**Entrepôt de données**

**Rapport Final**

***Sébastien Quiquerez***

***Johan Steiner***

Table des matières

[1. Introduction 4](#_Toc472368015)

[2. Analyse de l’existant 4](#_Toc472368016)

[2.1. Fichier CSV délimité 4](#_Toc472368017)

[2.2. Fichier Microsoft Access Database 4](#_Toc472368018)

[2.2.1. Devises 5](#_Toc472368019)

[2.2.2. Vendeurs 5](#_Toc472368020)

[2.3. Fichier Microsoft Excel 5](#_Toc472368021)

[2.4. Fichier texte non délimité 5](#_Toc472368022)

[2.5. Fichier texte délimité 6](#_Toc472368023)

[2.6. Transmission d’informations sans fichier 6](#_Toc472368024)

[2.7. Structure et base de données 6](#_Toc472368025)

[3. Analyse des besoins 6](#_Toc472368026)

[3.1. Définition des contraintes sur les données 6](#_Toc472368027)

[3.2. Importation/transformation automatique des données 7](#_Toc472368028)

[3.3. Gestion des erreurs 7](#_Toc472368029)

[3.4. Structure de stockage unique 7](#_Toc472368030)

[3.5. Indicateurs 7](#_Toc472368031)

[3.6. Mise en place de hiérarchies 8](#_Toc472368032)

[3.7. Contrôle 8](#_Toc472368033)

[4. Cahier des charges 8](#_Toc472368034)

[5. Modèle dimensionnel en étoile 9](#_Toc472368035)

[5.1. Explications des choix 10](#_Toc472368036)

[5.1.1. Tables de faits 10](#_Toc472368037)

[5.1.2. Mini dimension Commande 10](#_Toc472368038)

[5.1.3. Dimension Temps : Liaisons multiples 10](#_Toc472368039)

[5.1.4. Dimensions Client et Geographie : flocon 10](#_Toc472368040)

[5.2. Hiérarchies 11](#_Toc472368041)

[5.2.1. Hiérarchie du temps 11](#_Toc472368042)

[5.2.2. Hiérarchie des vendeurs 11](#_Toc472368043)

[5.2.3. Hiérarchie des produits 12](#_Toc472368044)

[5.2.4. Hiérarchie de la géographie 12](#_Toc472368045)

[6. Données : importation/transormation 12](#_Toc472368046)

[6.1. Staging 12](#_Toc472368047)

[6.1.1. Types de données 13](#_Toc472368048)

[6.2. Data Warehouse 13](#_Toc472368049)

[6.2.1. Master data 13](#_Toc472368050)

[6.2.2. Qualité des données 13](#_Toc472368051)

[6.2.3. Audit 13](#_Toc472368052)

[6.2.4. Importation des prix 13](#_Toc472368053)

[6.3. Cube 13](#_Toc472368054)

[7. Indicateurs 14](#_Toc472368055)

[8. Conclusion 14](#_Toc472368056)

[9. Bibliographie 14](#_Toc472368057)

# Introduction

Mediazon est spécialisée dans la vente de produits multimédias. L’entreprise nous a mandaté pour mettre en place la structure de son Data Warehouse qui stockera de grandes quantités de données issues des commandes de ses clients.

Les données source sont donc sous une forme « brute », provenant de différentes sources, avec un certain nombre d’erreurs et il est nécessaire de mettre en place un système permettant d’obtenir des données d’une certaine qualité afin de permettre le développement d’un système décisionnel bien conçu.

Le développement du système de test a été réalisé depuis une machine virtuelle Microsoft Server 2012. Le logiciel Microsoft Visual Studio ainsi que Microsoft SQL Server Management Studio nous ont permis de réaliser la plus grande partie de notre travail.

# Analyse de l’existant

Au départ, les données sont dispersées entre plusieurs sources de données de différents types. A notre disposition, nous avons les types de sources suivants :

* Fichier CSV délimité
* Fichier Microsoft Excel
* Fichier Microsoft Access Database
* Fichier texte non délimités
* Fichier texte délimité
* Transmission d’informations sans fichier

Les informations contenues dans les différents fichiers sont décrites ci-dessous.

## Fichier CSV délimité

Dans ce fichier, les différentes villes de France sont présentes, avec leur NPA respectif et le département dans lequel elles se situent. Chaque ville est identifiée par un ID unique.

## Fichier Microsoft Access Database

Ce fichier contient 2 catégories de données distinctes.

### Devises

Mediazon livre ses produits dans plusieurs régions du monde, ce qui signifie qu’elle doit adapter la devise de paiement en fonction de ces régions. Elle possède donc la liste des devises présentes dans le monde avec, pour chacune d’elles, un code à trois chiffres uniques.

### Vendeurs

L’entreprise possède également une liste de ses vendeurs avec pour chacun d’eux son nom et prénom, la division et le groupe dans laquelle il travaille, ainsi que son manager respectif.

En plus de cela, des informations plus personnelles telles que le sexe de la personne, son quota, bonus ou commission sont renseignées. Les vendeurs sont aussi identifiés à l’aide d’un ID unique.

## Fichier Microsoft Excel

Un fichier Microsoft Excel composé de la liste des différents produits que l’entreprise vend nous a été mis à disposition.

On peut y trouver des informations sur le produit comme par exemple son titre, son auteur, son prix (catalogue et de base) ou encore les catégories et sous-catégories auxquelles il appartient. La référence du produit permet de l’identifier car elle est unique.

## Fichier texte non délimité

Le fichier texte non délimité ne contient pas de délimiteur (virgule, point-virgule, etc…) mais chaque « champ » a un nombre de caractères défini. Si le contenu de ce champ est plus petit que le nombre défini, les caractères disponibles restants sont comblés par des espaces. Cela crée des « pseudo-colonnes ».

Les fichiers textes ne contiennent pas titres de colonnes, néanmoins nous avons eu les informations sur la signification des valeurs/textes présents dans chaque « colonne ».

Ce fichier contient donc les informations globales des commandes passées par nos clients, leurs « en-têtes ». Le détail de celles-ci se trouve dans un autre fichier, celui-ci nous renseigne sur des éléments plus généraux tels que le numéro de la commande, les différentes dates (commande, livraison prévue, expédition si déjà expédié), la devise, les informations personnelles du client concerné, l’expéditeur de la commande ou encore le code du vendeur ayant réalisé la vente.

Parmi tous ces éléments, des identifiants uniques sont présents, comme le numéro de la commande, un ID pour le client, le code du vendeur ou encore celui de la devise.

## Fichier texte délimité

Ici, ce sont les lignes de nos commandes qui sont présentes. Logiquement, ce fichier contient plus d’enregistrements que celui des en-têtes étant donné qu’une commande contient souvent plusieurs lignes. On s’intéresse donc plus au produit vendu qu’à la commande globale.

On retrouve donc diverses informations sur le produit (titre, auteur, prix), la quantité commandée pour chaque produit et la remise sur le prix unitaire de celui-ci. La marge unitaire est également indiquée.

Chaque produit est identifiable par sa référence qui est unique.

## Transmission d’informations sans fichier

D’autres informations importantes nous ont été transmises mais elles ne sont pas inscrites dans un fichier depuis lequel nous pourrions directement les reprendre. Il s’agit par exemple des informations sur les expéditeurs, par exemple la liste des différentes compagnies d’expédition avec lesquelles l’entreprise travaille, leurs coûts de base et unitaires pour l’expédition des produits.

## Structure et base de données

Au début du projet, la base de données n’est pas encore en place. Les tables n’existent donc pas encore et l’importation des données n’a pas encore été réalisée. Nous partons donc d’une base « vide » et brut (en ce qui concerne les données) à ce niveau.

# Analyse des besoins

Plusieurs besoins ressortent au début du projet.

## Définition des contraintes sur les données

Avant la définition des contraintes, il faudra tout d’abord comprendre le sens des données. Certains fichiers ne contiennent donc pas d’en-tête et ne sont pas forcément délimités. Il faudra savoir ce que chaque colonne de nos données représente et pouvoir créer des « colonnes » correctes.

A l’heure actuelle, les données se trouvent dans un format « brut », ce qui signifie qu’elles sont sous forme de simple texte, sans aucune contrainte de taille, sans interdiction d’une valeur nulle, etc…

Il est donc nécessaire de définir des contraintes sur celles-ci. Il va notamment falloir définir quelle donnée devra obligatoirement ou non obligatoirement être renseignée pour chaque enregistrement. Certaines données nécessiteront des contrôles (CHECK) sur la valeur entrée afin qu’elle respecte par exemple une liste de valeurs autorisées.

## Importation/transformation automatique des données

A partir des données brutes recueillies dans les différents documents cités ci-dessus, nous devons effectuer divers traitements sur les données afin qu’elles respectent les contraintes définies précédemment et qu’elles soient converties dans le bon format (format *décimal* pour des prix par exemple).

Ces traitements devront être automatisés afin qu’à chaque chargement de nouvelles données ou modification de données existantes, les données ressortent dans le même format que celui souhaité et défini. Il ne devrait donc pas y avoir de génération d’erreur de la part du système qui sera mis en place.

## Gestion des erreurs

Il est certain que parmi les données à transformer, certaines généreront des erreurs. Par exemple, s’il y a une liaison à faire avec une autre donnée mais que celle-ci n’est pas trouvée, cela stoppera le processus de traitement.

Il va donc falloir mettre en place une gestion d’erreur appropriée pour que, malgré certaines données erronées, l’ensemble des enregistrements que l’on possède soit traité et, si possible, validé. On pourrait par exemple imaginer remplacer les valeurs erronées par une valeur définie comme « standard pour les erreurs ».

## Structure de stockage unique

Actuellement, nos données sont stockées dans plusieurs types de fichiers. La mise en commun des informations est donc actuellement impossible. En effet, il serait compliqué d’unir les données un document Access avec un document Excel (ou du moins, pas « confortable » pour les combiner).

Il est donc nécessaire que toutes nos données puissent être stockées à un seul et même endroit où il est possible de les utiliser ensemble afin de pouvoir mettre en place des indicateurs par exemple. Dans cette même structure, on pourra les extraire avant leur transformation, cela permettra de donner à nos processus de traitements une source unique où trouver les données qui provenaient auparavant des divers types de fichiers à notre disposition.

## Indicateurs

Une fois que les données auront été transformées et importées dans leur version finale, elles pourront être utilisées pour créer des indicateurs.

Ceux-ci devront être définis de manière cohérente afin qu’ils soient pertinents. Le but ne sera pas d’en avoir en qualité mais d’avoir des indicateurs de qualité.

## Mise en place de hiérarchies

Pour plusieurs de nos données, il serait possible de créer des hiérarchies qui permettraient de créer des indicateurs de différentes granularités (unité de temps : jour / mois / année).

## Contrôle

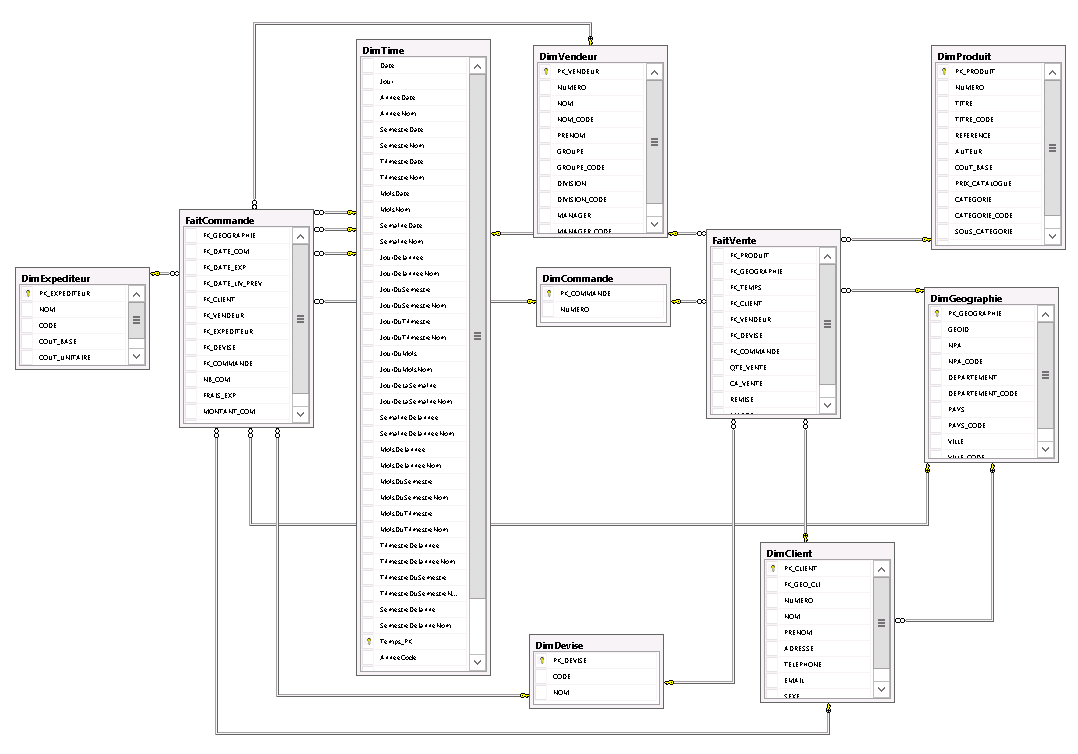
Lors du chargement des données, il sera nécessaire de mettre en place des éléments permettant de contrôler le processus : est-ce que celui-ci s’est bien déroulé ? Y a-t-il eu des éléments qui ont créé des problèmes ?

# Cahier des charges

Un cahier des charges pour le projet a été défini.

|  |  |
| --- | --- |
| **N° tâche** | **Tâche** |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |

# Modèle dimensionnel en étoile



## Explications des choix

### Tables de faits

Notre schéma contient 2 tables de faits : la table **Commande** et la table **Vente**.

La table *Vente* contient les informations propres aux produits et plutôt à une ligne de la commande (En quelle quantité un produit a-t-il été acheté ? De quel pourcentage de remise a-t-on bénéficié sur chaque produit ? Quelle marge a été réalisée sur la vente ? Etc…).

La table *Commande* est plus orientée sur la commande en générale, par exemple pour connaître les dates de commande, livraison et expédition, l’expéditeur ainsi que ses frais ou encore le client qui a passé la commande.

### Mini dimension Commande

Avant la création de cette dimension, il était impossible de lier une commande aux ventes. On ne pouvait par exemple pas connaître le prix total de la commande.

La mini dimension Commande permet de créer cette liaison : elle contient le numéro de chaque commande et dans chaque table de faits, une clé secondaire référence cette table afin qu’il soit possible de faire correspondre les informations des deux tables de faits.

Par exemple, si l’on souhaite connaitre le montant total des produits d’une commande (laissons de côté les frais d’expédition pour cet exemple), il suffit de multiplier les prix d’un produit par sa quantié et de répéter cette opération pour chaque produit de la commande. Grâce à notre mini dimension Commande, nous pouvons lier le numéro de commande de la table de faits *Commande* à la table de faits *Vente* et donc savoir quel produit appartient à quelle commande.

### Dimension Temps : Liaisons multiples

Nous avons créé 3 liaisons entre la dimension Temps et la table de faits Commande, car dans le cas, il est mentionné 3 types de dates distinctes :

* La date de commande
* La date de livraison prévue
* La date d’expédition

### Dimensions Client et Geographie : flocon

Un flocon a été mis en place entre les tables Client et Geographie car nous avons jugé intéressant de connaitre la provenance de nos clients sans qu’on doive forcément se lier à une vente. Cela permettra la mise en place d’indicateurs supplémentaires.

## Hiérarchies

### Hiérarchie du temps

**Année**

**Année**

**Semaine**

**Mois**

**Semestre**

**Jour**

**Jour**

**Trimestre**

**Mois**

**Jour**

### Hiérarchie des vendeurs

**Groupe**

**Division**

**Manager**

**Vendeur**

### Hiérarchie des produits

**Catégorie**

**Sous-catégorie**

**Article**

### Hiérarchie de la géographie

**Pays**

**Département**

**Ville**

# Données : importation/transormation

L’importation des données dans notre Data Warehouse a été réalisée à l’aide de la méthode ETL (Extract, Transform, Load).

## Staging

La zone staging représente la zone contenant les données qui n’ont pas encore été transformées. Nous extrayons les données depuis nos différentes sources (fichiers Excel, fichiers Access, fichiers texte, etc…), les plaçons dans notre staging area, à la suite de quoi nous les transformons (conversion de type, modification de la longueur des champs, etc…) pour qu’elles correspondent à ce qu’attendent les tables présentes dans notre Data Warehouse et les chargeons dans celui-ci.

### Types de données

L’importation des données dans la staging a été simplifiée au niveau des types de données. En effet, nous n’avons à ce moment-là pas encore procédé à une transformation des différents types, mais avons simplement importé toutes les données sous forme de chaînes de caractères (sauf les identifiants).

## Data Warehouse

### Master data

### Qualité des données

### Audit

### Importation des prix

L’un de nos choix d’importation concerne l’importation des prix et calcul du montant de la commande : du fait que les commandes sont payées en différentes devises, nous avions en effet le choix entre une transformation en une devise unique avant l’importation dans notre Data Warehouse ou une simple importation dans la devise de base.

Nous avons choisi de garder la devise de base lors de l’importation dans nos tables de fait. Ce choix s’explique du fait que nous avons pris en considération les variations du taux de change des devises. Voici un exemple :

Admettons que nous possédons un livre qui coûte 100€ avec un taux de change de 1.2 entre l’euro et le franc suisse (CHF). Admettons encore que quelques mois plus tard, ce taux a diminué à 1.1. Si nous avions converti le prix de notre produit en CHF à l’importation, celui-ci aurait valu 120 CHF et, dans le cas où nous aimerions savoir son prix actuel en magasin en euro, il vaudrait théoriquement 109€ alors qu’en magasin, il est certainement toujours vendu à 100€, soit actuellement 110 CHF.

Nous avons donc préféré opter pour un prix « à jour », qui sera, si besoin, converti en temps réel avec le taux de change actuel pour sa monnaie.

## Cube

Une fois l’importation des données dans notre Data Warehouse réalisée, nous avons pu passer à la génération de notre cube.

# Indicateurs

# Conclusion

# Bibliographie