



Document de conception – Projet ODE

|  |  |  |
| --- | --- | --- |
| **Suivi des versions** | | |
| **Date** | **Version** | **Objet de la version** |
| 08/07/2015 | 01 | [Olivier] Création du document |
|  |  |  |
|  |  |  |

Sommaire

[Introduction 3](#_Toc424116529)

[Aspects métier 3](#_Toc424116530)

[Quelques chiffres 3](#_Toc424116531)

[Structure organisationnelle 4](#_Toc424116532)

[Architecture de la chaine décisionnel 5](#_Toc424116533)

[Vue d’ensemble 5](#_Toc424116534)

[Partie ETL 5](#_Toc424116535)

[Partie DWH / Cubes 5](#_Toc424116536)

[Partie Reporting 6](#_Toc424116537)

[Base opérationnelle 6](#_Toc424116538)

[Vue d’ensemble 6](#_Toc424116539)

[Table UNIVERS\_PRODUITS 6](#_Toc424116540)

[Table RAYONS\_PRODUITS 7](#_Toc424116541)

[Table FAMILLES\_PRODUITS 7](#_Toc424116542)

[Table SOUS\_FAMILLES\_PRODUITS 7](#_Toc424116543)

[Table PRODUITS 8](#_Toc424116544)

[Table CLIENTS 8](#_Toc424116545)

[Table LIEUX 9](#_Toc424116546)

[Table VILLES 9](#_Toc424116547)

[Table STOCKS 10](#_Toc424116548)

[Table VENTES 10](#_Toc424116549)

[Table TICKETS 11](#_Toc424116550)

[Table PRIXPRODUITS 11](#_Toc424116551)

[Entrepôt de données 12](#_Toc424116552)

[Vue d’ensemble 12](#_Toc424116553)

[Table de dimension CATEGORIES 12](#_Toc424116554)

[Table de dimension PRODUITS 13](#_Toc424116555)

[Table de dimension TEMPS 13](#_Toc424116556)

[Table de dimension LIEUX 14](#_Toc424116557)

[Table de dimension CLIENTS 14](#_Toc424116558)

[Table de dimension VILLES 15](#_Toc424116559)

[Table de faits VENTES 16](#_Toc424116560)

[Table de faits STOCKS 16](#_Toc424116561)

# Introduction

Le projet « Optimisation des Données de l’Entrepôt (ODE) » consiste à utiliser les techniques mathématiques vues dans le Master 2 MIAGE afin de construire l'entrepôt de données (Data-Warehouse - DWH) de manière optimal, en termes de temps de réponse à l’interrogation des cubes et d’occupation disque.

# Aspects métier

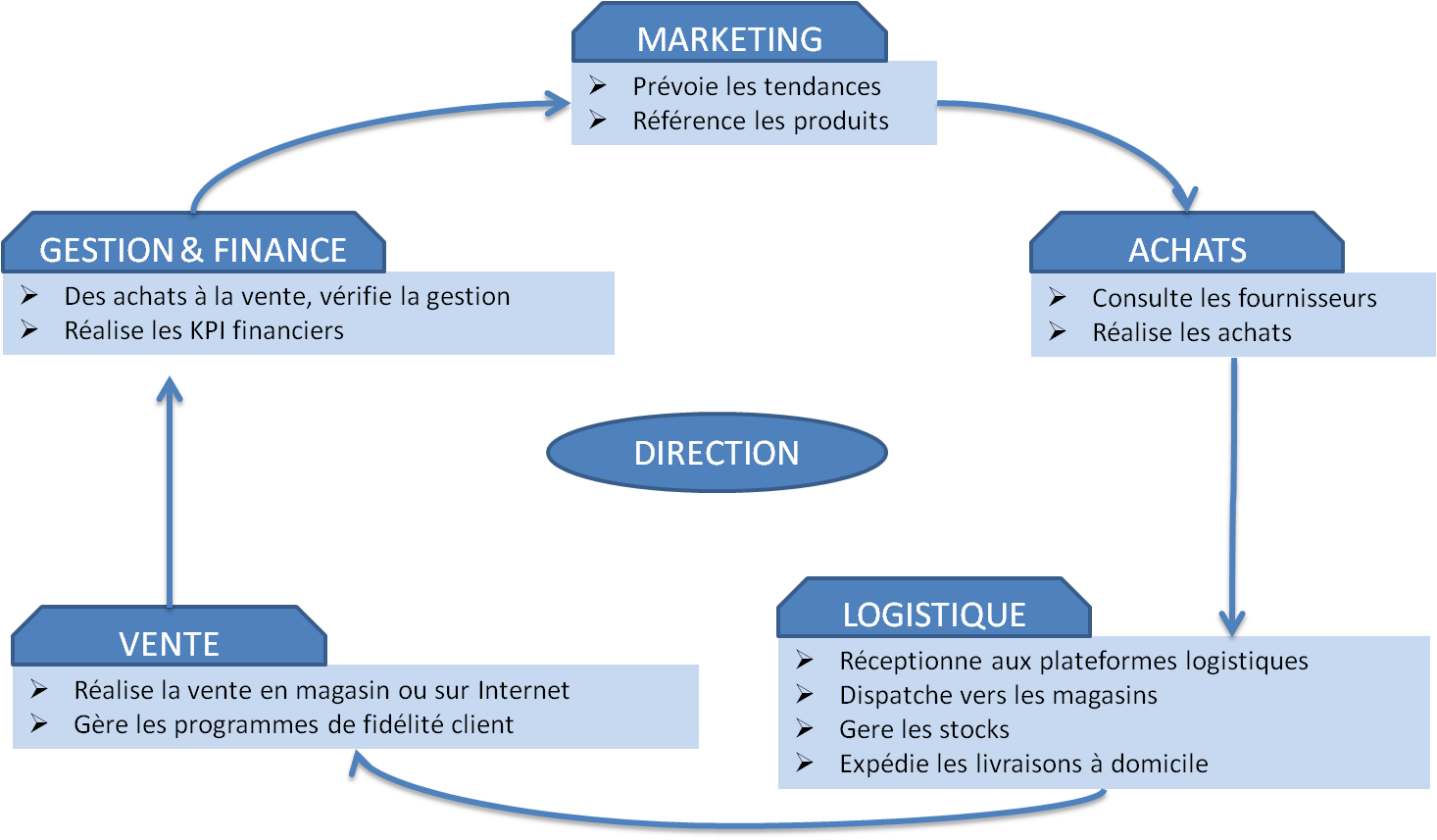
CASTO-MERLIN : Grande distribution de bricolage

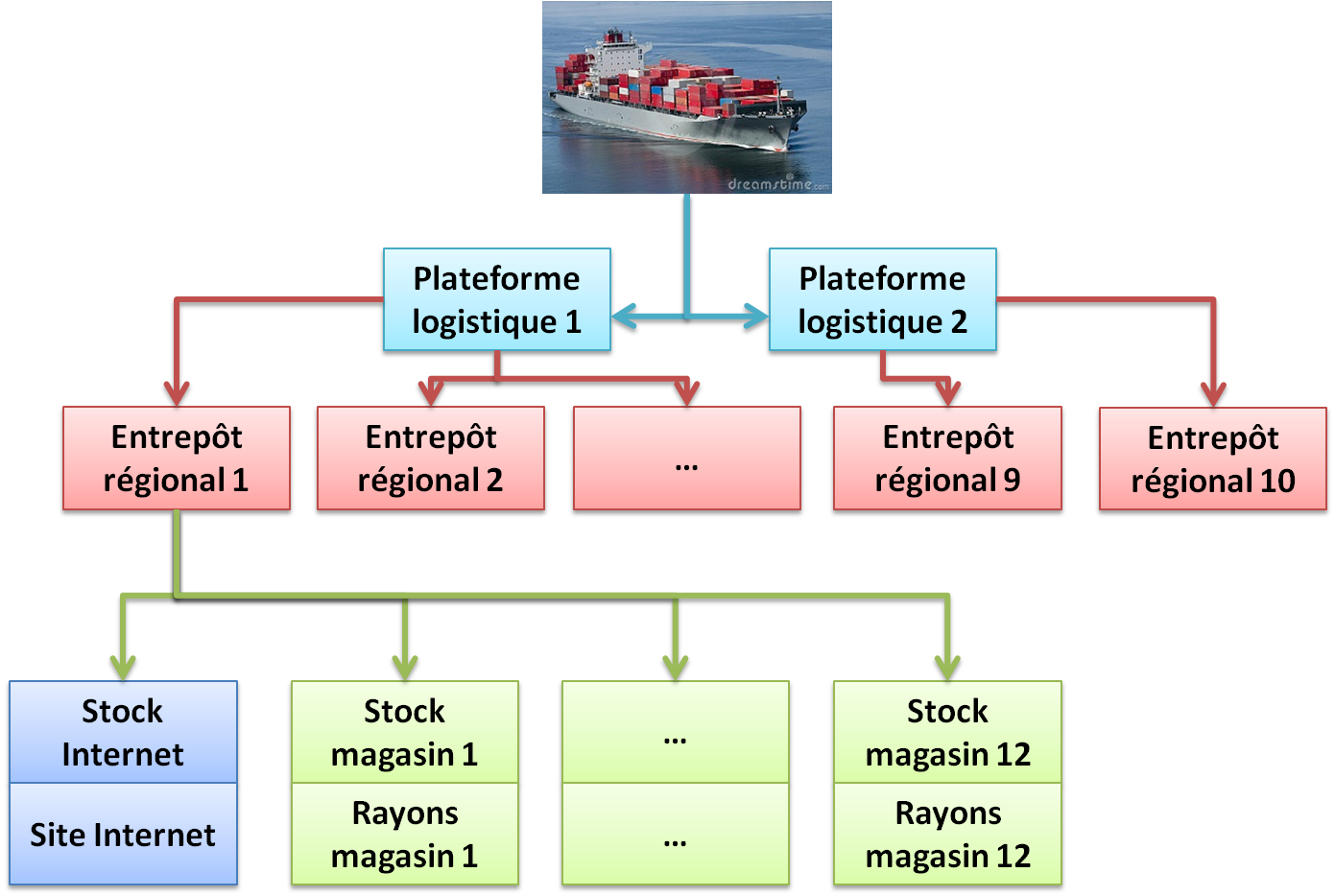


## Quelques chiffres

* 31 300 collaborateurs
* 2 plateformes logistiques nationales
* 10 centrales de distribution régionales
* 121 magasins + 1 site Internet
* 60 000 références magasins plus 20 000 « sur commande »
* 3 856 fournisseurs

## Structure organisationnelle





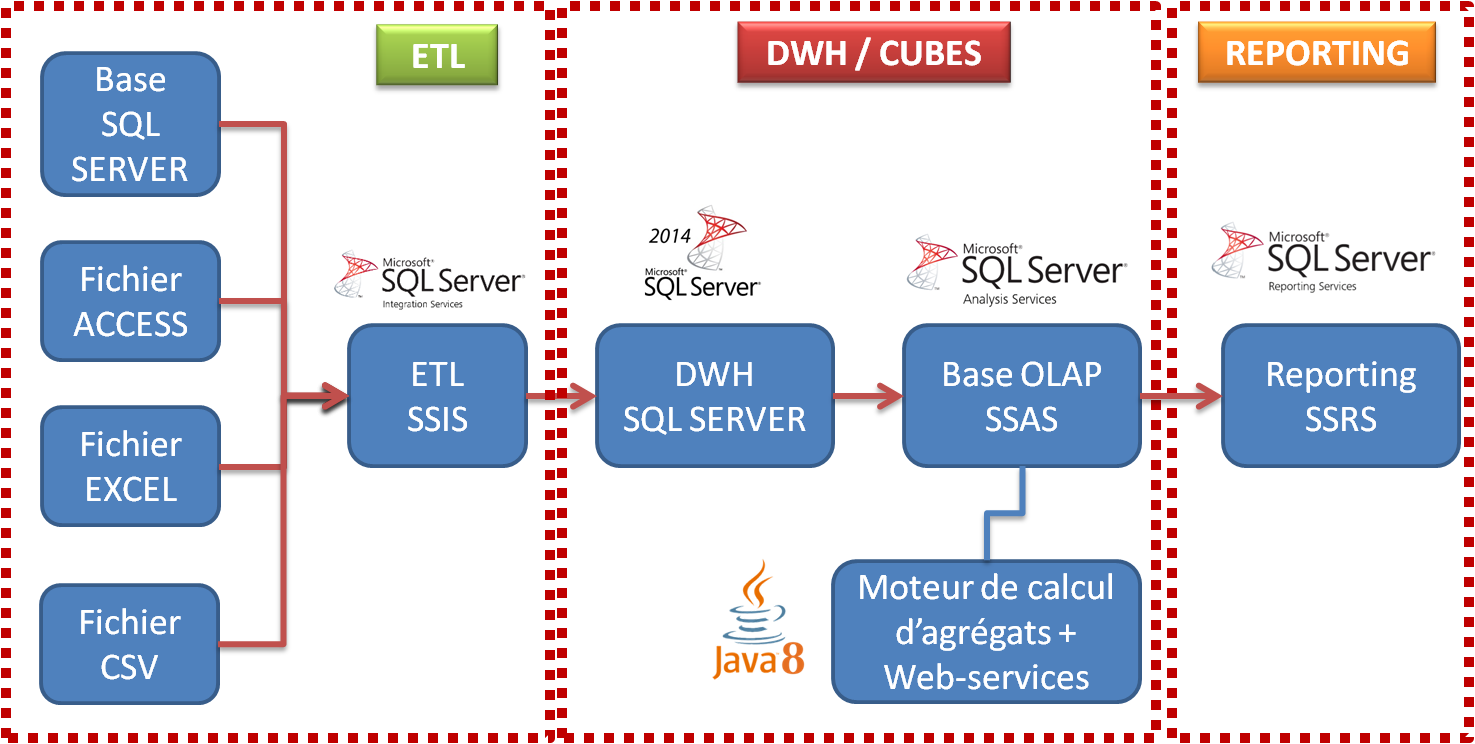
# Architecture de la chaine décisionnel

## Vue d’ensemble

Les données opérationnelles de l’entreprise, c'est-à-dire celle générées et utilisés pour la gestion de l’activité courante de l’entreprise, sont stockées principalement dans une base OLTP SQL Server 2014. Nous la désignerons par « base opérationnelle » dans la suite de ce document.

D’autres sources de données sont possibles :

* Fichiers « plats » type CSV
* Fichiers EXCEL
* Fichiers ACCESS



## Partie ETL

L’ETL sera en charge de collecter les informations à partir des différentes sources de données opérationnelles de l’entreprise :

* Base opérationnelle
* Fichiers CSV
* Fichiers EXCEL
* Fichiers ACCESS

Cette partie sera réalisée avec **SQL Server 2014 – Integration Services (SSIS)**

## Partie DWH / Cubes

C’est le « cœur » de la chaine décisionnelle que nous allons réaliser.

Le Datawarehouse (DWH – Entrepot de données) est modélisé en flocon, et hébergé sur une base OLTP **SQL Server 2014.**

Le cube sera hébergé sur une base OLAP **SQL Server 2014 Analysis Service (SSAS)**

Le moteur de calcul d’agrégats est un programme Java servant à « optimiser » la structure et le calcul des cubes (Agrégats, cf. cours du D111)

## Partie Reporting

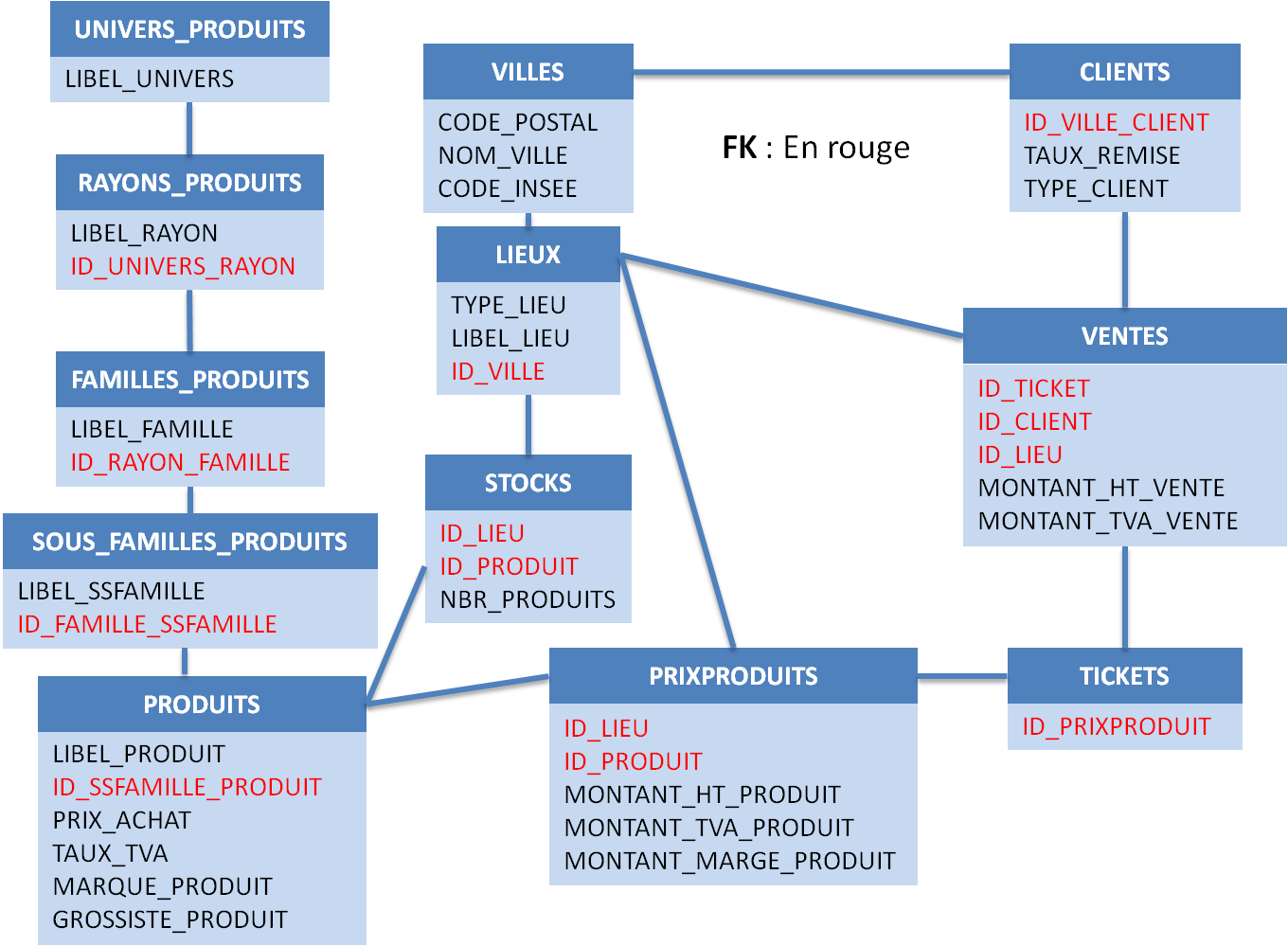
La chaine décisionnelle génère des reports sur un sujet « métier ». Par exemple : Calcul du CA, de la marge, évolution des ventes…

Cette partie sera réalisée avec **SQL Server 2014 – Reporting Services (SSRS)**

# Base opérationnelle

## Vue d’ensemble

Dans la base dédiée « **BaseOperationelleODE** », nous créons un schéma dédié « **ODE\_VENTES** », qui contient les tables suivantes :



## Table UNIVERS\_PRODUITS

But

D

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_UNIVERS | int | Non |
| LIBEL\_UNIVERS | nvarchar(256) | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table RAYONS\_PRODUITS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_RAYON | int | Non |
| LIBEL\_RAYON | nvarchar(256) | Non |
| ID\_UNIVERS\_RAYON | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table FAMILLES\_PRODUITS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_FAMILLE | int | Non |
| LIBEL\_FAMILLE | nvarchar(256) | Non |
| ID\_RAYON\_FAMILLE | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table SOUS\_FAMILLES\_PRODUITS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_SSFAMILLE | int | Non |
| LIBEL\_FAMILLE | nvarchar(256) | Non |
| ID\_RAYON\_SSFAMILLE | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table PRODUITS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_PRODUIT | int | Non |
| LIBEL\_PRODUIT | nvarchar(256) | Non |
| DESC\_PRODUIT | nvarchar(1024) | Non |
| ID\_SSFAMILLE\_PRODUIT | int | Non |
| CODE\_BARRE\_PRODUIT | int | Non |
| PRIX\_ACHAT | money | Non |
| TAUX\_TVA | decimal(4, 1) | Non |
| MARQUE\_PRODUIT | nvarchar(256) | Non |
| GROSSISTE\_PRODUIT | nvarchar(256) | Oui |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table CLIENTS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_CLIENT | int | Non |
| NOM\_CLIENT | nvarchar(256) | Non |
| TYPE\_CLIENT | char(1) | Non |
| DATE\_NAISSANCE | date | Non |
| DATE\_SOUSCRIPTION | date | Non |
| ID\_VILLE\_CLIENT | int | Non |
| LIBEL\_ADRESSE | nvarchar(256) | Oui |
| CODE\_CARTE\_FIDEL | nvarchar(32) | Oui |
| TAUX\_REMISE | decimal(4, 1) | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table LIEUX

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_LIEU | int | Non |
| TYPE\_LIEU | char(1) | Non |
| LIBEL\_LIEU | nvarchar(256) | Non |
| ID\_VILLE | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table VILLES

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_VILLE | int | Non |
| CODE\_POSTAL | nvarchar(6) | Non |
| CODE\_COMMUNE | tinyint | Non |
| CODE\_REGION | tinyint | Non |
| CODE\_DEPARTEMENT | tinyint | Non |
| CODE\_ARRONDISEMENT | tinyint | Non |
| CODE\_CANTON | tinyint | Non |
| NOM\_VILLE\_MAJ | varchar(64) | Oui |
| NOM\_VILLE\_MIN | varchar(64) | Oui |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table STOCKS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_STOCK | int | Non |
| DATE\_RECENSEMENT | datetime | Non |
| OPER\_RECENSEMENT | nvarchar(64) | Non |
| ID\_LIEU | int | Non |
| ID\_PRODUIT | int | Non |
| NBR\_DISPO | int | Non |
| NBR\_DEFECTUEUX | int | Non |
| NBR\_RETOUR\_SAV | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table VENTES

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_VENTE | int | Non |
| OPER\_VENTE | nvarchar(64) | Non |
| ID\_TICKET | int | Non |
| ID\_CLIENT | int | Non |
| ID\_LIEU | int | Non |
| MONTANT\_HT\_VENTE | money | Non |
| MONTANT\_TVA\_VENTE | money | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table TICKETS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_TICKET | int | Non |
| ID\_PRIXPRODUIT | int | Non |
| QUANTITE | int | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |

Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

## Table PRIXPRODUITS

But

/\* **OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| ID\_PRIXPRODUIT | int | Non |
| OPER\_PRIXPRODUIT | nvarchar(64) | Non |
| ID\_LIEU | int | Non |
| ID\_PRODUIT | int | Non |
| MONTANT\_HT\_PRODUIT | money | Non |
| MONTANT\_TVA\_PRODUIT | money | Non |
| MONTANT\_MARGE\_PRODUIT | money | Non |
| DATE\_CREAT | datetime | Non |
| OPER\_CREAT | nvarchar(64) | Non |
| DATE\_MODIF | datetime | Oui |
| OPER\_MODIF | nvarchar(64) | Oui |

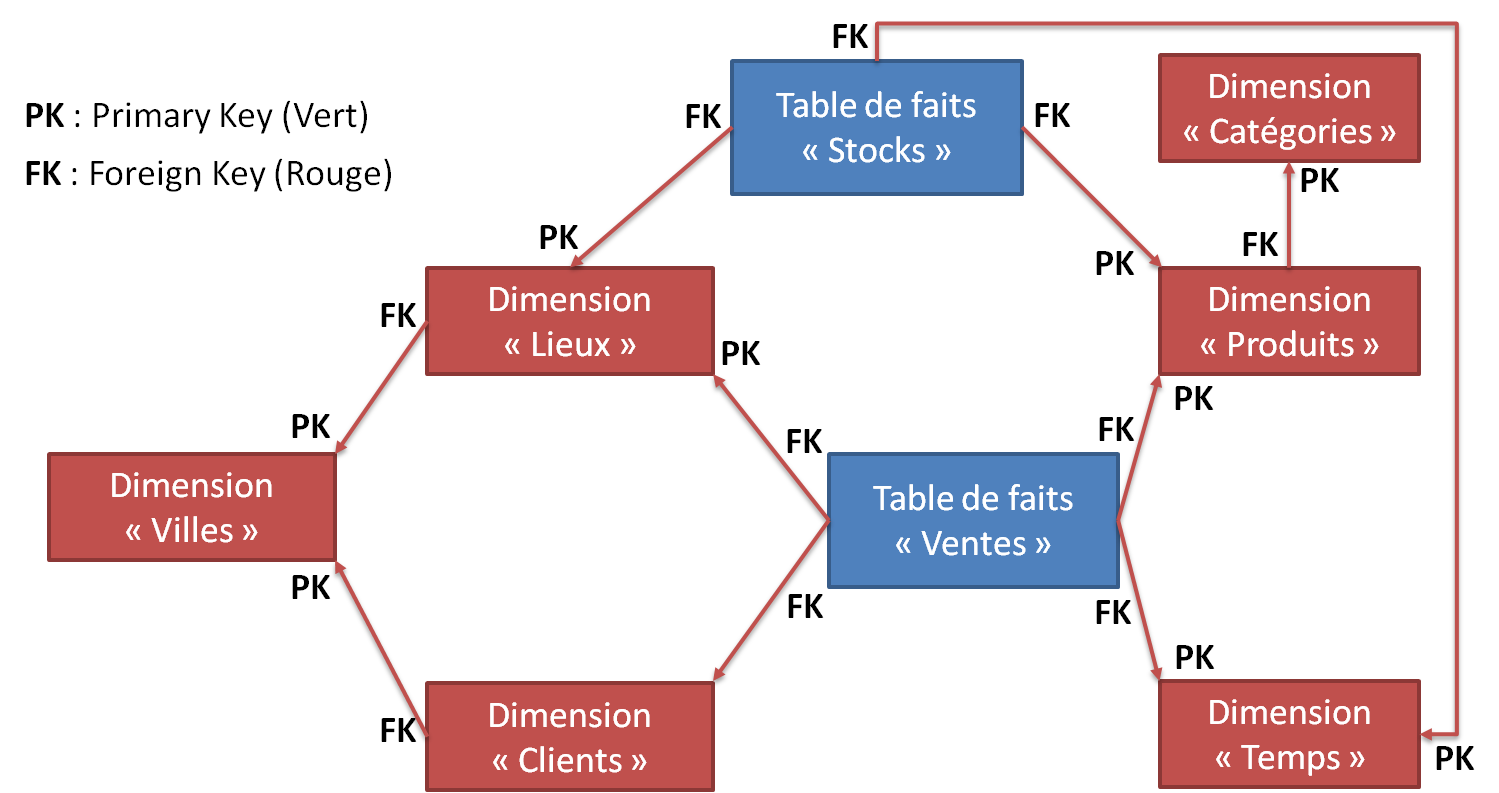
Remplissage

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

# Entrepôt de données

## Vue d’ensemble

Dans la base dédiée « **DataWarehouseODE** », nous créons un schéma dédié « **ODE\_DATAWAREHOUSE** », qui contient les tables suivantes :



## Table de dimension CATEGORIES

But

La table CATEGORIES est une table de dimension qui permet de choisir une catégorie de produit en fonction des univers, famille ou sous-famille des produits. Ces informations servent à aiguiller ou cibler les recherches sur les catégories de produits.

Exemple :  [*Rangement et dressing*](http://www.leroymerlin.fr/v3/p/produits/rangement-dressing-l1308216924)

 > [*Accessoires de rangement*](http://www.leroymerlin.fr/v3/p/produits/rangement-dressing/accessoires-de-rangement-l1400104992)

 > Panier, malle et boite de rangement

/\***BRICE** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| CATEGORIE\_PK | int | Non |
| LIBEL\_UNIVERS | nvarchar(256) | Non |
| ID\_UNIVERS | int | Non |
| LIBEL\_RAYON | nvarchar(256) | Non |
| ID\_RAYON | int | Non |
| LIBEL\_FAMILLE | nvarchar(256) | Non |
| ID\_FAMILLE | int | Non |
| LIBEL\_SSFAMILLE | nvarchar(256) | Non |
| ID\_SSFAMILLE | int | Non |

**Clé primaire** : CATEGORIE\_PK (Clé technique)

**Clé étrangère** : ∅ (C’est une table de dimension en fin de branche du flocon)

La table CATEGORIES contient :

1 clé primaire : **CATEGORIE\_PK** et

4 clés étrangères : ID\_UNIVERS\_FK, ID\_RAYON\_FK, ID\_FAMILLE\_FK, ID\_SSFAMILLE\_FK

**Brève explication des principaux champs**

* Le libellé Univers qui servira à déterminer dans quel univers se retrouve une catégorie .exemple : univers Rangement et dressing
* Le libellé rayon qui sert à déterminer le rayon où retrouver la catégorie de produit.
* Les libellés Famille et sous-familles qui serviront à cibler la recherche .exemple : [*Accessoires de rangement*](http://www.leroymerlin.fr/v3/p/produits/rangement-dressing/accessoires-de-rangement-l1400104992) > Panier, malle et boite de rangement

Ce découpage permettre de déterminer rapidement les catégories des produits donc d’effectuer un ciblage rapide et efficace.

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 : ETL \*/

Remplissage pour les tests

/\***BRICE** : A COMPLETER \*/

Le remplissage de la table a été effectué via un script T-SQL qui a pour objectif d’importer plusieurs fichiers CSV .

Donc il s’agit d’un script basique pour effectuer du loading data .Aucune intelligence particulière à retrouver dans ce script.

## Table de dimension PRODUITS

But

/\***THOMAS** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| PRODUIT\_PK | int | Non |
| CATEGORIE\_FK | int | Non |
| LIBEL\_PRODUIT | nvarchar(256) | Non |
| PRIX\_ACHAT | money | Non |
| TAUX\_TVA | decimal(4, 1) | Non |
| MARQUE\_PRODUIT | nvarchar(256) | Non |
| GROSSISTE\_PRODUIT | nvarchar(256) | Oui |

**Clé primaire** : PRODUIT\_PK (Clé technique)

**Clé étrangère** : CATEGORIE\_FK → CATEGORIES.CATEGORIE\_PK

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***THOMAS** : A COMPLETER \*/

## Table de dimension TEMPS

But

/\***OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| TEMPS\_PK | int | Non |
| DATE | date | Non |
| JOUR | nvarchar(16) | Non |
| ANNEE\_CODE | int | Non |
| ANNEE\_DATE | date | Non |
| ANNEE\_NOM | nvarchar(16) | Non |
| TRIMESTRE\_CODE | int | Non |
| TRIMESTRE\_DATE | date | Non |
| TRIMESTRE\_NOM | nvarchar(16) | Non |
| MOIS\_CODE | int | Non |
| MOIS\_DATE | date | Non |
| MOIS\_NOM | nvarchar(16) | Non |
| SEMAINE\_CODE | int | Non |
| SEMAINE\_DATE | date | Non |
| SEMAINE\_NOM | nvarchar(16) | Non |
| JOUR\_CODE | int | Non |
| JOUR\_DATE | date | Non |
| JOUR\_NOM | nvarchar(16) | Non |

**Clé primaire** : TEMPS\_PK (Clé technique)

**Clé étrangère** : ∅ (C’est une table de dimension en fin de branche du flocon)

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***OLIVIER** : A COMPLETER \*/

## Table de dimension LIEUX

But

/\***THOMAS** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| LIEU\_PK | int | Non |
| VILLE\_FK | int | Non |
| TYPE\_LIEU | char(1) | Non |
| LIBEL\_LIEU | nvarchar(256) | Non |
| DATE\_OUVERTURE | date | Non |
| DATE\_FERMETURE | date | Non |
| SURFACE\_M2 | numeric(6, 1) | Oui |

**Clé primaire** : LIEU\_PK (Clé technique)

**Clé étrangère** : VILLE\_FK → VILLES.VILLE\_PK

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***THOMAS** : A COMPLETER \*/

## Table de dimension CLIENTS

But

/\***CEDRIC** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| CLIENT\_PK | int | Non |
| VILLE\_FK | int | Non |
| TAUX\_REMISE | decimal(6, 2) | Non |
| TYPE\_CLIENT | char(1) | Non |
| NOM\_CLIENT | nvarchar(256) | Non |
| DATE\_NAISSANCE | date | Non |
| DATE\_SOUSCRIPTION | date | Non |
| CODE\_FIDELITE | nvarchar(32) | Oui |

**Clé primaire** : CLIENT\_PK (Clé technique)

**Clé étrangère** : VILLE\_FK → VILLES.VILLE\_PK

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***CEDRIC** : A COMPLETER \*/

## Table de dimension VILLES

But

/\***BERNARD** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| VILLE\_PK | int | Non |
| CODE\_POSTAL | nvarchar(6) | Non |
| CODE\_COMMUNE | int | Non |
| CODE\_REGION | int | Non |
| CODE\_DEPARTEMENT | int | Non |
| CODE\_ARRONDISEMENT | int | Non |
| CODE\_CANTON | int | Non |
| NOM\_VILLE\_MAJ | nvarchar(256) | Non |
| NOM\_VILLE\_MIN | nvarchar(256) | Non |
| POPULATION | int | Oui |

**Clé primaire** : VILLE\_PK (Clé technique)

**Clé étrangère** : ∅ (C’est une table de dimension en fin de branche du flocon)

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***BERNARD** : A COMPLETER \*/

## Table de faits VENTES

But

/\***OLIVIER** : A COMPLETER \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| DATE\_VENTE\_FK | int | Non |
| PRODUIT\_FK | int | Non |
| CLIENT\_FK | int | Non |
| LIEU\_FK | int | Non |
| MONTANT\_HT\_VENTE | money | Non |
| MONTANT\_TVA\_VENTE | money | Non |
| MARGE\_BRUTE | money | Non |
| UNITES\_VENDUES | int | Non |
| NUM\_TICKET | nvarchar(256) | Non |

**Clé primaire** : ∅ (C’est une table de faits)

**Clés étrangères** :

* DATE\_VENTE\_FK → TEMPS.TEMPS\_PK
* PRODUIT\_FK → PRODUITS.PRODUIT\_PK
* CLIENT\_FK → CLIENTS.CLIENT\_PK
* LIEU\_FK → LIEUX.LIEU\_PK

Remplissage via l’ETL

/\* A COMPLETER LORS DU LOT 3 \*/

Remplissage pour les tests

/\***OLIVIER** : A COMPLETER \*/

## Table de faits STOCKS

But

/\* A COMPLETER SI VALORISATION ULTERIEURE \*/

Structure

|  |  |  |
| --- | --- | --- |
| Nom colonne | Type | Nullable ? |
| DATE\_INVENTAIRE\_FK | int | Non |
| PRODUIT\_FK | int | Non |
| LIEU\_FK | int | Non |
| NBR\_DISPO | int | Non |
| NBR\_DEFECTUEUX | int | Non |
| ID\_INVENTAIRE | int | Non |

**Clé primaire** : ∅ (C’est une table de faits)

**Clés étrangères** :

* DATE\_INVENTAIRE\_FK → TEMPS.TEMPS\_PK
* PRODUIT\_FK → PRODUITS.PRODUIT\_PK
* LIEU\_FK → LIEUX.LIEU\_PK

Remplissage via l’ETL

/\* A COMPLETER SI VALORISATION ULTERIEURE \*/

Remplissage pour les tests

/\* A COMPLETER SI VALORISATION ULTERIEURE \*/