

Rapport

Introduction

Dans le cadre du module de Business Intelligence, il nous a été demandé de réaliser une analyse sur l'activité de préparation de commande de la société "Northwind". Pour réaliser cette dernière, nous devons mettre en pratique tous ce qui à été vu en cours. Tant en TD / CM qu'en TP.

La première partie du travail consistait à identifier le besoin de l'entreprise. Pour ce faire nous avons employés les notions vu en cours et particulièrement celle étudiées dans le chapitre 3 : "Modélisation" afin de réaliser un modèle en étoile.

La seconde partie du projet consistait à utiliser l'ETL "Talend" étudié durant les premières séances de TP.

L'utilisation de ce logiciel avait pour but de nettoyer les fichiers que l'on avait en entrée (fichier CSV) afin d'en intégrer les données (nettoyées et cohérentes) dans une base de donnée crée au préalable avec MySQL.

Enfin, pour la dernière partie nous devons utiliser le logiciel de data-visualisation "QlikView" également vu en cours. Ce dernier permet de réaliser différents graphiques afin de mettre en avant clairement et facilement un fait, une tendance.

Ce rapport fait office de résumé du travail que nous avons fournis afin que notre projet aboutisse.

1) Identifier le besoin

1) Identifier les tables utiles, les indicateurs clés, les dimensions

Table les plus utiles : Customers, Employees, Orders, Order details, Products, Categories.

L'ensemble des tables de notre système nous ont semblé importante afin de réaliser le travail de demander à l'exception d'une.

Nous avons décidé de ne pas garder la table "Suppliers" (fournisseur) que nous jugeons inutile dans notre modèle. En effet, les analyses qui nous sont demandées n'ont à aucun moment besoin de cette table.

Tableau indicateurs / dimensions :

Dimensions ou axes⇒	Temps	Client	Produits	Commandes		Employé	Categories
Indicateurs v				Commande	Détails_co mmande		
Familles_produits_rentables	X		X		X		X
Employés qui gèrent le plus gros CA	X			X	X	X	
Clients les plus fidèles	X	X		X	X		
Remises_accordées	X	X	X	X	X		

Nous avons placés en dimensions l'ensemble des tables que nous jugeons utiles. Nous avons également ajouté la dimension "Temps" qui est essentielle dans un modèle en étoile.

Explications des relations indicateurs / Dimensions:

Nous tenons à expliquer les relations entre les différents indicateurs et dimensions que nous pouvons observer dans le tableau ci dessous (cf. Tableau indicateurs/dimensions)

La dimension "Temps" est nécessaire pour l'ensemble des indicateurs. En effet, nous devons toujours exprimer ces indicateurs pour un temps donné autrement les indicateurs n'auront pas grande signification. (Nous ne parlerons pas de la dimension temps dans le tableau ci-dessous pour éviter les répétitions)

Indicateur	explications des relations
Familles_produits_rentables	<p>Pour obtenir les familles des produits les plus rentables, nous avons besoin de la dimension "Details_commande". Cette dimension possède les attributs "UnitPrice" et "Quantity" dont nous avons besoins afin de connaître la rentabilité d'un produit. Cette dimension fait également office de lien entre les tables puisque elle possède des clés étrangères. Ici, celle qui nous intéresse est "ProductID" qui nous permet de savoir quel produit nous avons dans chaque commande. Et donc de connaître la rentabilité pour un produit donné à un moment donné (avec la dimension temps). La dimension "Produits" va permettre de connaître le nom du produit (et pas seulement l'id qui est peu voir pas du tout parlant).</p> <p>Enfin, nous avons besoin de la dimension "Catégorie" pour connaître la catégorie et donc la famille auquel appartient le produit.</p>
Employés qui gèrent le plus gros CA	<p>Afin de connaître les employés gérant le plus gros CA, nous avons besoin tout d'abord de la dimension "Employé". Les attributs "FirstName" et "LastName" vont permettrent de récupérer le nom et le prénom de l'employé.</p> <p>La dimension "Commande" permet pour une commande donnée de connaître l'employé qui s'en charge avec l'attribut "EmployeeID". On va alors pouvoir associer le nom et le prénom de la dimension Employé avec l'EmployeeID de la dimension Commande.</p> <p>Enfin la dimension "Details_commande" permettra de connaître le montant total d'une commande en multipliant tous les prix des produits (UnitPrice) par la quantité de tous les produits (Quantity). Ce montant total est enfaite le chiffre d'affaire.</p>
Clients les plus fidèles	<p>Afin de connaître les clients les plus fidèles, nous avons besoin tout d'abord de la dimension "Client". Son attribut "CompanyName" permet de connaître le nom du client.</p> <p>La dimension "Commande" permet pour une commande donnée de connaître le client qui passe cette dernière (avec l'attribut CustomerID). A partir de CustomersID, nous pouvons retrouver le nom du client (ayant la même ID).</p> <p>Enfin, la dimension "Détails_commande" permet de connaître le montant de la commande (voir ci-dessus "Employés qui gèrent le plus gros CA").</p> <p>Nous pouvons interpréter la fidélité de deux manières différentes.</p> <p>Un client peut être considéré comme "fidèle" si il passe des commandes régulièrement.</p> <p>L'autre manière de voir les choses est la suivante : un client peut être considéré comme "fidèle" si les montants des commandes qu'il a passé sont importants. C'est la seconde méthode que nous avons choisi de mettre en place.</p>
Remises_accordées	<p>Afin de connaître les remises accordées à chaque client, nous avons besoin tout d'abord de la dimension "Client". Cette dimension va permettre de connaître le nom du client ayant le droit à la remise.</p> <p>La remise (en pourcentage) est disponible dans la dimension "Détails_commande". Cette remise est associé à un produit (ProductID) lors d'une commande (OrderID).</p> <p>La dimension produit va permettre de récupérer le nom du produit (ProductName) en fonction de son ID dans la commande.</p> <p>Enfin la dimension "Commande" va faire le lien entre la dimension Client et Commande en récupérant l'id du client qui passe la commande.</p>

2) Le modèle en étoile

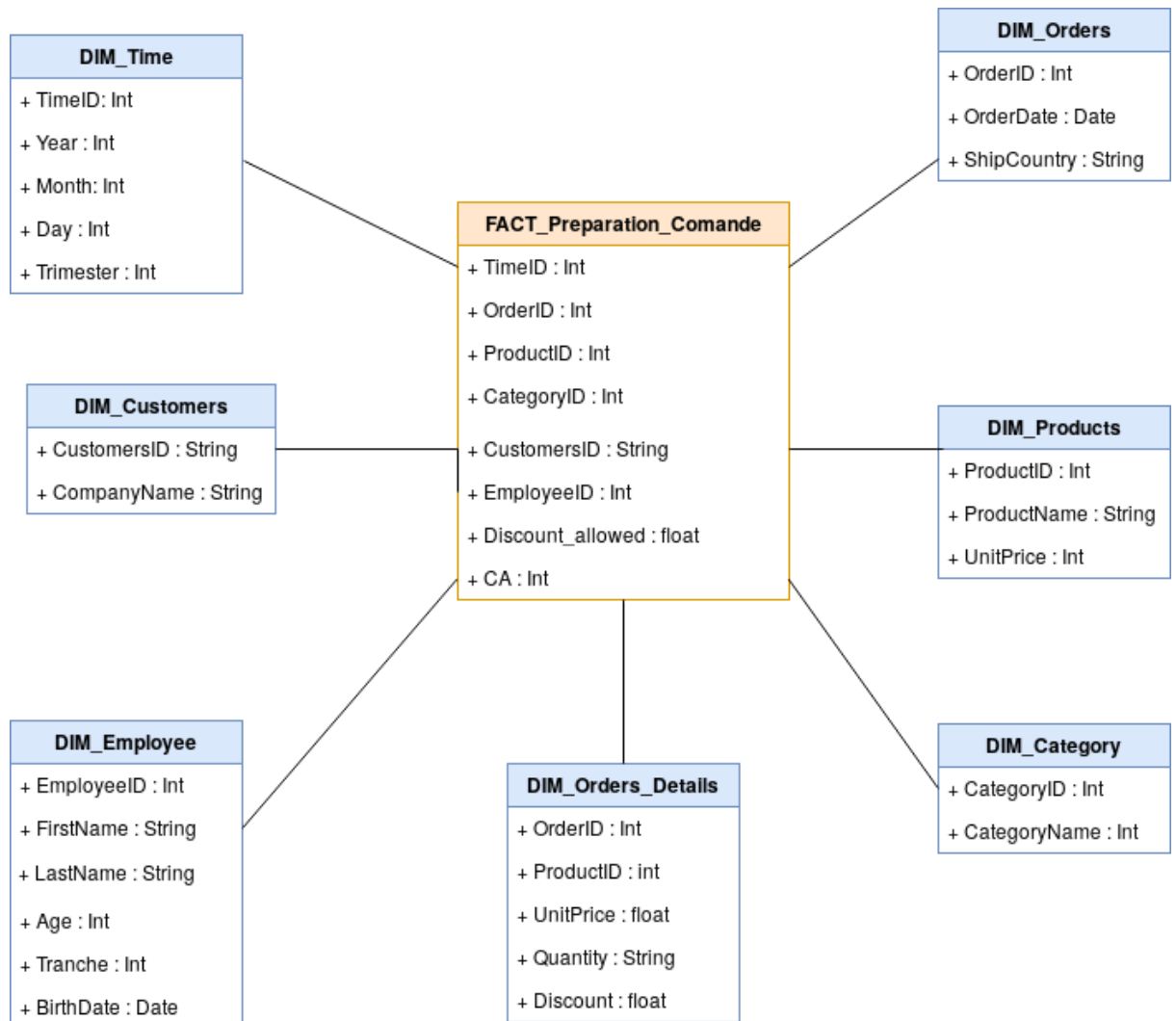


image 1.1 : Modèle en étoile

Nous avons gardé comme attributs uniquement ceux qui nous paraissait nécessaire pour l'analyse que nous devons faire.

- Est-il selon vous intéressant de garder le numéro de commande dans la table de faits?
Pour quel besoin?

Oui car c'est ce numéro de commande qui nous permet de faire la "relation" ou encore "jointure" du produit au employés/clients afin de savoir les informations demandées par les familles de produits les plus rentables

2) Développement Talend

Talend Open studio est un ETL (Extract Transform Load) open source ce qui en fait un avantage au vu de son coût.

Nous avons à notre disposition 7 fichiers de type CSV (représentant des données tabulaires sous forme de valeurs séparées par des virgules). Ces fichiers contiennent toutes les données de l'entreprise (commandes, clients, produits etc.)

Le premier travail consistait à créer une base de données sous MySQL conforme à notre modèle en étoile.

Nous devons dans un premier temps avec Talend effectuer la connexion avec cette base de données de la même manière que nous l'avons vu en cours de TP.

Une fois ce travail effectué, nous devons récupérer tous les fichiers CSV que nous jugions utiles (tous sauf suppliers.csv).

Nous insérons chaque fichier CSV dans un Job sous forme de TFileInputDelimited de type CSV.

Chaque TFileInputDelimited va être individuellement relié à un TMap. Le TMap va permettre de faire le tri des attributs que l'on souhaite garder dans notre fichier de sortie. Une fois que l'on a fait cette étape, il nous reste à relier le TMap à notre fichier de sortie, ici nous avons nos différentes tables de notre base de données.

De cette manière, nos tables seront alimentées par les données que nous souhaitons.

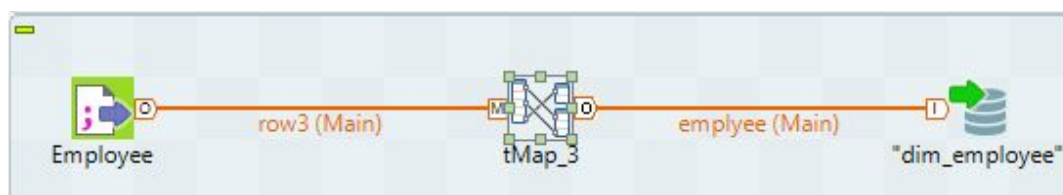


image 2.1 : récupération, triages et insertions des données "Employee"

Certains fichiers CSV sont erronés. En effet, dans le fichier "Employee" par exemple, EmployeeID est de type Int. Or dans certaines lignes les données de cet attribut étaient des String. Il a donc fallu ne pas prendre en compte ces données erronées.

Un autre problème nous a ralenti dans notre avancée : le fait que les attributs "UnitPrice" et "Discount" de la table "Order_Details" étaient des String en entrée et que nous voulions en sortie des doubles. Pour palier à ce problème, nous avons utilisé l'expression suivante (ici pour UnitPrice): `Double.parseDouble(row2.UnitPrice.replaceAll(",", "."))`.

La méthode "**Double.parseDouble()**" permet de transformer un String en Double. Le problème est que les doubles ne peuvent être séparés par une ",". C'est pour cette raison que nous utilisons la méthode suivante : `replaceAll(",", ".")`. Cette méthode va permettre de transformer les séparateurs de décimales "," en "." qui lui est compris pour le type double.

Il nous a été demandé de calculer l'âge de chaque employé afin qu'il apparaisse comme attribut dans la table "Dim_Employee". Pour ce faire, nous devons ajouter un attribut "âge" dans la table "Dim_Employee". Ensuite, dans le TMap nous devons créer une expression afin que l'attribut âge renvoie l'âge des employés.

Cette expression est la suivante :

```
(1998- Mathematical.INT(TalendDate.formatDate("yyyy",row3.BirthDate)) )
```

Dans un premier temps nous avons mis la date current mais au vu des résultats que ça nous renvoyait nous avons mis comme date 1998 conformément à l'intitulé du projet.

Cette expression est simple, il suffit de soustraire à l'année 1998, l'année de la date d'anniversaire des employés (BirthDate).

Ensuite nous devons faire une expression afin de faire appartenir chaque employé à une tranche d'âge. Les tranches d'âges sont les suivantes :

- 1 : Moins de 30 ans
- 2 : De 30 ans à 45 ans inclus
- 3 : Plus de 45 ans

Voici l'expression que nous avons réalisée afin de répondre à cette demande :

```
((1998- Mathematical.INT(TalendDate.formatDate("yyyy",row3.BirthDate))) < 30) ? 1 :  
((1998- Mathematical.INT(TalendDate.formatDate("yyyy",row3.BirthDate))) > 45) ? 3 : 2
```

Si l'âge de l'employé est inférieur à 30 alors il est dans la catégorie 1, si l'âge de l'employé est supérieur à 45 alors il est dans la catégorie 3 sinon il est dans la catégorie 2.

D'après nos résultats, aucuns employés sont dans la tranche d'âge 1 (moins de 30ans).

3) Optimisation

Pour optimiser les performances vous devez bâtir une table d'agrégat qui contient la quantité et le montant vendu par mois, trimestre, année, employé, et pays de la vente. Créer la table et l'alimenter dans Talend à partir des faits et des dimensions.

Nous avons donc créé une table aggregate qui nous sera utile pour les graphiques sur QlikView. Nous l'avons alimenté sur Talend dans le job aggregate_table à l'aide de la table Orders, Orders_details et Employee. Nous calculons le mois, année trimestre en fonction de la date de commande (OrderDate) et des fonctions disponibles dans Talend (getMonth(), getYear()) et de différentes conditions pour le trimestre. Il nous faut ensuite utiliser un tAggregateRow afin de réaliser l'agrégat (group by) ainsi que la somme des montants pour avoir le CA dépendant de différentes dimensions.

4) Développement Qlikview

- Charger dans Qlikview les tables utiles du schéma Northwind (le code doit être commenté)

Du fait que nous avons créé une table `aggregate_table` dans Talend et qui nous sert pour calculer toutes nos données, après avoir réalisé la connection de notre base de donnée à qlik, nous avons choisi d'importer uniquement cette table dans QlikView qui va nous permettre de répondre aux questions futures..

- Calculer l'indicateur Remise (€) avec seulement 2 décimales

Nous calculons au préalable cette indicateur dans la table `aggregate_table`, ils nous suffit d'utiliser la fonction `Num` : `Num(Remise,'##.00')` AS Remise, qui va nous permettre d'avoir la remise avec uniquement deux décimal

- Ajouter la dimension Temps (contenant le jour, le mois, le trimestre, et l'année) pour au moins les années 1997 à 1999 : à faire avec une boucle

Voir dans le script, ils nous a semblé plus intéressant (sans aucune prétention) de choisir comme date minimum la plus ancienne date de commande plutôt que 1997, ainsi que la date la plus récente (plutôt que 1999).

Nous utilisons donc :

```
Let vMinDate = Peek('OrderDate', 0, 'Orders'); // ou 1997
```

```
Let vMaxDate = Peek('OrderDate', -1, 'Orders'); // ou 1999
```

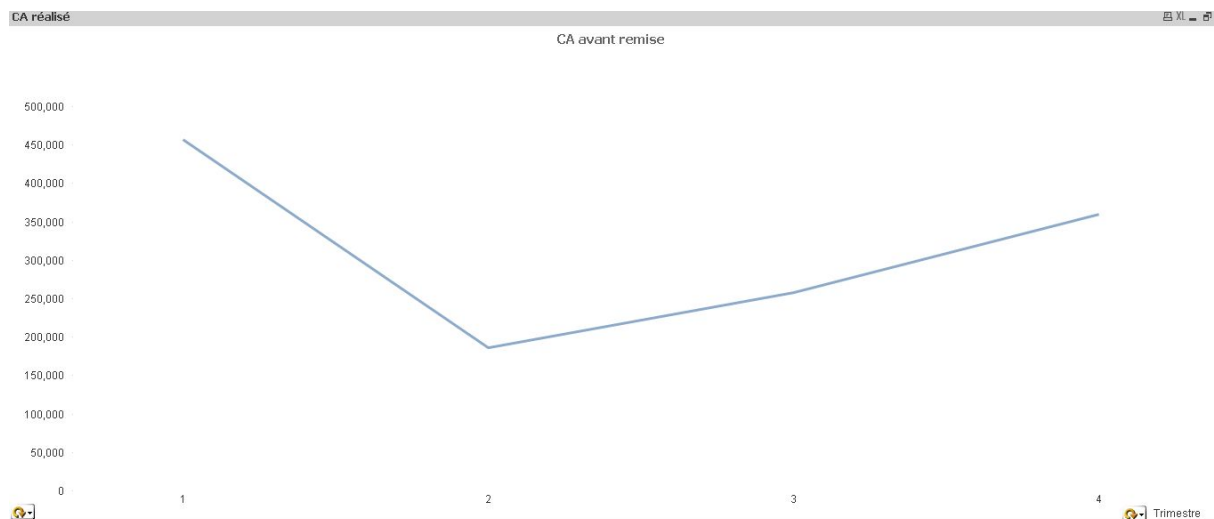
- Construire les 2 restitutions suivantes. Par souci de cohérence de navigation regroupez les sélecteurs globaux ensemble, et positionnez les de la même manière :
- Onglet Analyse des ventes : restituer sous la forme de différents graphiques, carte ou tableau le CA réalisé (montant des ventes) avant et après remise. Les critères de filtre possibles seront les suivants : année, trimestre, mois, catégories de produit, pays de la vente


Du fait de notre `aggregate table`, nous avons pu réaliser ces graphiques, nous avons utilisé des cycles groups, pour pouvoir en quelques cliques changer de dimension en



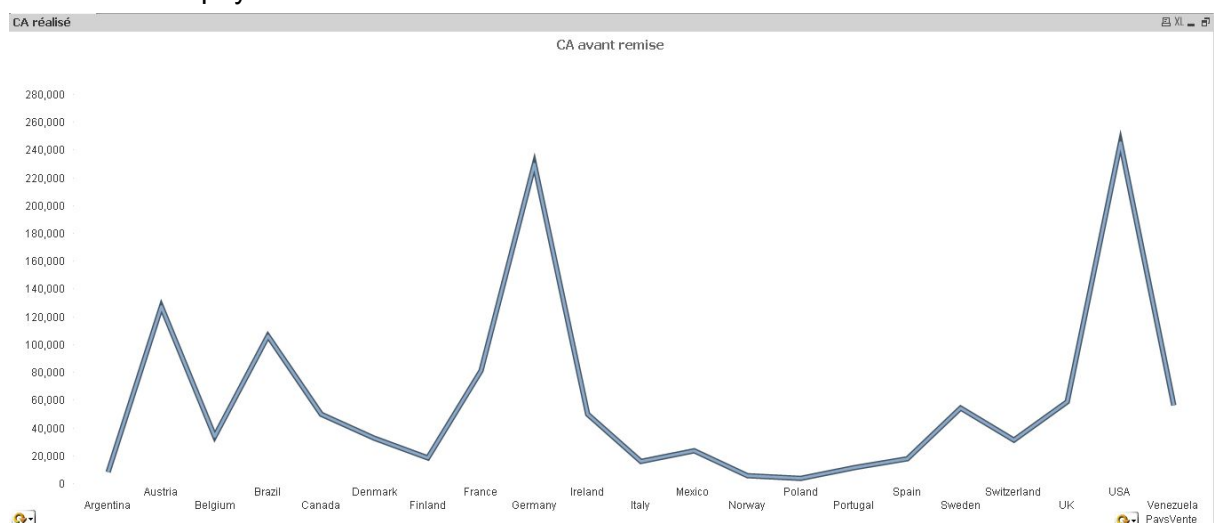
fonction du CA réalisé avec cette option : et ils va nous faire apparaître les informations voulues.

Exemple CA réalisé avant remise en fonction du trimestre :



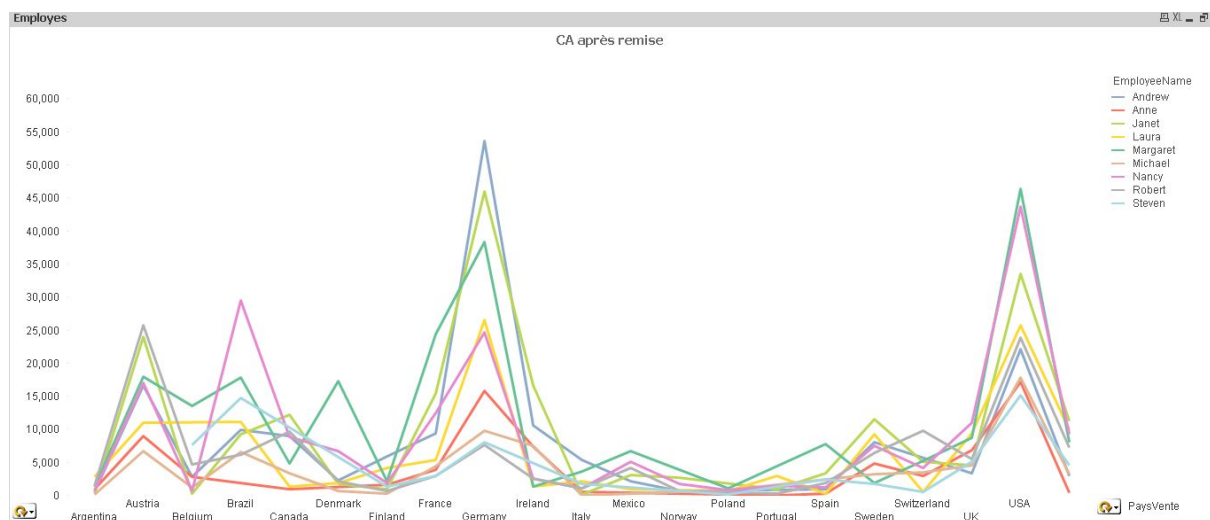
à l'aide de cette option  Année (le cyclics group) il est possible de changer la dimension et de choisir celle voulue.

En fonction du pays de vente :

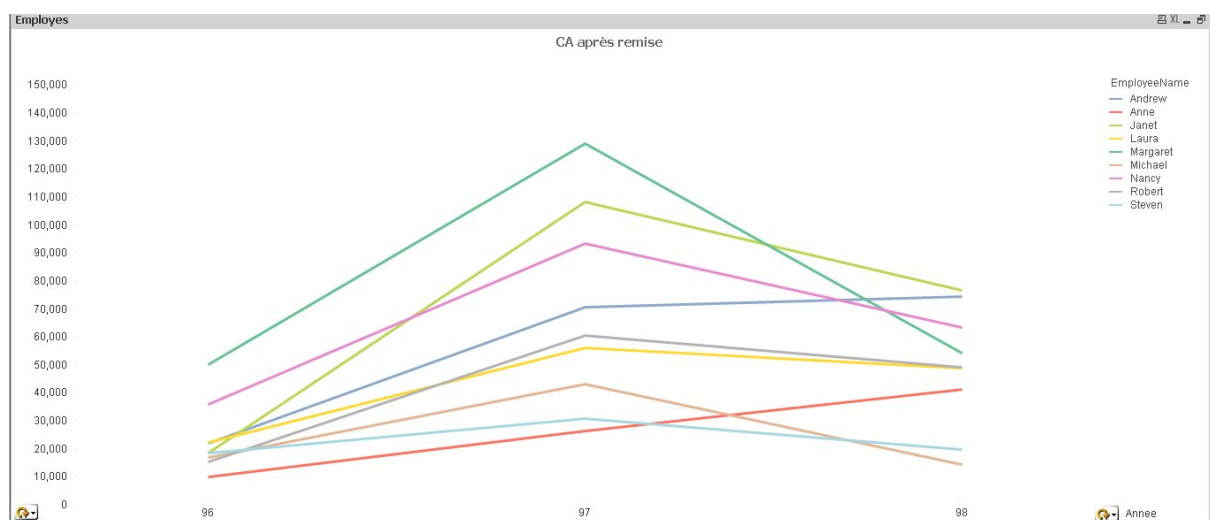


- Onglet Employés : restituer sous la forme de différents graphiques ou tableaux la liste des employés, et leur CA réalisé (remise). Les critères de filtre possibles seront les suivants : année, trimestre, mois, employés, tranche d'âge de l'employé, et pays de la vente

De même pour cette question, nous avons un seul graphique où il est possible de retrouver ces informations.



Il est également possible de changer la dimension, en année par exemple :



5) Conclusion

Lors de la réalisation de ce projet nous avons rencontré de nombreuses difficultés liés à notre manque de pratique sur les outils Talend et QlikView et également à cause de notre manque d'expérience sur la business intelligence à proprement parlé. En effet, de nombreux problèmes nous sommes survenus sans que l'on puisse les corrigés rapidement. Nous avons perdu beaucoup de temps à les traiter. L'analyse du sujet à également été une source de difficulté et particulièrement notre base de donnée où nous avons eu de longue hésitation quant aux tables et attributs que nous devons modéliser.

Nous sommes néanmoins satisfait du travail que nous avons réalisé, en effet la majeure partie du travail a été achevée avec succès. Il ne reste plus qu'à corriger un problème concernant les familles les plus rentables.

Ce projet a été une bonne occasion de mettre en pratique les différents aspects et fonctionnalités que nous avons vu précédemment sur les outils que sont Talend et QlikView.