Haute Ecole de Namur - Liège - Luxembourg
Département économique
Implantation IESN

Bachelier en Informatique de Gestion

Bloc 3

Business Intelligence

Projet : Démarche BI

Par Lempereur Pierre-Henry et Fricot Damien

Samuel Scholtes

Année académique 2018-2019

Table des matières

Analyse des besoins métiers et de la solution 3
Description du domaine d'application 3
Analyse des besoins métiers 3
Analyse de la base de données opérationnelle 4
Diagramme de la base de donnée opérationnelle 5
Conception du DataWarehouse 6
Conception de la base de donnée multidimensionnelle 6
Diagramme du DataWarehouse 7
Alimentation du DataWarehouse 8
Alimentation des dimensions 8
Alimentation de fact 10
Gestion des erreurs 11
Limites rencontrées 11

Analyse des besoins métiers et de la solution

Description du domaine d'application

Vous êtes un consultant BI envoyé en mission chez un client, un gros groupe industriel, qui possède plusieurs chaînes de supermarchés, dont les chaînes « Grand Souk» et « Alim
2000 ».

Ces chaînes de supermarché enregistrent tous les produits vendus grâce à des systèmes de caisses enregistreuses. Le groupe souhaite pouvoir tirer de l'information de toutes les données enregistrées par les caisses. Il souhaiterait disposer d'un outil simple à utiliser qui permettrait d'obtenir des réponses rapidement à des questions précises. Il souhaiterait pouvoir analyser les données sur plusieurs axes, faire des recoupements. Pour le moment, le groupe s'intéresse particulièrement aux données produites par Grand Souk. Celles d'Alim
2000 suivront dans un futur proche. Votre système doit être conçu de manière à absorber facilement l'intégration d'Alim 2000 dans le futur.

Le groupe souhaiterait pouvoir analyser les ventes par critère temporel, par produit et catégorie et par les caractéristiques des clients qui ont acheté des produits.

Analyse des besoins métier

Le besoin métier recherché à travers ce domaine d'application est une analyse des ventes de la société, dont dans un premier temps la chaine « Grand Souk ». Les différents axes seront :

- Par critère temporel
- Par produit et catégories
- Par les caractéristiques des clients qui ont acheté

Dans un avenir proche la société souhaiterait pouvoir ajouter les informations d'autres chaines de supermarché comme la chaine « Alim2000 ».

Analyse de la base de données opérationnelle

La base de données opérationnelle fournie est nommée « BiProject_OLTP » et se situe sur le serveur « vm-sml2.iesn.be/Stu3IG ». Elle se représente comme ci-dessous à l'image 1 :

Pour notre analyse des ventes nous allons nous concentrer sur les tickets de caisse, ce qui représentera notre granularité.

Dans le cas de l'analyse des ventes, les tables qui seront à exploitées sont :

1. Par produits et catégories :

   - PRODUCT
   - CATEGORY
   - DEPARTMENT

   Dimension Product (DimProduct)

   - STORE
   - LOCALITY
   - COUNTRY

   Dimension Store (DimStore)

   La dimension Store nous permettra de localiser géographiquement les produits vendus.

2. Par les caractéristiques des clients :

   - CUSTOMER
   - MARITAL_STATUT

   Dimension Customer (DimCustomer)

3. Par critère temporel :

   Nous nous baserons sur un script SQL qui dans un premier temps va générer un ensemble de dates brutes puis remplir la table avec ces informations ainsi que les caractéristiques correspondantes (nom du jour et du mois dans les 4 langues).[1]

   Dimension Date (DimDate)

Et pour notre table des faits : RECEIPT et RECEIPT_LINE **Fact Receipt (FactReceiptLine)**

[1] Script d'origine provenant de « https://www.codeproject.com/Articles/647950/Create-and-Populate-Date-Dimension-for-Data-Wareho » consulté le 19 Janvier 2019. Ce choix est différent de celui du cours car il nous a semblé plus simple dans la solution. De plus nous avons préféré jouer la sécurité, étant officiellement inscrit en 2em IG notre maitrise du C# est loin d'être parfaite au moment du projet.

Diagramme de la base de donnée opérationnelle

Image 1- Diagramme Relationnelle "BiProject_OLTP"

Conception du DataWarehouse

Conception de la base de donnée multidimensionnelle

- Pour la création de la base de données multidimensionnelle nous utiliserons l'approche de schéma en étoile (Star scheme). Avec cette approche on dénormalise ce qui induit des attributs répétés. Les avantages de cette approche sont :

  - une diminution des jointures => une augmentation des performances (requêtes plus rapides)

  - un modèle plus simple à comprendre

- Nous remplacerons les attributs NULL de la base de données opérationnelle par un attribut définit dans la modélisation Multidimensionnelle pour la table des faits et l'exploitation des données du fichier Excel des codes postaux de Belgique.
  - Pour une chaine de caractère : « INCONNU »
  - Pour une date : « 1100-01-01 00:00:01.10000000 »
  - Pour un entier : « -1 »

- Nous préférerons l'utilisation de Surrogates key comme identifiants des nouvelles tables et sauvegarderons donc les identifiants d'origine dans un attribut « OriginalId ». Ceci dans le but d'éviter d'éventuels conflits d'identifiants après les opérations de fusion.

- Dans l'optique d'étendre notre DataWarehouse à d'autres chaines de magasin on enregistrera le nom de la chaine de magasin étudiée dans l'attribut « BrandName »

Le script SQL de création de la base de données en question est placé dans le dossier src. Dans ce script on précisera un auto-incrément de l'identifiant (id) de chaque table. Le DataWarehouse correspondant est créé sur le serveur « vm-sml2.iesn.be/Stu3IG » et nommé « 1819_etu35597_BD ».

Les technologies utilisées dans le cadre de la conception du DataWarehouse sont :

- « SQL server 2017 Developper » pour la visualisation et création de la structure de notre DataWarehouse via les instructions SQL. Il nous est également utile pour visualiser notre base de données opérationnelle.
- « SQL server data Tools » sur Visual Studio avec la création d'un projet de type « Business Intelligence – Intégration Services Project ». C'est à partir de ce projet que nous créerons la démarche à suivre pour le remplissage des tables du DataWarehouse.

Diagramme du DataWarehouse

Nous modéliserons notre modèle multidimensionnelle comme ci-dessous à l'image 2 :

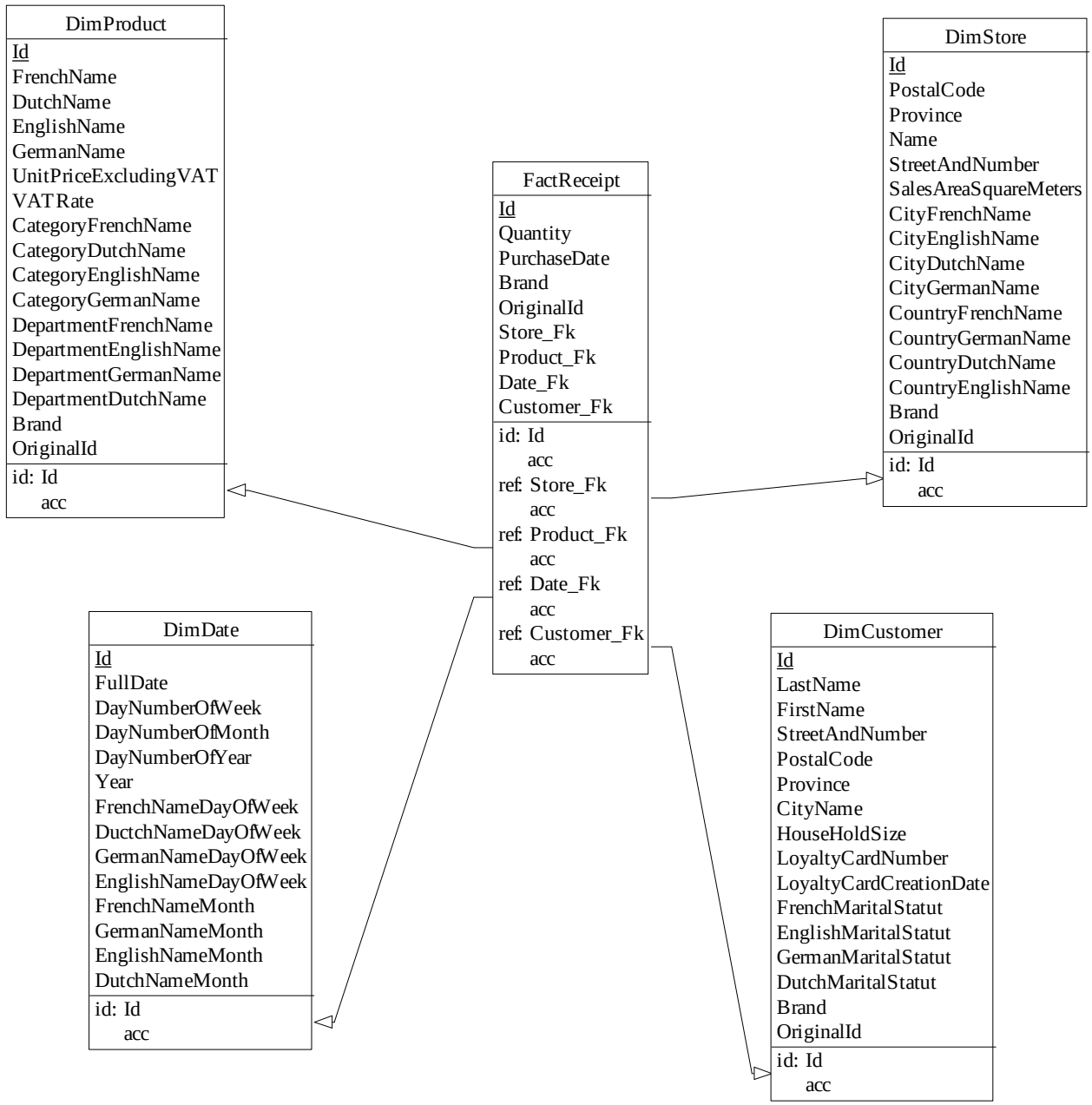
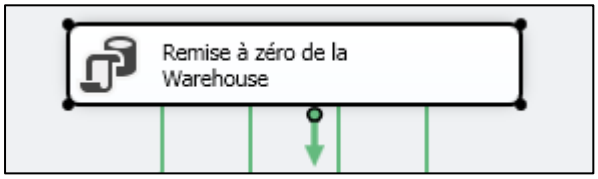


Image 2 - Diagramme Multidimensionnelle

Alimentation du DataWarehouse

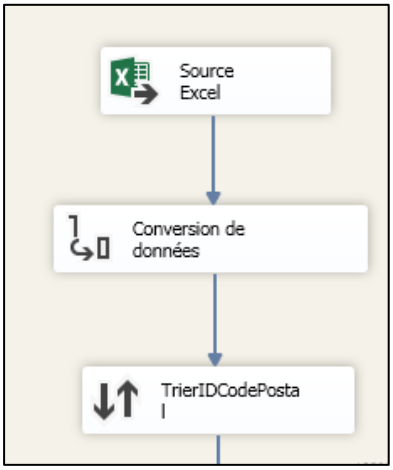
Alimentation des dimensions

1. Comme précisé dans les consignes on commence par vider entièrement les tables du DataWarehouse à l'aide de l'instruction SQL « Delete from [TABLE] ».



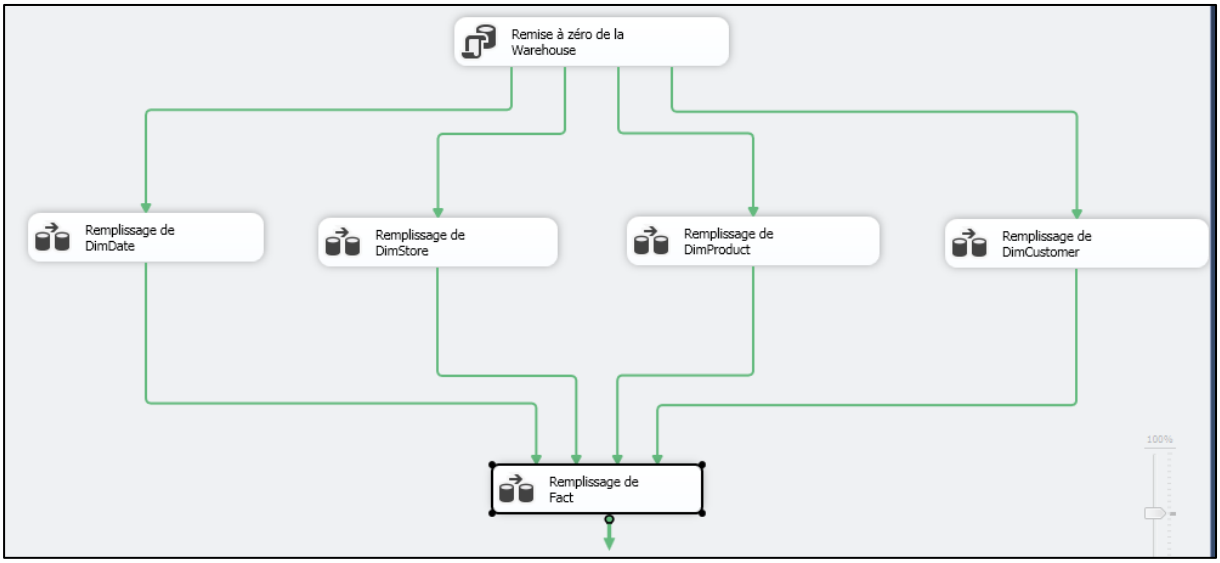
2. Ensuite, pour chaque table de dimension du DW on va commencer par exploiter les données des tables de la base opérationnelle comme explicité dans l'analyse. Afin de fusionner ces tables en une seule table du DW on doit les trier en conséquence sur un même attribut. Le choix de cet attribut n'est pas anodin car il va déterminer quel identifiant des deux tables sera conservé comme identifiant à la sortie de la fusion. Précisons également qu'on procède la fusion avec une jointure externe gauche pour ne perdre aucune donnée.

3. Pour DimStore et DimCustomer nous avons besoin d'un fichier des codes Postaux de Belgique. Ce dernier au format Excel provient du site : « http://www.bpost.be/site/fr/envoyer/adressage/rechercher-un-code-postal » consulté le 19 Janvier 2019. Pour l'utilisation de ce dernier on a besoin d'utiliser une conversion de donnée pour exploiter ces dernières.



4. Avant chaque remplissage des tables de dimension nous utiliserons une fonction de colonne dérivée afin d'ajouter dans «BrandName » le nom de la chaine de magasin dont proviennent les données. Dans notre cas « Grand Souk ». Cette opération a pour but de donner une piste pour l'intégration de données provenant de plusieurs chaines de magasin.

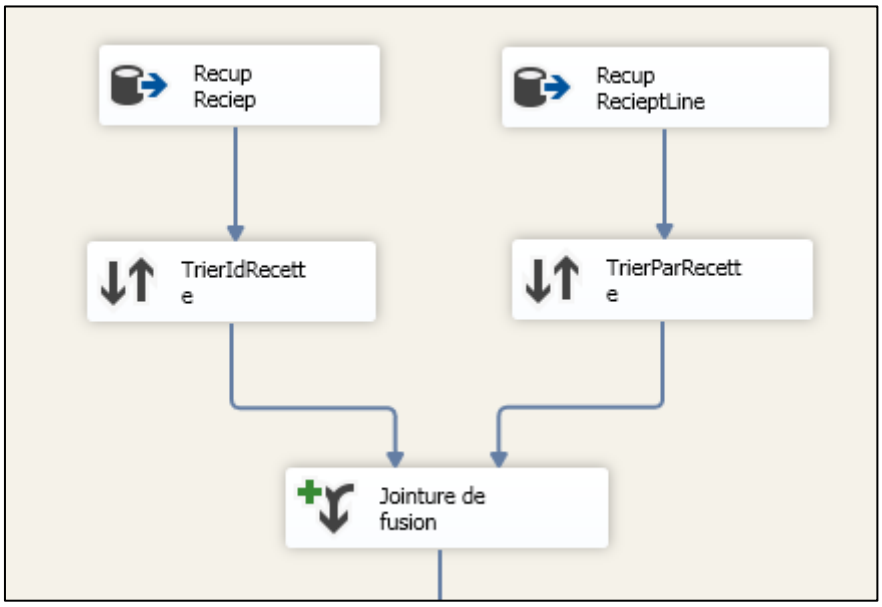
5. On termine le remplissage des tables de dimensions en vérifiant le mappage entre la table résultant des différentes fusions et la table de dimension. C'est d'ailleurs à ce moment qu'on vérifie que l'identifiant depuis les tables opérationnelles est placé dans l'attribut « OriginalId » de notre dimension.



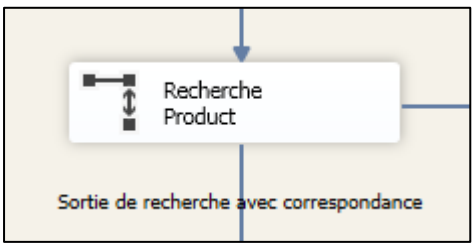
Remarque : Dans l'utilisation des données provenant du fichier Excel des codes postaux de Belgique nous avons été confrontés à plusieurs erreurs. La plus importante est liée à l'identification de chaque ligne du fichier. En effet pour identifier une localité le code postal de cette dernière n'est pas suffisant. En prenant cet enregistrement comme identifiant on se retrouve avec de nombreux enregistrements du DataWarehouse totalement identiques. On a donc besoin d'une clé primaire composée du code postal et du nom de la localité. Cependant pour pallier aux difficultés techniques dues aux différents types de données et de fusion nous avons imaginé une solution temporaire. En effet nous avons décidé de rajouter une étape de tri avant l'alimentation de la dimension. Dans ce module, on trie les enregistrements par tous leurs attributs et on coche l'option supplémentaire de suppression des données en doubles. Nous sommes conscient que cette solution n'est que temporaire et qu'il est nécessaire de retravailler l'exploitation de ces données.

Alimentation de fact

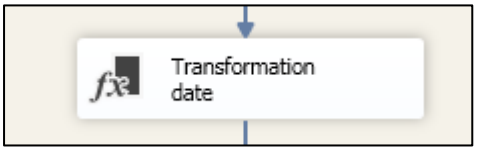
- Pour le remplissage de la table des faits, on va commencer par extraire les données des tables « Receip » et « ReceiptLine ». Puis comme pour les dimensions, les trier et les fusionner.



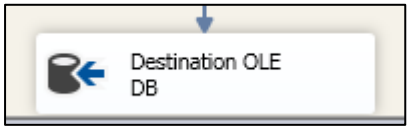
- Ensuite on va rechercher dans les dimensions, précédemment alimentées, les informations correspondantes en lien avec nos clés étrangères. Ces étapes nommées « RechecheProduct », « RechercheDate », « RechercheStore » et « RechercheCustomer » vont nous permettre de finalement alimentée notre table des faits avec les clés étrangères correspondantes aux dimensions.



- Précisons tout de même que nous avons besoin d'exécuter une transformation du type de date et d'ajouter le brand « Grand Souk » comme dans l'alimentation des dimensions.



- Finalement on va mapper les données pur remplir notre table des Faits.

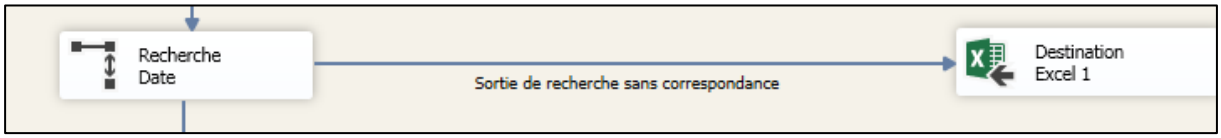


Gestion des erreurs

1. Sortie de recherche sans correspondance

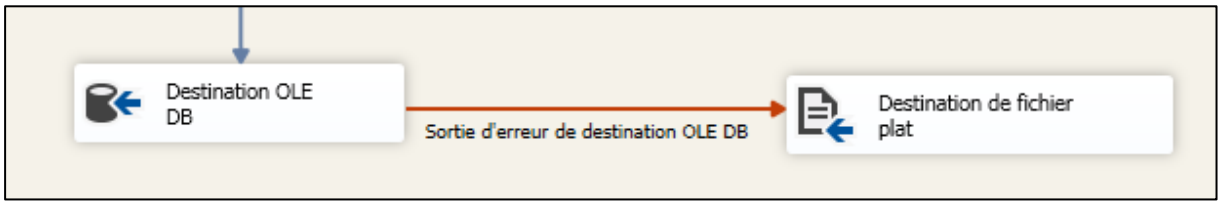
Dans l'alimentation de la table des faits, lors de la recherche de correspondances des clés étrangères avec les dimensions du DataWarehouse il est fréquent d'avoir des résultats de recherches sans correspondance. Par exemple dans la table opérationnelle « Customer » on retrouve dans l'attribut « City Name » la ville de « Gotham », or cette dernière n'existe pas en Belgique et n'est donc pas dans le fichier Excel des codes Postaux.

Pour malgré tout garder une trace de ces enregistrements on place une sortie de Destination vers un fichier Excel. Ce fichier Excel « FichierExcelErreur » reprendra une feuille par recherche et dans chaque feuille on y retrouvera les différents éléments de sortie de recherche sans correspondance.



2. Sortie d'erreur de destination

Dans le cas d'une erreur lors de l'exécution de l'ETL, on prévoit un fichier texte classique de sortie de destination. Si lors de l'exécution une erreur survient, alors le code d'erreur retourné par le système est enregistré pour une consultation facilitée.



Limites rencontrées

Les limites aux questions possibles pour le système sont d'une part dépendantes des informations héritées de la base de donnée opérationnelle et d'autre part de notre choix de représentation des dimensions ainsi que de la table des faits.

Par exemple on ne retient pas les dates de naissance des clients donc on ne peut pas répondre à une question du style « Les personnes nées en Janvier achètent-elle plus de galette des rois ? ».