DAVID Ruben

LEDOYEN Augustin

Examen du 23 janvier 2017

**Business Intelligence**

*Henallux – 2017/2018 – 3ème IG Groupe A*

Table des matières

[1. Solution proposée 3](#_Toc504477762)

[2. Diagrammes des différents modèles 3](#_Toc504477763)

[2.1 BD opérationnelle 3](#_Toc504477764)

[2.2 BD dimensionnelle 3](#_Toc504477765)

[3. La granularité de la table de faits 3](#_Toc504477766)

[4. Justifications de différents choix 3](#_Toc504477767)

[4.1 Clés primaires 3](#_Toc504477768)

[4.2 Choix de modélisation 4](#_Toc504477769)

[4.2.1 Faits FactsReceiptLine 4](#_Toc504477770)

[4.2.2 DimCustomer 4](#_Toc504477771)

[4.2.3 DimStore 4](#_Toc504477772)

[4.2.4 DimProduct 5](#_Toc504477773)

[4.2.5 DimDate 5](#_Toc504477774)

[5. Limites rencontrées 5](#_Toc504477775)

[6. En cas d’erreurs 5](#_Toc504477776)

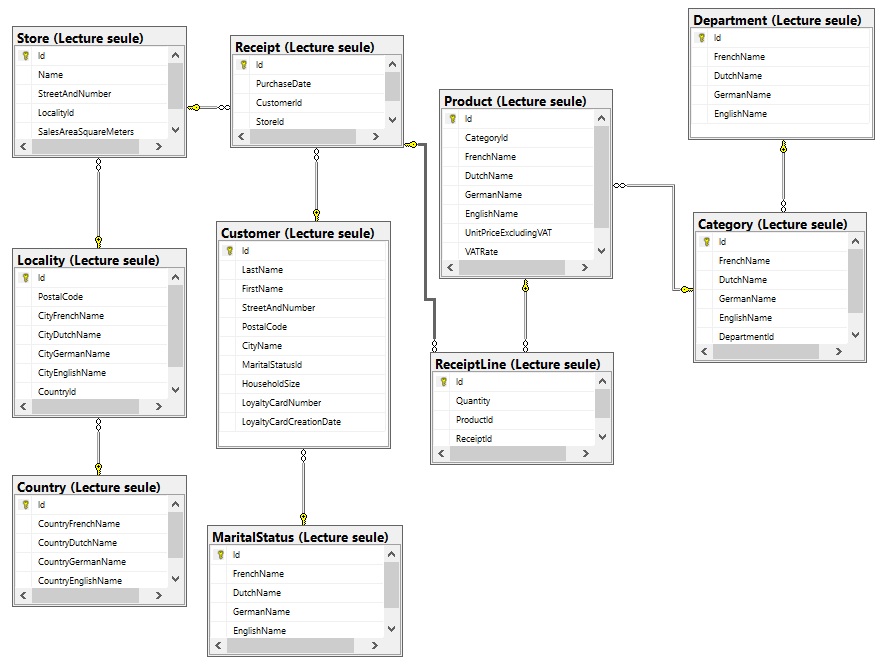
# Solution proposée

Nous avons choisi le schéma en étoile pour la base de données dimensionnelle. Malgré que cela prenne plus de place mémoire, nous pouvons obtenir via ce schéma de meilleures performances car il y a nettement moins de jointures. Sachant que pour ce genre d’installations, l’espace de stockage supplémentaire que cela génère ne coûte presque rien pour la société.

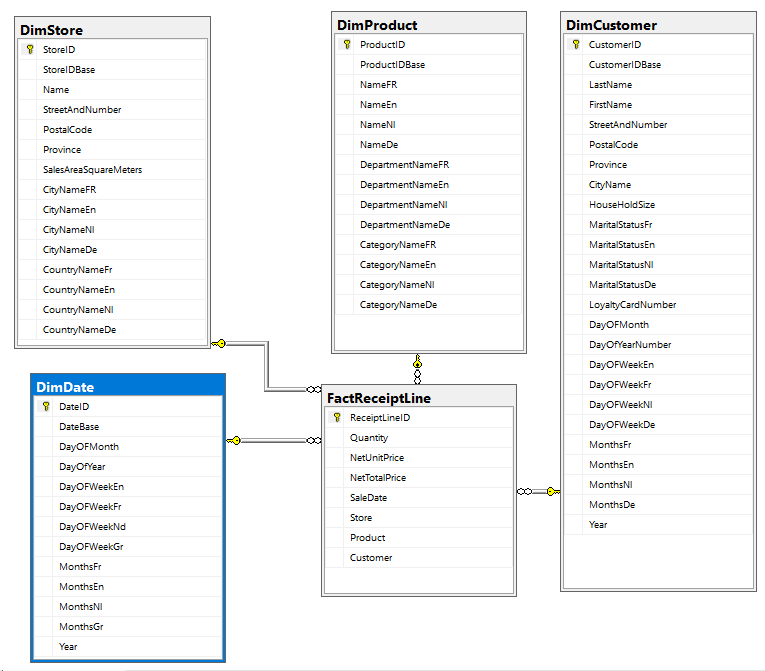
La technologie utilisée est : Sql Server Data Tools

# Diagrammes des différents modèles

## 2.1 BD opérationnelle



## 2.2 BD dimensionnelle



# La granularité de la table de faits

Nous travaillons au niveau d’une ligne d’un ticket de caisse. Cette précision permettant d’offrir la possibilité de creuser et d’obtenir plus de détails.

# Justifications de différents choix

## 4.1 Clés primaires

Ce sont des surrogate key. Ce sont donc des valeurs (entières) auto-incrémentées par la Data Warehouse.

Cela permet, lorsque les données des bases de données de « Grand Souk » et « Alim 2000 » seront mises en commun dans le Data Warehouse, d’éviter d’avoir certaines collisions entre les id de chacune de ces sociétés qui seraient identiques.

Ou bien si la société désire réinitialiser une partie de la base de données tous les x temps, dont notamment les séquences (utilisées pour auto-incrémenter leurs identifiants artificiels), d’éviter une collision entre ces doublons sous les même id venant de la même base de données mais étant différents.

Pour l’index, n’avoir qu’une colonne de type entier est plus performant.

## 4.2 Choix de modélisation

### 4.2.1 Faits FactsReceiptLine

Nous avons donc inclus le calcul de la TVA directement dans le prix unitaire car la notion de TVA (son pourcentage exact) a été retirée. En effet elle n’est pas importante pour une société dans ce genre de cas d’analyses statistiques.

Nous avons également calculé directement le prix total alors qu’un simple calcul suffit en multipliant la quantité par le prix unitaire et ce afin de rendre plus rapide l’utilisation future de cette donnée lorsque l’on aura besoin d’additionner le montant total des ventes selon un certain critère.

### 4.2.2 DimCustomer

Les cityName de customer ne sont qu'en une langue. On a réfléchi à comment éventuellement uniformiser avec les noms de ville en multilingue. Malheureusement ce n'est pas possible car nous n'avons pas dans nos ressources de fichiers comprenant toutes les villes dans les différentes langues (nous n'avons que dans leur langue d'origine).

Devrait-on mettre les infos de dates de création d'une carte de fidélité en clé étrangère dans dimCustomer ou bien répéter le contenu d'une dimDate (au niveau des colonnes) ? Vu que l'espace mémoire ne coûte pratiquement rien, on préfère répéter les colonnes.

La carte de fidélité est créée tous les jours à la même date. On suppose donc que tout est effectué par un logiciel automatiquement pour ajouter les dossiers de la journée. Il n’y a donc pas besoin des infos d’heure-minutes-secondes pour de quelconques statistiques

### 4.2.3 DimStore

Une StoreLine a été créée afin de pouvoir retenir de quelle chaine de supermarché vient les entreprises que l’on encode. Cette donnée sera encodée par la personne qui se charge du transfert d’une base de données afin que tous les magasins de cette dernière soient mis dans une même chaine. Ce choix a été fait en vue de l’intégration futur d’Alim2000 au sein de la même base de données dimensionnelles afin de pouvoir faire des statistiques également en fonction des chaines d’origine des magasins.

### 4.2.4 DimProduct

Pas de débat

### 4.2.5 DimDate

Nous avons décidé d’amener la précision jusqu’aux jours. Nous aurions pu aller jusqu’aux heures si nous jugions être utile de pouvoir analyser les heures de rush ou creuses dans les différents magasins. Mais ce genre d’informations démultiplierait toutes les données dans DimDate sans que cela ne soit nécessaire car aucune question posée par le métier ne demandait une telle précision. Par contre une analyse basée sur les minutes, secondes ou millisecondes nous semblait inutile.

# Limites rencontrées

Il nous est impossible de répondre à la question pour savoir si les personnes nées en Janvier achètent plus de galette des rois.

En effet nous ne disposons pas de l’information nécessaire : la date de naissance d’un client.

Même si nous disposions de cette information (pour aller plus dans le détail même de question), il est impossible de savoir ce à quoi nous devons comparer la quantité de galette des rois achetées. La question ne nous dit pas si on est censé la comparer à la quantité des autres produits achetés par les gens nés en Janvier, à la quantité d’un autre produit en particulier ou encore par rapport aux galettes des rois (ou autre produit) achetés par ceux nés un certain autre mois.

# En cas d’erreurs

Toutes erreurs sortent dans des fichiers nommés selon l’endroit auquel l’erreur est survenue.

Si l’ETL s’arrête on peut le relancer