

**Projet Business Intelligence**

Table des matières

[1 Analyse du projet du client 3](#_Toc504584312)

[1.1 Enoncé de la problématique 3](#_Toc504584313)

[1.2 Solution proposée 3](#_Toc504584314)

[1.3 Technologie utilisée et composants impliqués 3](#_Toc504584315)

[2 Le diagramme de la BD décisionnelle 5](#_Toc504584316)

[3 Granularité de la table de faits et justification des choix 6](#_Toc504584317)

[4 Les limites rencontrées 6](#_Toc504584318)

[5 Stratégie d’exécution 7](#_Toc504584319)

# Analyse du projet du client

## Enoncé de la problématique

Un gros groupe industriel qui possède plusieurs chaines de supermarchés souhaiterait pouvoir analyser les ventes selon certains critères. Tous les produits vendus par ces supermarchés sont stockés dans une base de données grâce aux systèmes des caisses enregistreuses.

Le client nous donne une liste d’exemple de questions qu’il aimerait pourvoir répondre grâce à notre outil :

* De quelle province viennent les clients qui représentent le plus grand total d’achat (en termes de montant)
* Quelle chaîne de supermarchés vend le plus de produits d’une catégorie X
* Le nom du mois de l’année au cours duquel les ventes sont les plus importantes (toujours en termes de montants)
* Le total de la surface de vente disponible dans chaque province du pays.
* Quelles sont les catégories de produits qui sont les plus achetées par les célibataires ?
* Les montants vendus ventilés par département.
* Les personnes nées en Janvier achètent-elle plus de galette des rois ?

Ils souhaitent que le multilinguisme soit supporté.

## Solution proposée

Tout d’abord, nous avons identifié les besoins du métier. Une fois ces besoins identifiés, nous avons identifié les données de la BD opérationnelle qui permettait d’y répondre.

Ensuite nous avons mis en place notre *data wharehouse* et nous avons créé un *ETL* qui nous a permis de l’alimenter. Ensuite, normalement, nous devions créer *OLAP* pour exploiter le *data wharehouse* et *OLAP* a alimenté le cube ! Mais par manque de temps, nous n’avons pas réalisé cette partie.

## Technologie utilisée et composants impliqués

Nous avons utilisé plusieurs outils proposés par *Visual studio* pour créer notre *ETL.*

Tout d’abord, nous avons dû créer plusieurs dimensions, chacune d’elle est représentée par des « taches de flux de données ». Une fois dans cette tâche, nous avons importé les données de la BD opérationnelle(*OLTP*) mise à disposition. Ensuite, nous avons dû trier les données obtenues à l’aide d’un « sort », d’un coté sur la clé étrangère d’une table et de l’autre sur la clé primaire correspondante. Grace aux tris, nous avons pu effectuer un « *merge join* » et ainsi créer une nouvelle table avec comme information le contenu des deux tables jointes. Après, nous avons dû convertir le type des données grâce à l’outil, « *converting data* », car pour les chaines de caractère, par exemple, le format n’est pas le même dans la BD opérationnelle que dans la BD décisionnelle. Une fois les données converties, nous avons fait des colonne dérivées grâce au « *Derived column* » cette dernière opération nous permet de par exemple dans la dimension date à partir du fichier plat, de créer les colonnes nécessaires, c’est également un écran de qualité. Nous trouvons aussi des « *lock up* », ces derniers sont utilisés pour chercher la *surogate-key* grâce à l’identifiant source et créer une nouvelle colonne avec la valeur de la *surogate-key*. Nous utilisons également le « *coditional split* » qui va nous permettre d’écarter un enregistrement sous certaines conditions, c’est aussi notre écran de règle métier. Pour finir, nous trouvons la source où l’on doit envoyer toutes les informations (dans notre *data wharehouse*). C’est également à cet endroit que nous devons effectuer le « *mappage* », c’est-à-dire dire que la colonne qui se trouve dans notre *data wharehouse* va devoir prendre telle valeur.

# Le diagramme de la BD décisionnelle



# Granularité de la table de faits et justification des choix

Pour la table de faits, nous avons travaillé avec la granularité la plus fine(*drill-down*) c’est-à-dire, un fait est égal à une ligne de commande.

Pour les clés primaires, nous avons décidé de ne pas prendre les clés qui se trouvent dans la BD source mais de nouvelles clés dans le but d’éviter les conflits si dans l’avenir nous perdrions une autre BD source en plus. Cependant, nous avons décidé de garder les id qui se trouvent dans la BD source pour faire du « *survivorship* », pour nous permettre de gérer les soucis liés aux doublons

Pour que la compréhension du *data wharehouse* soit le plus simple possible pour le métier et pour une question de performance accrue, nous avons décidé de modéliser nos dimensions et nos faits par le biais du « *star scheme*».

# Les limites rencontrées

Il y a deux questions posées par le métier auxquelles nous ne pouvons pas répondre :

* Quelle chaîne de supermarchés vend le plus de produits d’une catégorie X ?

Nous ne possédons pas l’information sur quel magasin appartient à quelle chaine de supermarché.

* Les personnes nées en janvier achètent-elles plus de galette des rois ?

Nous ne pouvons y répondre tout simplement car il n’y a aucune information sur la date de naissance des clients dans la BD opérationnelle.

# Stratégie d’exécution

A chaque exécution de l’*ETL* le contenu des dimensions et du fait vont être supprimés et l’*ETL* va refaire le travail à chaque fois. Donc dans le cas où l’*ETL* s’arrête subitement, ce n’est pas un problème.

En ce qui concerne les erreurs, elles sont gérées de façon différentes grâce aux écrans de qualité. Nous avons rencontré deux erreurs durant le développent de l’*ETL*.

Tout d’abord, nous avons remarqué que le code postal de certaines provinces n’était pas présent et donc égale à NULL, nous avons donc décidé d’utiliser une colonne dérivée pour que lorsqu’il rencontre un code postal a NULL il indique à la place « province inconnue ».

Ensuite, dans la BD opérationnelle il y avait dans certaines commandes des quantités égales à 0, nous avons décidé d’écarter ces enregistrements et de les mettre dans un fichier plat. Ce n’est qu’en discutant avec le métier que nous pouvons savoir ce que nous devons faire de ces enregistrements.