

**Province de Namur  
Haute École de Namur-Liège-Luxembourg  
Campus IESN  
Rue Joseph Calozet 19  
5000 Namur**

**Business intelligence**

**Création d’un ETL**Adrien Leboutte – Danielysson Vasconcelos

2019-2020

Table des matières

[1. Description du projet 4](#_Toc29165535)

[2. Diagrammes 5](#_Toc29165536)

[2.1 Diagramme d’action de la base de données opérationnelle fournie 5](#_Toc29165537)

[2.2 Diagramme du data warehouse 6](#_Toc29165538)

[2.3 Granularité de la table de faits 6](#_Toc29165539)

[3. Justification des choix 7](#_Toc29165540)

[3.1 Les clés primaires du « Data Warehouse » 7](#_Toc29165541)

[3.2 Modélisation des dimensions et faits 7](#_Toc29165542)

[3.2.1 DimDate 7](#_Toc29165543)

[3.2.2 DimProduct 7](#_Toc29165544)

[3.2.3 DimCustomer 8](#_Toc29165545)

[3.2.4 DimSeller 8](#_Toc29165546)

[3.2.5 FactSale 8](#_Toc29165547)

[4. Limites rencontrées 9](#_Toc29165548)

[5. Stratégie d’exécution de l’ETL 9](#_Toc29165549)

[6. Gestion des erreurs 9](#_Toc29165550)

1. Description du projet

Dans le cadre de ce projet, il a été demandé de réaliser un « ***data warehouse*** » en respectant la marche à suivre ETL (***Extract, Transform, Load***) afin de l’utiliser pour de l’analyse informatique décisionnelle.

Nous avons dû extraire les données d’une base de données opérationnelle mais aussi les filtrer, les compléter et les nettoyer pour les restructurer sous la forme d’une base de données décisionnelle. On pourra ensuite prendre cette base pour la création d’un cube OLAP.

Nous n’avons pas élaboré de cube pour notre projet.

Nous avons structuré notre base de données décisionnelle sous la forme d’un schéma en étoile (ou en anglais « ***Star scheme*** »). Cela permettra à un utilisateur métier, une meilleure compréhension pour analyser. En plus de cela, le schéma en étoile réduits le nombre de joints (jointures), pour avoir plus de performance, mais il y aura beaucoup de redondance.

***Les composants impliqués :***

* **La base de données opérationnelle fournie pour le projet**, étant donné que le serveur de l’école était inaccessible nous avons travaillé en local sur nos ordinateurs. Pour ce faire nous avons utilisé le script de notre professeur qui a été mis sur Moodle. Le DA de cette base de données se trouve à la page suivante).
* **Un fichier texte (.txt)** avec des dates comprises entre le 1er janvier 2000 et le 31 décembre 2019, plus une date qui est le 1er janvier 1900[[1]](#footnote-1).

***Outils utilisés :***

* **Visual Studio 2019 (Entreprise Edition)**: *auquel on a greffé l’extension « SQL Server Data Tools » permettant d’utiliser les outils nécessaires à la réalisation de l’ETL et du data warehouse.*
* **SQL Server 17 (Develloper Edition)**
* **Microsoft SQL Server Management Studio :** *auquel nous avons utilisée pour utiliser nos scripts de création de base de données.*

1. Diagrammes

## Diagramme d’action de la base de données opérationnelle fournie

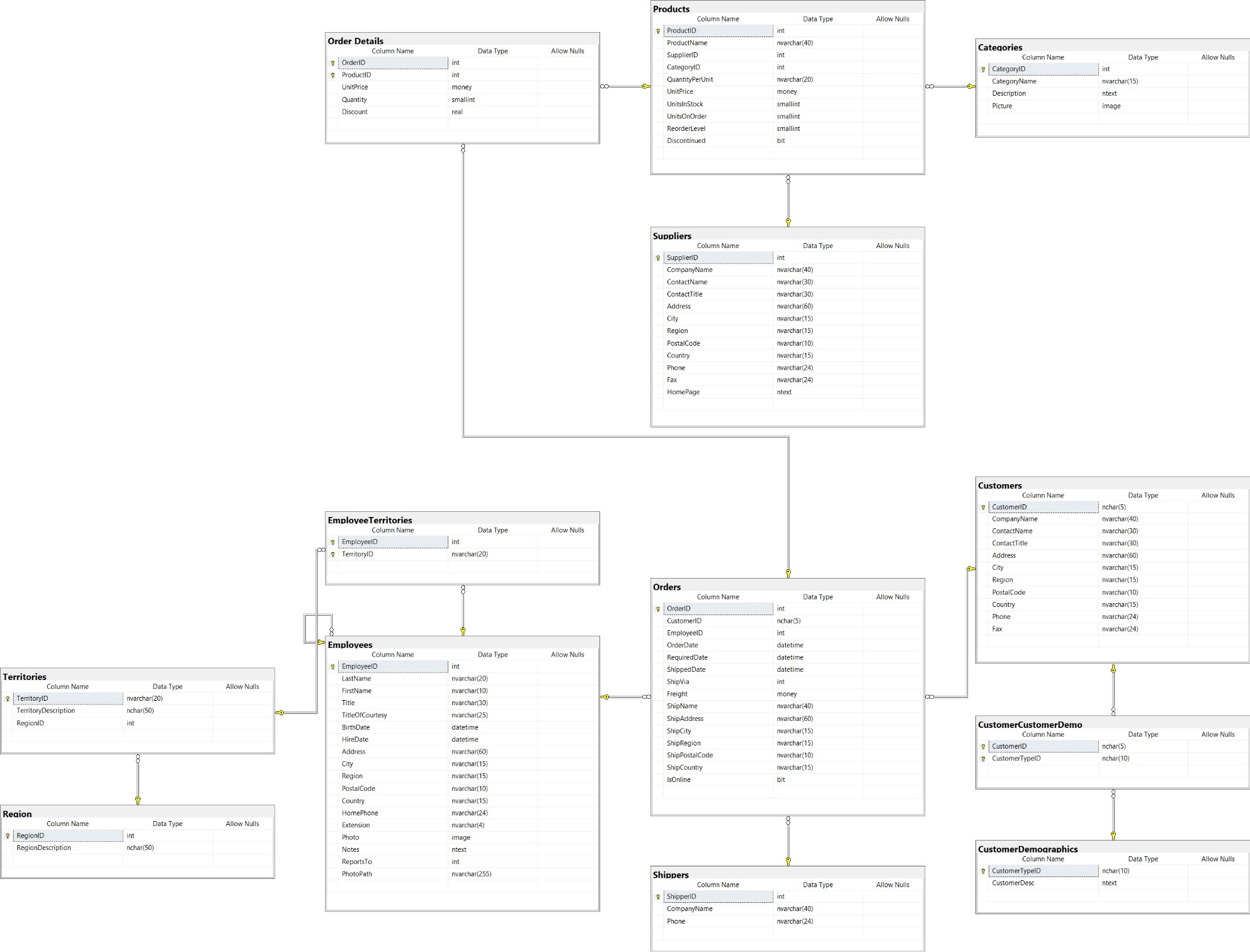


Figure 1 - Diagramme de la base de données opérationnelle

## 2.2 Diagramme du data warehouse

Figure 2- Diagramme de notre data warehouse

## 2.3 Granularité de la table de faits

Nous avons choisi une ligne d’un article d’une commande pour la granularité de la table de faits. Cela permet un niveau de structure plus fin et offre la possibilité aux utilisateurs de creuser

1. Justification des choix

## 3.1 Les clés primaires du « Data Warehouse »

Nous avons décidé de travailler avec des « *Surrogate keys* », nous avons déplacé les identifiants importés dans d’autres champs. ***DimCustomer***, ***Dimproduct*** et ***FactSale*** utilisent des « Surrogate keys ».

Cependant, ***DimDate*** et ***DimSeller*** ont des identifiants qui ne sont pas des « *Surrogate keys* ». La clé primaire de DimDate est composée d’un nombre représentant la date comme suit : YYYYMMDD (YYYY : année, MM : mois, DD : jour).

Les identifiants artificiels ont plusieurs avantages. D’abord, ils empêchent de possibles risques de collisions s’il y a plusieurs imports (comme cela pourrait arriver lorsque les données de la chaîne « MirkaProShop» seront ajoutées). Ensuite, cela permet de se préserver de changements de structure apportés dans la base de données importée et de possibles doublons dont on avait décidé de ne pas écraser les données précédentes chaque année. Enfin, cela permet de maximiser les performances au niveau de l’index puisque l’identifiant est une seule colonne de type entier.

* 1. Modélisation des dimensions et faits

### 3.2.1 DimDate

La dimension « Date » représente les enregistrements relatifs aux dates. Nous avons utilisé et importé un fichier texte, puis nous l’avons transformé pour pouvoir l’utilisé dans les formats souhaités. Nous avons ajouté via manipulation les noms des jours et les noms des mois dans les 2 langues qui sont l’anglais et le français. Nous avons rajouté de la redondance via les colonnes qui indiquent le jour, le mois, l’année, la date entière, le numéro du jour dans la semaine et le numéro du jour dans l’année.

### 3.2.2 DimProduct

La dimension « Product » représente les enregistrements relatifs aux produits vendus par la chaîne de magasins. Nous avons fusionné les différentes tables relatives aux produits de la base de données opérationnelle en une seule dimension (les tables qui ont été fusionnées : Suppliers, Product et Categories). Nous avant modifié certains champs qui étaient à « NULL » pour les remplacer par la chaine de caractères « Inconnue », cela permettra par la suite d’éviter les erreurs si l’on interdit certains champs à NULL.

### 3.2.3 DimCustomer

La dimension « Customer » représente les enregistrements relatifs aux clients. Pour cette dimension nous avons juste repris la table dans la base de données opérationnels et avons modifié les champs avec la valeur « ***NULL*** » par la chaine de caractères « Inconnue ». Nous avons décidé de ne pas prendre en compte les tables CustomerCustomerDemo et CustomerDemographics de la base de données opérationnelles.

### 3.2.4 DimSeller

La dimension « Seller » représente les enregistrements relatifs aux employés. Nous avons fusionné les différentes tables relatives aux employés de la base de données opérationnelle en une seule dimension (les tables qui ont été fusionnées : ***Territories***, ***Region,EmployeeTerrotories*** et ***Employees***).

Pour des soucis de performance et de lisibilité, nous avons évités de mettre dans la base de données opérationnels de ne pas mettre la photo de l’employé. Elle est peu compréhensible pour ceux qui manipule, mais le chemin d’accès à la photo est présent pour récupérer les photos si besoin.

Nous avons aussi modifié les champs avec la valeur « ***NULL*** » par la chaine de caractères « Inconnue ».

### 3.2.5 FactSale

La table de faits regroupe les enregistrements relatifs aux lignes des ventes. Nous avons fusionné les différentes tables relatives aux employés de la base de données opérationnelle en une seule dimension (les tables qui ont été fusionnées : ***Shippers***, ***Orders***, ***OrderDetails***).

Ensuite, nous avons fait des recherches pour lier et récupérer des données se trouvant dans les autres tables du data warehouse.

Nous avons calculé le prix total de 2 façons qui été mis dans 2 colonnes différentes. La 1ère façon pour le prix total est de calculé le prix du produit multiplié par les quantités demandées, et la 2ème façon est de faire comme la 1ère mais en multipliant en plus (***1 – discount du produit***).

Nous avons aussi modifié les champs avec la valeur « ***NULL*** » par la chaine de caractères « Inconnue ».

Enfin, nous avons ajouté une colonne qui indique la marque du produit, ici « ***Foodies997*** ».

# Limites rencontrées

D’abord, nous avons les données de la marque de produits alimentaires « ***Foodies997*** » mais pas celle de « ***MirkaProShop*** ». Le Data Warehouse ne permet donc pas, par exemple, de répondre à la question du métier concernant une comparaison entre les deux marques de produits alimentaires.

Ensuite, le multilinguisme a ces limites car à part les noms des jours ou des mois, nous n’avons pas adapté le multilinguisme pour tous. Il aurait été préférable d’avoir des fichiers qui permettent des traductions pour les noms des villes ou des régions par exemple.

Enfin, il reste le problème des dates qui sont comprises seulement qu’entre 1900 et 1999.Nous avons mis les dates avec la date poubelle /tampon qui est le 1er janvier 1900. Pour résoudre cela nous pourrions utiliser un fichier avec plus de dates.

# Stratégie d’exécution de l’ETL

La stratégie utilisée pour ce projet repose sur une importation annuelle des données. Par soucis de simplification, les données déjà existantes (de l’année précédente) sont écrasées pendant le processus.

Ainsi, s’il arrive que l’ETL s’arrête en plein traitement, il est possible de le relancer depuis le début. Dans le control flow, j’ai défini une première tâche permettant de vider les tables de la base de données du Data Warehouse.

Si on doit relancer le processus, le résultat restera juste.

# Gestion des erreurs

Nous avons constaté qu’il serait mieux que lorsque l’on a des erreurs, de marquer les enregistrements anormaux et de poursuivre l’importation. Mais nous avons préféré mettre les enregistrements qui seraient incorrecte dans des fichiers textes

Pour le cas les dates, nous avons constatées qu’elles se trouvaient entre 1900 et 1999, or notre fichier de dates de départ qui nous permettait d’avoir toutes les dates entre le 1er janvier 2000 et le 31 décembre 2019 (avec la date du 1er janvier 1900) ce qu’il nous provoquait des erreurs. Nous avons décidé de mettre les dates vers une date qui serait une date tampon/poubelle. Le plus adéquates aurait été d’allongé les dates du fichiers texte.

1. Nous l’avons utilisé comme date « poubelle » pour les dates qui ne seraient pas correctes. [↑](#footnote-ref-1)