
Options de génération du script
Utiliser Colonne et cocher « Avant Load »

Ajoute la table sélectionnée
Permet de passer ensuite à une autre table

Valide la sélection



Récupération des tables – Analyse dans le visionneur de table



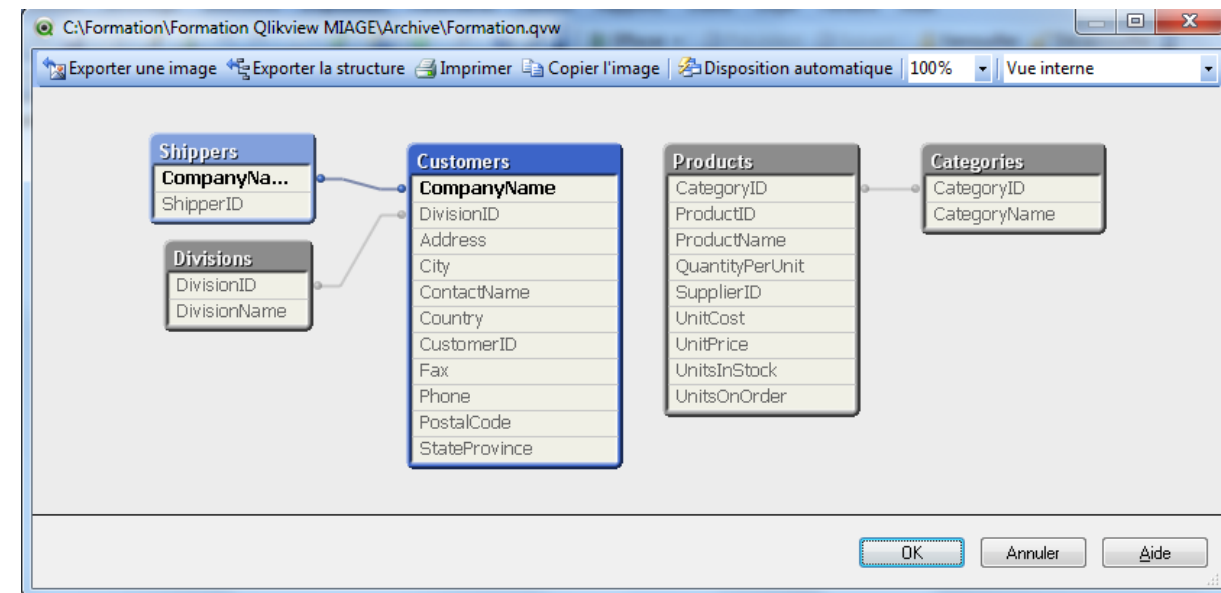
Le visionneur de table permet d'observer et d'analyser les données chargées dans l'application Qlikview

CTRL + T permet d'y accéder

La jointure sur **CompanyName** n'est pas correcte, il faut retourner dans le script pour casser la jointure en renommant l'un des champs à l'aide de la commande **AS**

La fenêtre de progression permet de suivre le chargement

La fenêtre se ferme toute seule, mais il est possible de modifier l'option dans : Paramètres\Préférences utilisateur...\Conserver la fenêtre de progression ouverte après le rechargement ou de décocher « **Fermer après achèvement** » pour la garder visible.



Récupération des tables – Ajout des tables de faits

1) Récupérer les tables de faits

Order Details : tous les champs
Orders : tous les champs

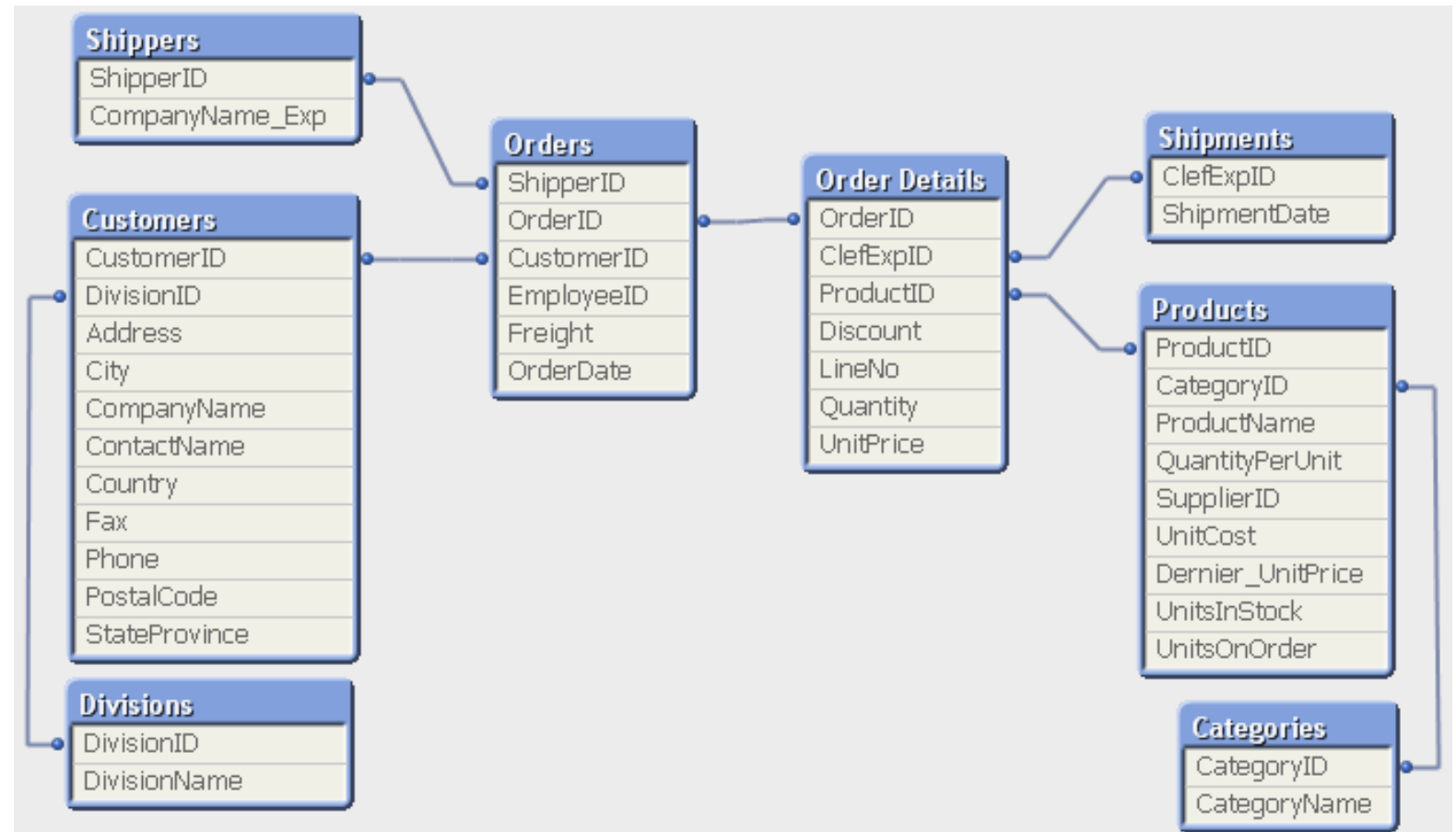
2) Analyser le chargement

3) Résoudre les problèmes

4) Refaire l'exercice avec la table de faits

Shipments : tous les champs

Résultat attendu



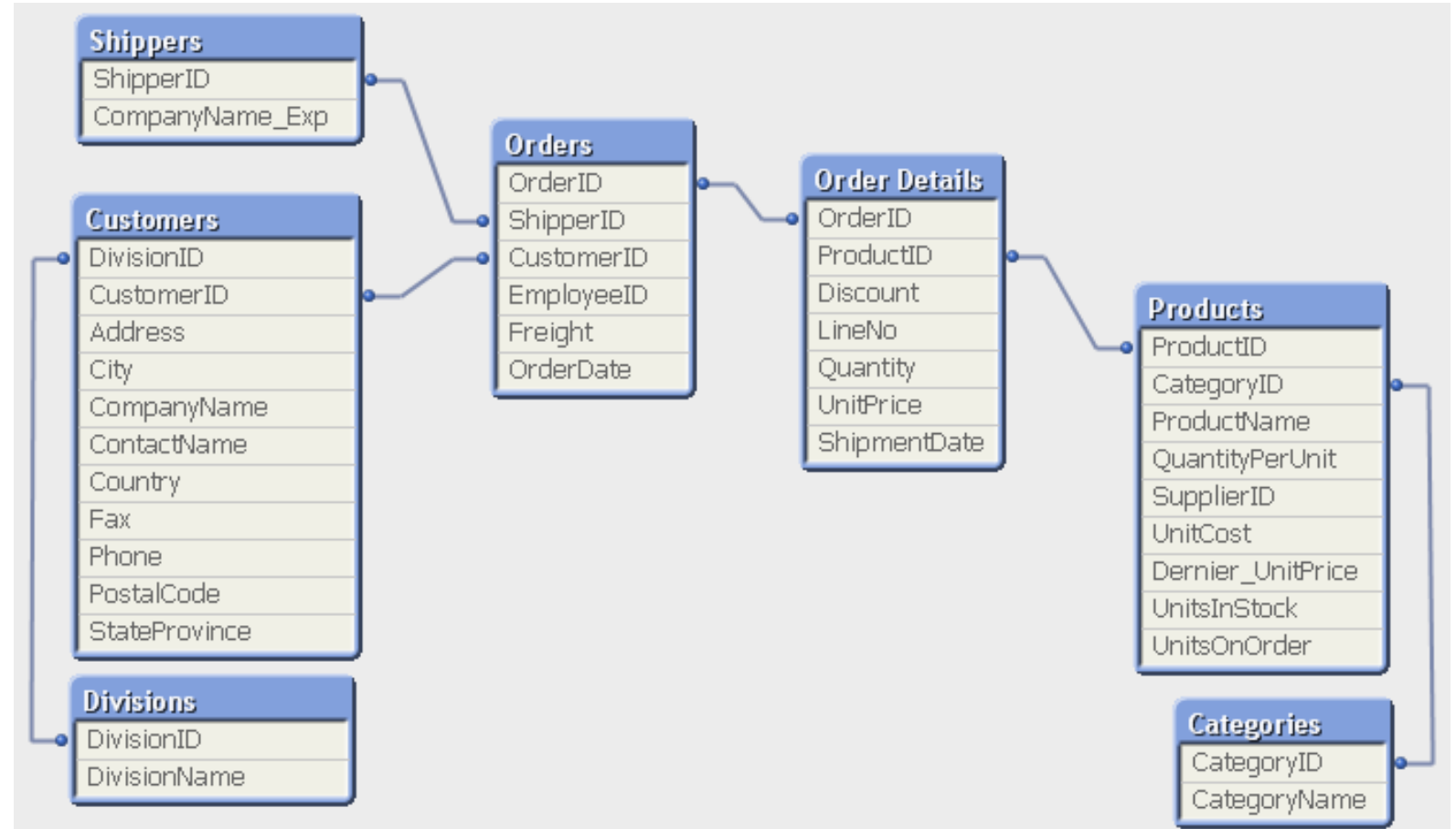
Récupération des tables – Optimisation

Règles d'optimisations

- 1) Si 2 tables de faits ou 2 tables de dimensions ont sensiblement la même volumétrie et la même clef, alors elles doivent être fusionnées avec la commande **JOIN**
- 2) Si un champ est inutile alors il peut être supprimée avec la commande : **Drop Field Nom_Champ;**

Optimisez le modèle

Résultat attendu



Ajout de fichiers – utilisation de l'assistant

Chargement des données des employés

Utiliser l'assistant d'import des fichiers pour intégrer le fichier des employés.

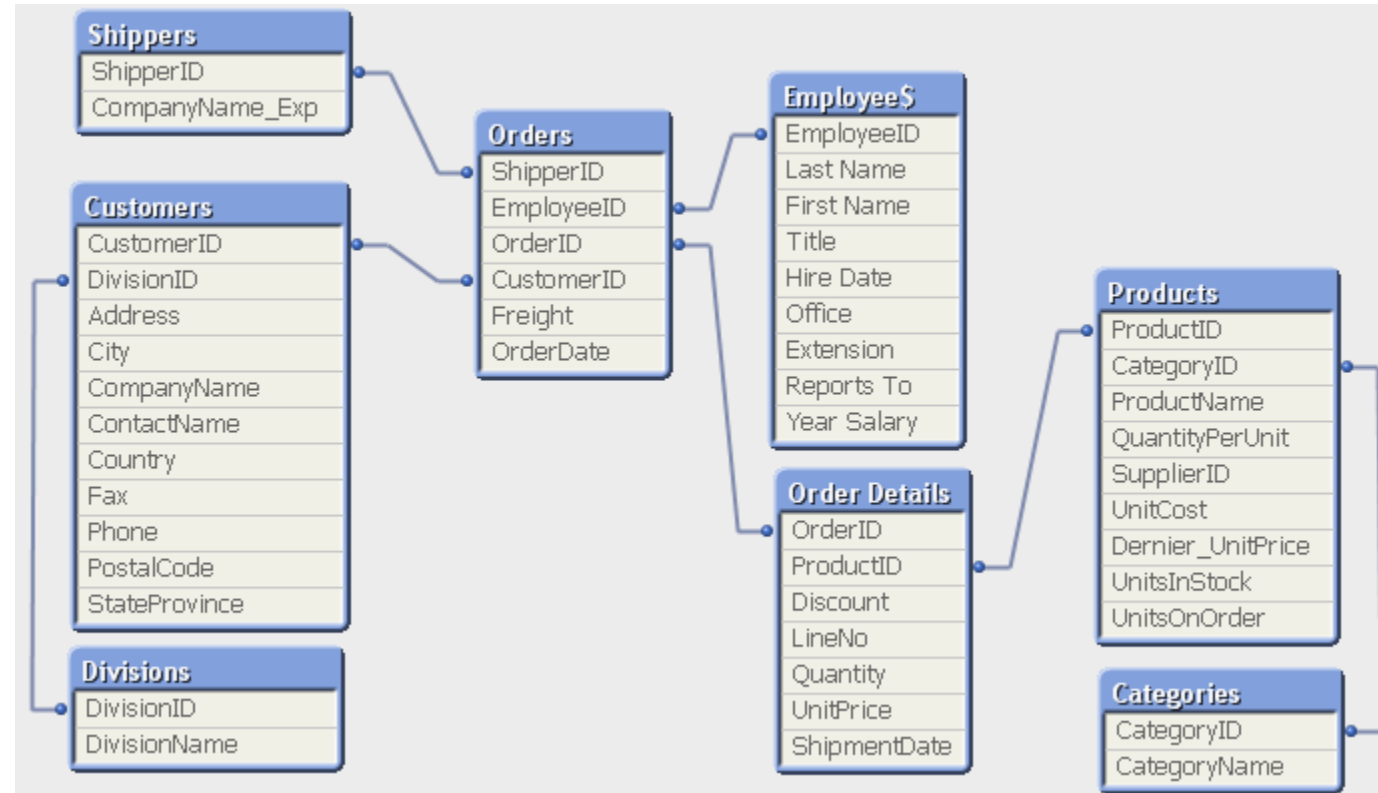
Le fichier **EmpOff.xls** se trouve dans le dossier **Datasources**

Chargez l'onglet **Employee**

Cet onglet contient les informations sur les employés qui doivent être liées au champ EmployeeID de la table Orders

Remarque : utiliser l'option étiquette incorporée dans l'assistant pour utiliser la première ligne pour nommer les champs.

Résultat attendu



Création de données calculées - Formule et If()

1) Créer un champ calculé **Montant_Net** dans la table **Order Details**

Règle :

$$\text{Montant_Net} = \text{Quantity} * \text{UnitPrice} * (1 - \text{Discount})$$

2) Créer un champ calculé **CategoryType** dans la table **Categories**

Règle :

Si CategoryID=5 ou CategoryID=6

Alors CategoryType = 'Footwear'

Sinon CategoryType = 'Others'



Scripts avancés – Sum, Resident et Join

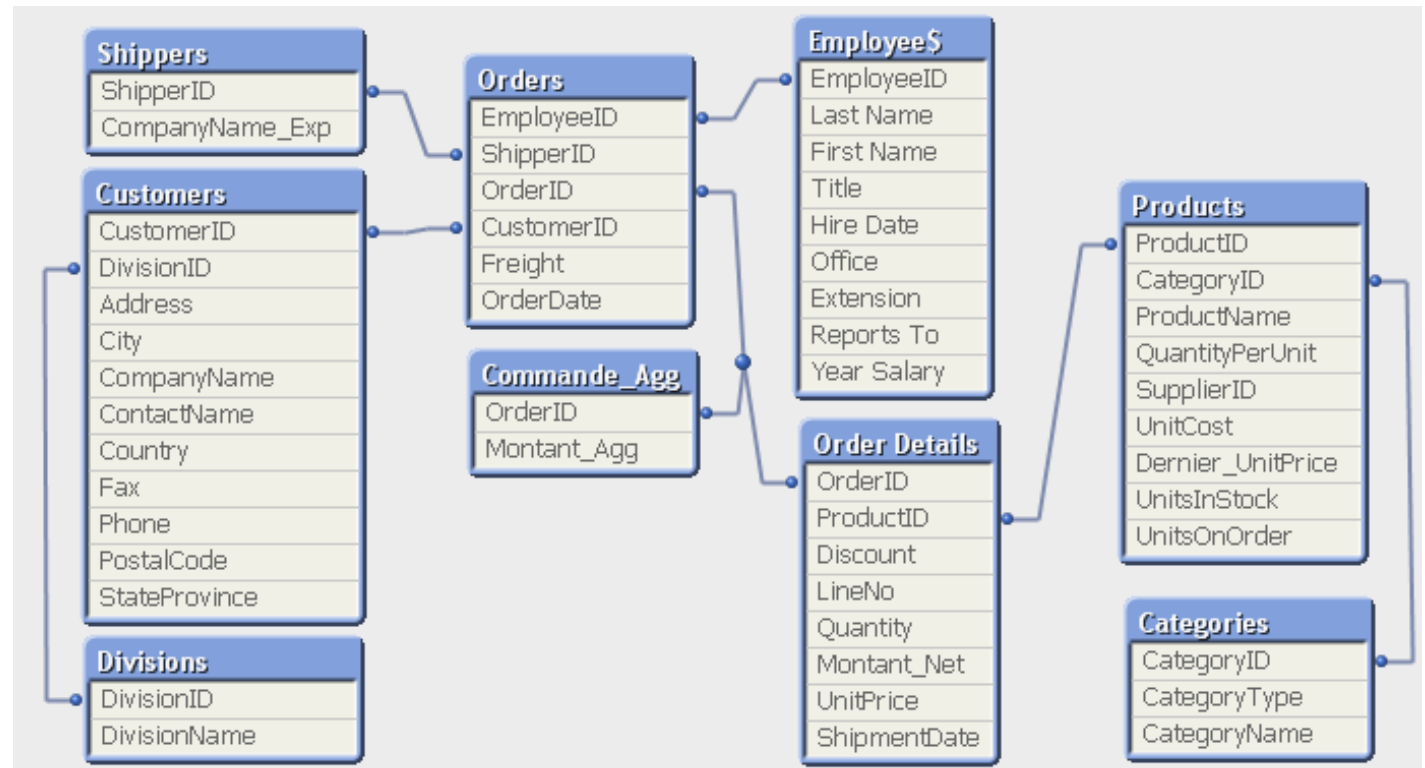
Agrégation des valeurs

Notre utilisateur souhaite calculer la part de l'indicateur Montant_Net d'une ligne de commande (OrderID, LineNo) par rapport au Montant_Net de la commande (OrderID)

Pour cela il a besoin que vous calculiez la somme des Montant_Net par OrderID à partir du champ précédemment calculé. On nommera ce champ Montant_Agg

Le résultat attendu :

Remarque : ce modèle est optimisable



Script avancé - Concatenate

Instruction CONCATENATE

A l'aide de l'assistant d'import de fichier, charger le fichier des nouveaux employés : **Employees_New.xls**

A l'aide de l'instruction CONCATENATE, ajouter les nouveaux enregistrements dans la table existante des Employes.

Remarque : il est possible d'utiliser une 2eme solution en changeant légèrement le script existant à l'aide d'un joker *

Lorsque l'on charge 2 tables avec exactement la même structure, Qlikview fait un concatenate implicite

```
LOAD EmpID as EmployeeID,  
    [Last Name],  
    [First Name],  
    Title,  
    [Hire Date],  
    Office,  
    Extension,  
    [Reports To],  
    [Year Salary]  
FROM  
[C:\Cours developpeur\Datasources\Emp*.xls]  
(biff, embedded labels, table is [Employee$]);
```



Script avancé - Resident, Where Exists()

L'utilisateur souhaite un champ Nom_Vendeur qui sera stocké dans une table Vendeur

Ce champ sera alimenté par le champ [Last Name] de la table des employés

Mais il ne devra contenir que les employés ayant réellement passés une commande

Indications :

Dans la table **Orders**, il faudra dupliquer le champ **EmployeeID** en le renommant (en **empvendeur** par exemple)

C'est nécessaire pour retrouver uniquement les employés ayant passé une commande car le champ **EmployeeID** s'alimentent avec toutes les nouvelles valeurs qu'il rencontre sur les chargements successifs (donc lors du chargement de la table des employés)

Résultat attendu :

Nom_Vendeur
Brolin
Callins
Carsson
Hendrix
Lindwall
Presley
Roll
Shine
Skoglund



Fichiers de données QVD - Store

Store permet d'exporter une table (tous ses champs ou une partie) dans un fichier .QVD

Exemple sur la table Orders :

```
STORE Orders INTO c:\temp\Orders .qvd;
```

```
STORE Orders INTO c:\temp\Orders .txt (txt); // permet de vérifier le contenu au format text.
```

Remarque :

NoOfTables() permet de connaître le nombre de tables

TableName() permet de récupérer le nom d'une table par rapport à son indice

Pour exporter toutes les tables, on peut effectuer une boucle à la fin du script

```
for i = 0 to NoOfTables()-1
```

```
    LET d = TableName(i);
```

```
    store $(d) into c:\temp\$(d).QVD;
```

```
next
```



Questions / Réponses

