

Réalisé par : ENSAM Casablanca

Lifi El Mostafa & Amine Gatou IA&GI

Projet BI : Analyses

Des plaintes d'une assurance

**Sommaire**

[**I. Introduction : Business Intelligence ou L’informatique décisionnelle 3**](#_Toc73214486)

[**II. Présentation de Projet 4**](#_Toc73214487)

[*1. Problématique 4*](#_Toc73214488)

[*2. Objectifs préliminaires 4*](#_Toc73214489)

[*3. Outils utilisés 5*](#_Toc73214490)

[**III. Partie Théorique 7**](#_Toc73214491)

[*1. Etude du besoin 7*](#_Toc73214492)

[*2. Table d’analyse Décisionnelle 7*](#_Toc73214493)

[*3. Schéma en Etoile 8*](#_Toc73214494)

[**IV. Réalisation du projet 9**](#_Toc73214495)

[*1. Création du Data Warehouse 9*](#_Toc73214496)

[*2. Intégration des données 14*](#_Toc73214497)

[*3. Création du cube – Analyses Services multidimensionnelles 22*](#_Toc73214498)

[*4. Visualisation sur Power BI 32*](#_Toc73214499)

[**V. Conclusion 39**](#_Toc73214500)

# Introduction : Business Intelligence ou L’informatique décisionnelle

Business intelligence ou L’informatique décisionnelle couvre toutes les solutions informatisées pour améliorer la prise de décision des décideurs dans l’organisation. Dans ses débuts, l’informatique décisionnelle s’est contentée tout d’abord de dupliquer les bases de données des systèmes de gestion, afin d’isoler les requêtes d’analyse de données des requêtes opérationnelles. Les requêtes d’analyse étant souvent très lourdes, l’objectif était surtout de préserver les performances des systèmes opérationnels. Ensuite cette base de données dédiée aux requêtes et à l’analyse a progressivement muté et s’est organisée.

Partant du constat qu’il est difficile de croiser des données contenues dans des bases de données distinctes, le plus simple a été de regrouper ces données éparses. Le concept de la base unique pour centraliser les données de l’entreprise est plus que jamais d’actualité. Il s’agit du concept d’entrepôt de données (ou Data Warehouse).

S’il est plus simple d’analyser ces données une fois qu’elles sont dans l’entrepôt de données, il n’en reste pas moins qu’il faut tout de même remplir l’entrepôt de données. L’extraction et le croisement des données des différents systèmes opérationnels puis le chargement dans l’entrepôt de données, ont fait émerger des outils dédiés à cette tâche, avec des concepts métiers qui leur sont propres : les outils d’ETL (Extract Transform Load).

Si au début, les requêtes d’analyses portaient sur une base relationnelle (dites OLTP pour OnLine Transaction Processing), le concept de base multidimensionnelle (dites OLAP pour OnLine Analytical Processing) s’est démocratisé fin des années 90.

Ce concept de bases de données offrait des performances très largement supérieures aux bases OLTP pour répondre à des requêtes d’analyse. Ces bases OLAP se sont alors couplées avantageusement avec l’utilisation de l’entrepôt de données. En effet, elles offraient à la fois un environnement plus performant, mais permettaient également aux utilisateurs finaux de bénéficier d’une interface simplifiée d’accès aux données, beaucoup plus intuitive qu’une base de données OLTP. On parle alors de métamodèle.

# Présentation de Projet

## Problématique

Une société d’assurance voudrait faciliter l’action d’analyse des données et améliorer la prise des futures décisions. Pour cela, l’administration a décidé de réaliser un projet « Business Intelligence » dans le but d’analyser les données des différents types de plaintes que la société a reçu depuis sa création et aussi trouver une solution au problème de stockage de ces données qui sont stockées dans des fichiers Excel non-sécurisés.

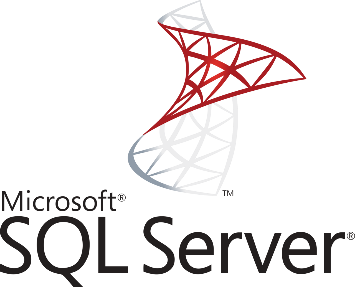
La gestion des plaintes est une mission d’une grande importance pour la société d’assurance car une bonne gestion des plaintes permet la société d’améliorer leur relation avec les clients et de prendre des décisions proactives concernant les courtiers et les remboursements. C’est pour ceci, ce projet doit permettre l’analyse des données des plaintes par rapport plusieurs axes, par exemple :

* **Client (Sexe, Satisfaction, Age, Groupe d’Age…)**
* **Courtier (Canal de distribution, réseau, performance…)**
* **Priorités (Urgent, Non-Urgent)**
* **Statut (Ouvert, Assigné, Fermer…)**
* **Date (Date de dépôt, date de résolution, nombre de jour…)**
* **Région (Ville, Etat, Village…)**

## Objectifs préliminaires

* Création d’une base de données sécurisées pour pouvoir stocker les données.
* Utilisation des outils d’intégration des données (Integration Services) pour réaliser le flux d’alimentation ETL.
* Utilisation des outils d’analyse service pour la création des bases multidimensionnelles (OLAP).
* Utilisation des outils de visualisation pour extraire les données de puis l’entrepôt de données et les importer pour les utiliser dans la création des rapports interactifs.

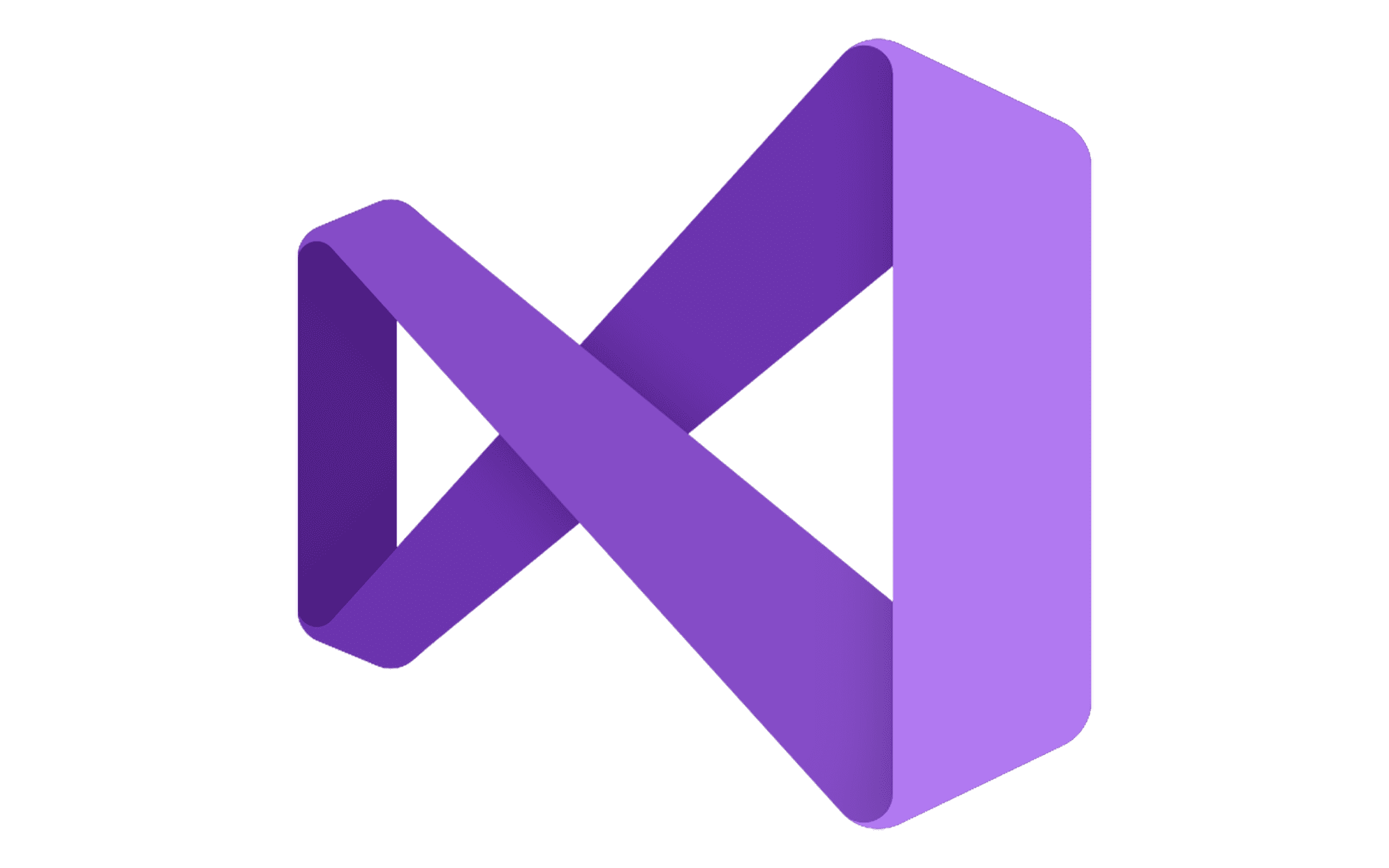
## Outils utilisés



**Microsoft SQL Server 2019** : est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft. Microsoft a inclus un certain nombre de services dans SQL Server liés à l'intelligence d'affaires et à l'entreposage de données. Ces services incluent Integration Services, Reporting Services et Analysis Services

****

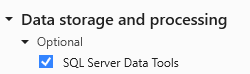
**Microsoft SQL Server Management Studio** : (SSMS) est l'outil multilingue de gestion des bases de données de Microsoft SQL Server et permet l'interaction entre le code SQL nécessaire à la manipulation des bases de données par les développeurs, comme à la gestion par les administrateurs de bases de données des différentes instances SQL Server.



**Visual Studio 2019 :** Microsoft Visual Studio est une suite de logiciels de développement pour Windows et mac OS conçue par Microsoft. La dernière version s'appelle Visual Studio 2019.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web, des services web, des applications bureautiques et des applications mobiles et gestion des bases de données.

Mais ce qui nous intéresse c’est :



Grâce à cette installation on peut utiliser maintenant tout ce qui concerne la gestion des bases de données SQL Server en utilisant plusieurs outils parmi eux :

* **SQL Server Integration Services (SSIS)** est un composant du logiciel de base de données Microsoft SQL Server qui peut être utilisé pour effectuer un large éventail de tâches de migration de données.

SSIS est une plate-forme pour l'intégration de données et les applications de flux de travail. Il comporte un outil d'entreposage de données utilisé pour l'extraction, la transformation et le chargement de données (ETL). L'outil peut également être utilisé pour automatiser la maintenance des bases de données SQL Server et les mises à jour des données de cube multidimensionnelles.

* **Microsoft SQL Server Analysis Services SSAS** est un traitement analytique en ligne (OLAP) et un outil de data mining dans Microsoft SQL Server. SSAS est utilisé comme un outil par les organisations pour analyser et donner du sens aux informations éventuellement réparties sur plusieurs bases de données, ou dans des tables ou des fichiers disparates. Analysis Services comprend un groupe de fonctionnalités OLAP et d'exploration de données et se décline en deux versions : *multidimensionnelle et tabulaire.*



**Microsoft Power BI :** Power BI est un ensemble de services logiciels, d'applications et de connecteurs qui fonctionnent ensemble pour transformer différentes sources de données en informations visuelles, immersives et interactives. Plusieurs sources de données peuvent être utilisées telles que des fichiers Excel, des sources SQL, ou des entrepôts de données hybrides locaux ou sur le cloud. Les données sont personnalisées et interactives avec une interface suffisamment simple pour que les utilisateurs finaux créent leurs propres rapports et tableaux de bord.

# Partie Théorique

## Etude du besoin

Dans cette partie, on va déterminer les tables de faits et de dimensions.

* **Table de fait**

D’après notre problématique, l’objectif est d’analyser les données des plaintes suivant plusieurs axes, donc on peut conclure que notre table de fait correspond à la table des plaintes.

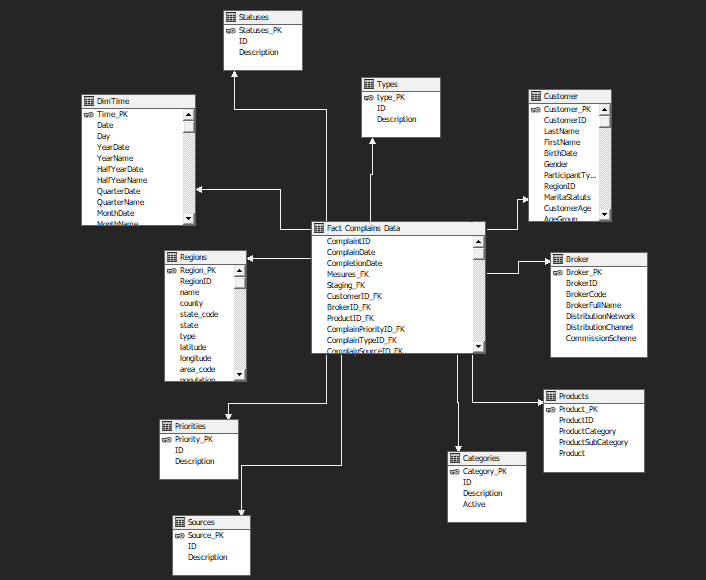
* **Les dimensions :**

Les dimensions sont les critères ou axes utilisés pour analyser les données des plaintes (*Dimension : Client, Courtier, Région, Produit, Catégories, Priorités, Temps, Sources, Statuts, Types*).

## Table d’analyse Décisionnelle

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Client** | **Courtier** | **Région** | **Produit** | **Catégories** | **Priorités** | **Temps** | **Statuts** | **Types** |
|  | ID | ID | ID | ID | ID | ID | Time ID | ID | ID |
|  | Nom | Code | Name | Famille | Description | Description | Date | Description | Description |
|  | Prénom | Full Name | Département | Sous-Famille | Active |  | Année |  |  |
|  | Date de naissance | Distribution Network | Code État | Nom Produit |  |  | Mois |  |  |
|  | Sexe | Distribution Channel | Etat |  |  |  | Jour |  |  |
|  | Type | Commission Scheme | Type |  |  |  | Semestre |  |  |
|  | RegionID |  | Latitude |  |  |  | Trimestre |  |  |
|  | EtatCivil |  | Longitude |  |  |  |  |  |  |
|  | Age |  | Population |  |  |  |  |  |  |
|  | Age Group |  | Households |  |  |  |  |  |  |
|  | Full Name |  | Médian income |  |  |  |  |  |  |
|  |  |  | Land area |  |  |  |  |  |  |
|  |  |  | Water area |  |  |  |  |  |  |
|  |  |  | Time zone |  |  |  |  |  |  |
| **Analyse des plaintes par les différentes dimensions** |  |  |  |  |  |  |  |  |  |

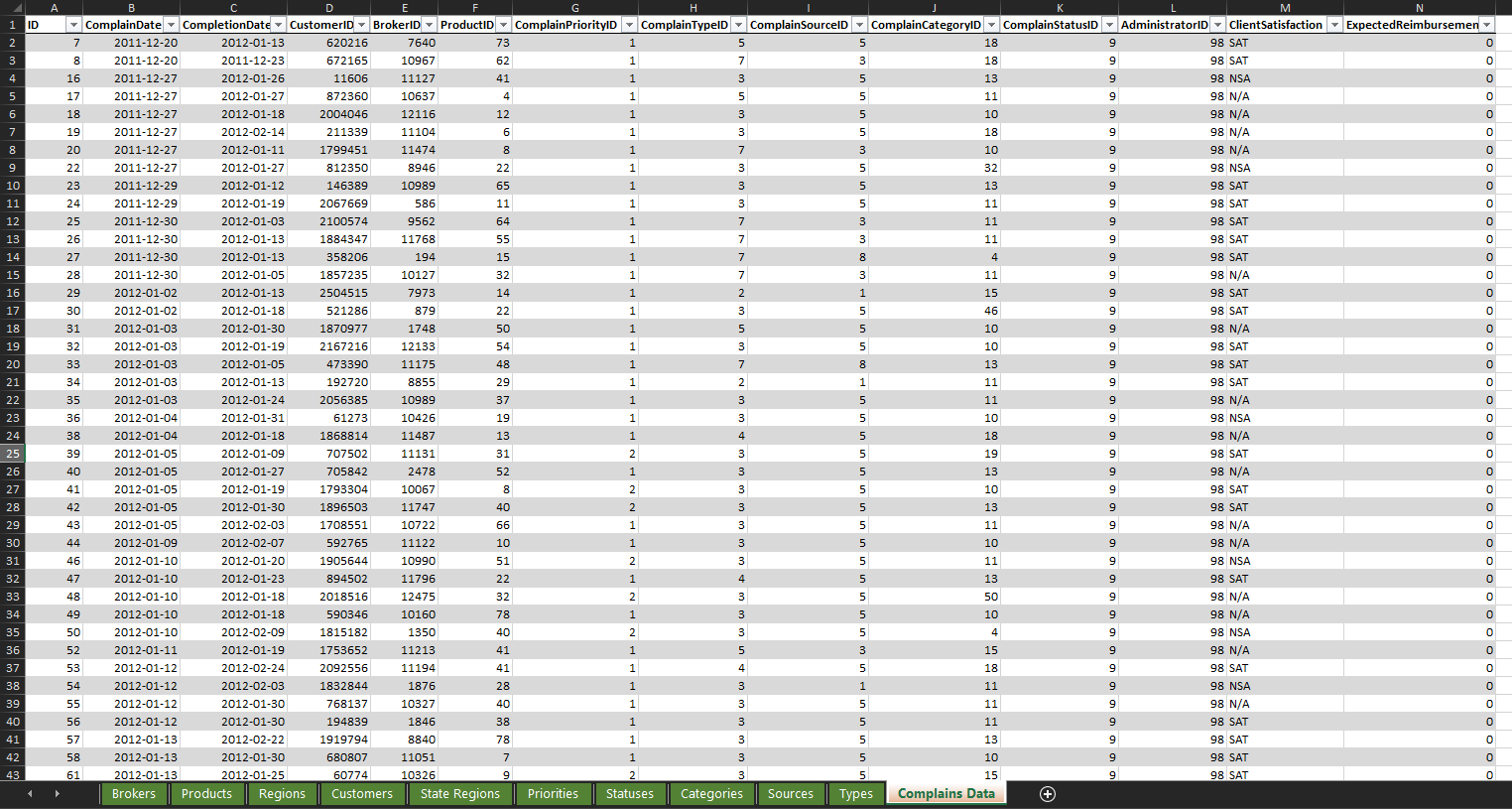
## 3. Schéma en Etoile



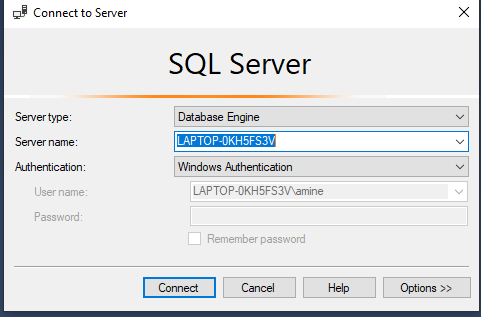
# Réalisation du projet

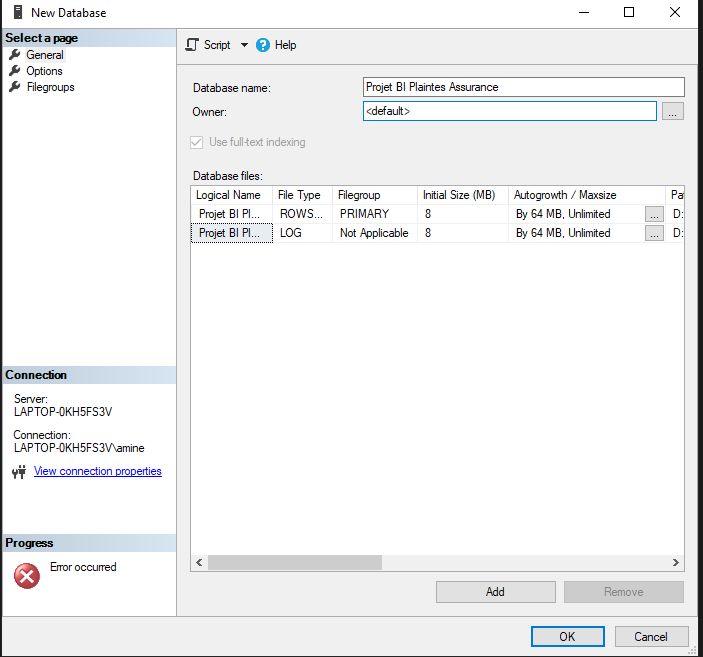
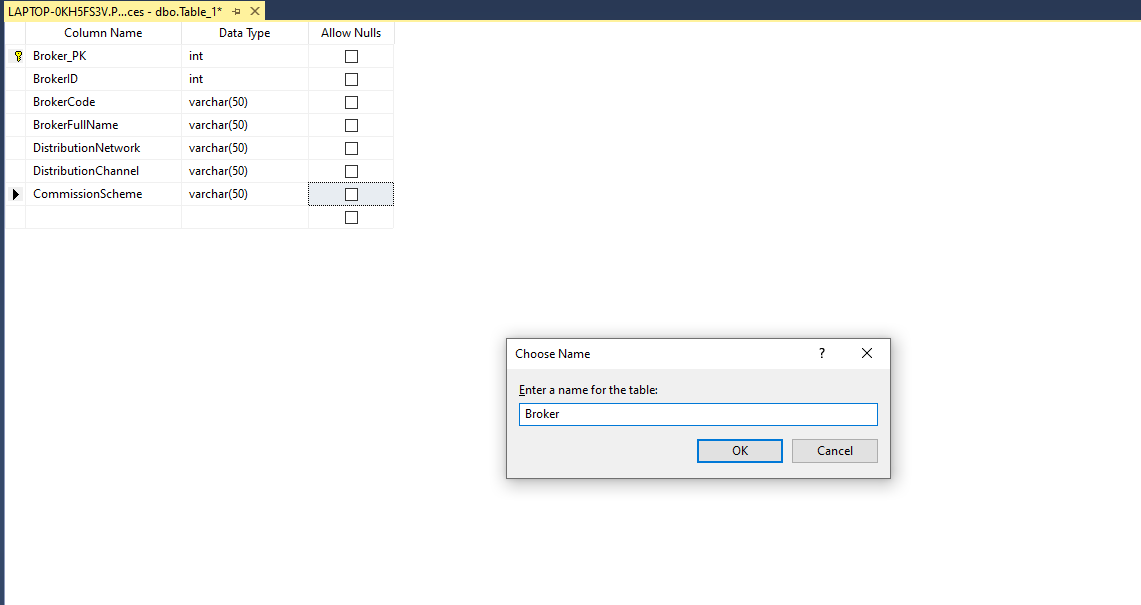
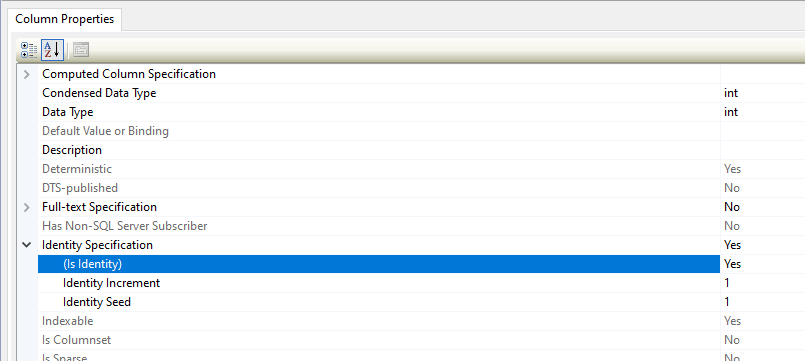
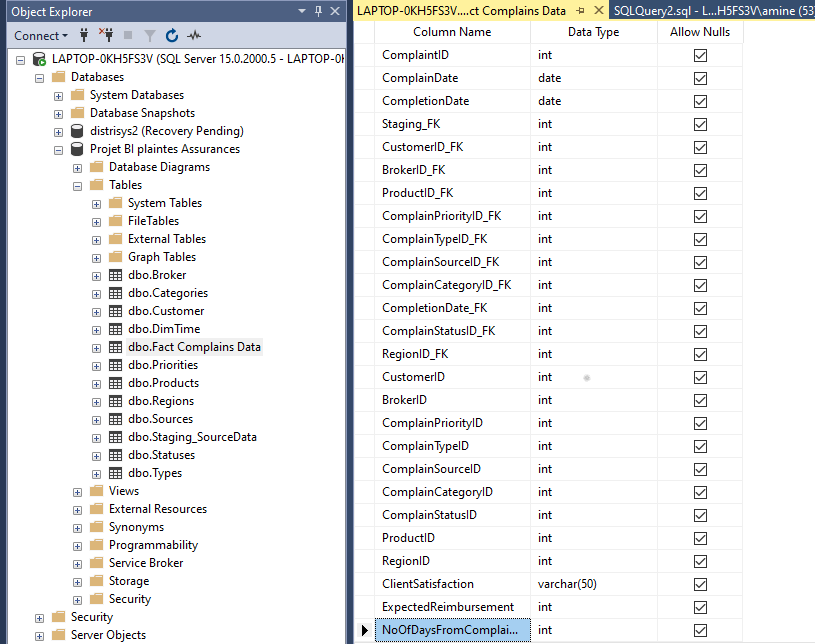
## Création du Data Warehouse

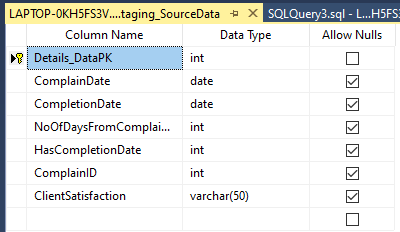
On a récupéré une base de données depuis le site [**https://forum.enterprisedna.co**](https://forum.enterprisedna.co)

Notre base de données prend la forme d’un fichier Excel qui contient les différentes données de gestion des plaintes d’une assurance pendant la période 2011-2021.

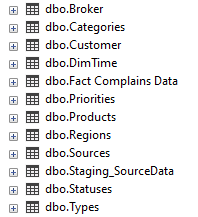
* On ouvre SQL Server Management Studio, on va se connecter :

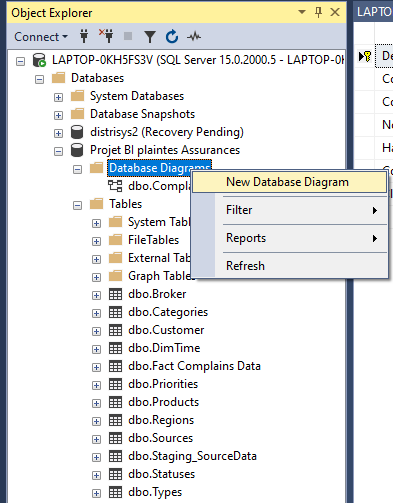


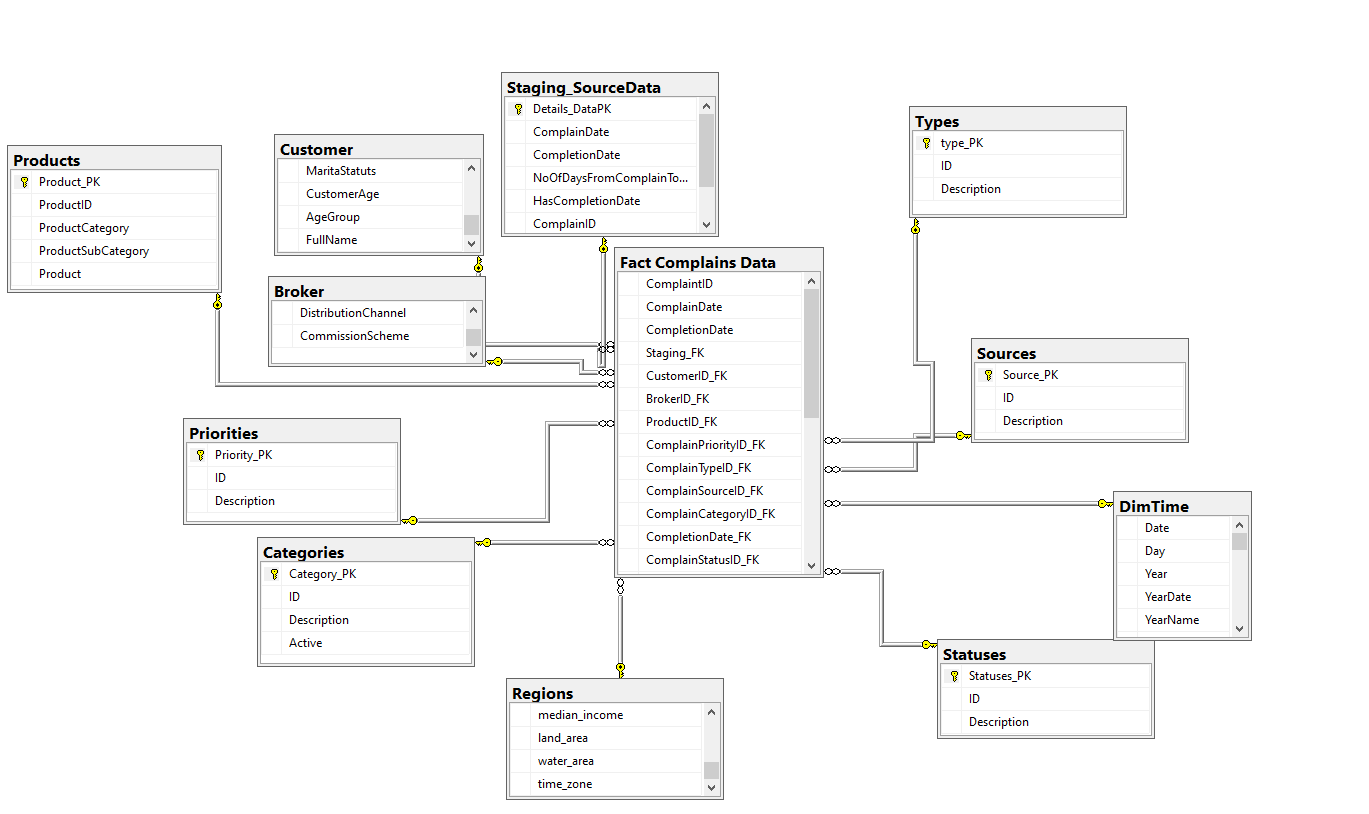
* Création d’une nouvelle base de données : Projet BI Plaintes Assurance
* Ensuite la création des tables : On ajoute des clés primaires spécifiques à la base de données pour toutes les tables.
* Sans oublier de mettre les clés primaires en mode « Auto Increment »
* On fait de même pour les autres dimensions avec leurs champs…
* Création de la table des faits : contient les clés étrangères + les champs + les différentes mesures.
* La table des faits contient des champs qu’on ne peut pas utiliser comme mesure mais on les utilise pour calculer les mesures, donc pour remédier à ce problème, on a créé une table nommée *Staging\_SourceData* qui comme son nom l’indique va être utile pour préparer nos données qui vont être utiliser pour le calcul des mesures.



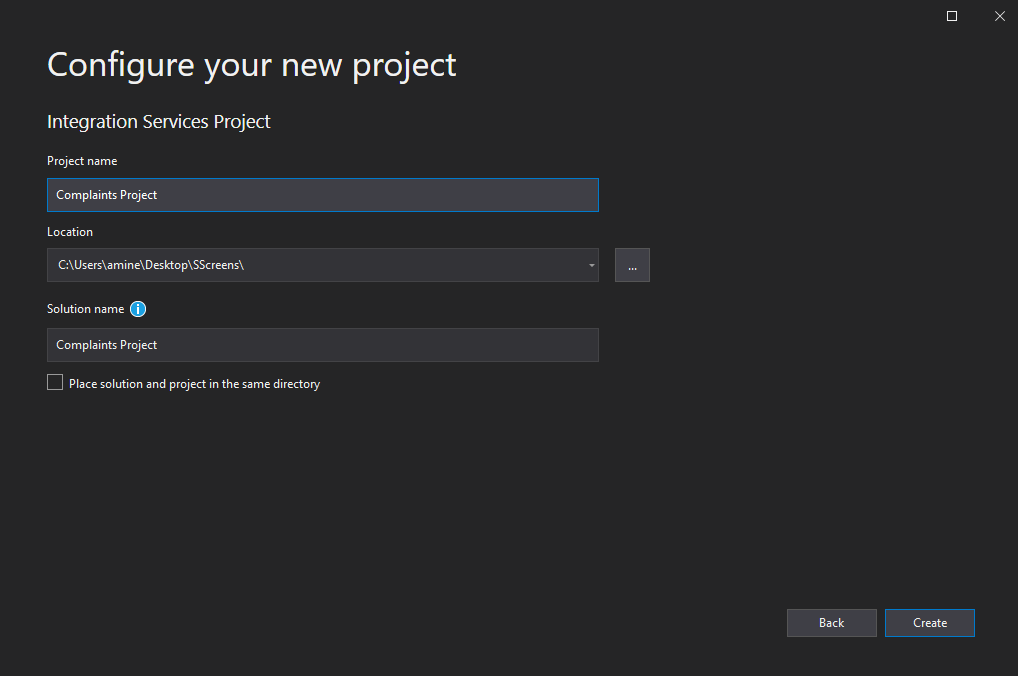
* Et au final, les tables sont correctement créées :



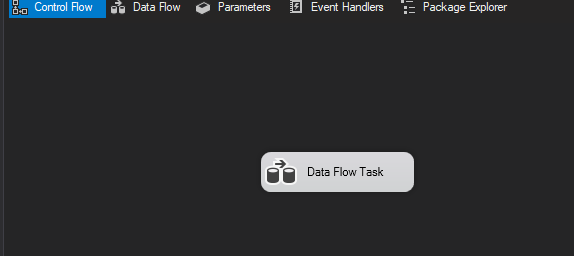
* ***Remarque :* La dimension ‘Time’ va être créer après !**
* **Pour créer le schéma de la base de données :
* Puis on ajoute les tables et on les relie à la table des faits grâce au clé étrangères et au final on obtient notre schéma :



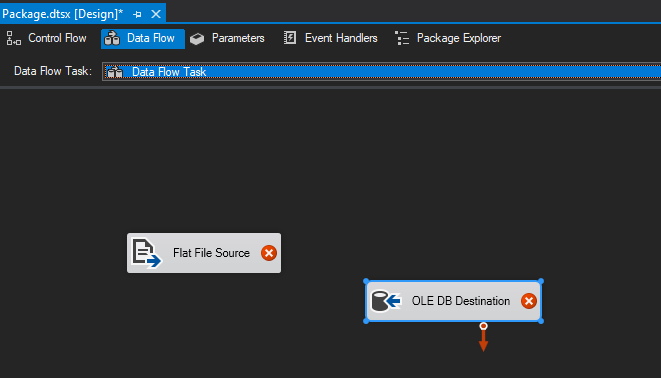
## Intégration des données

On va créer notre projet SSIS sur Visual Studio :

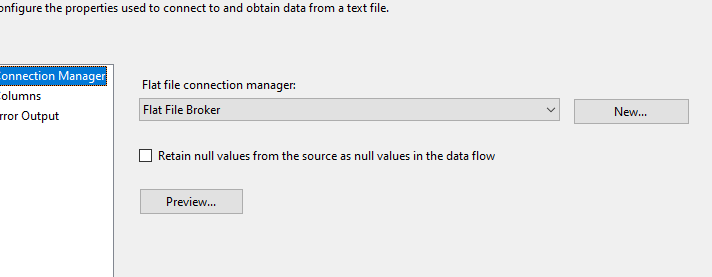
On ajoute un premier une tâche de flux de données dans le flux de contrôle :

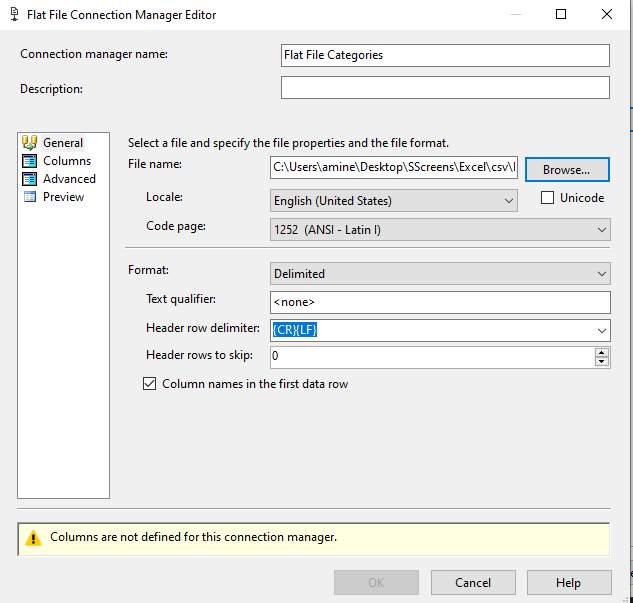


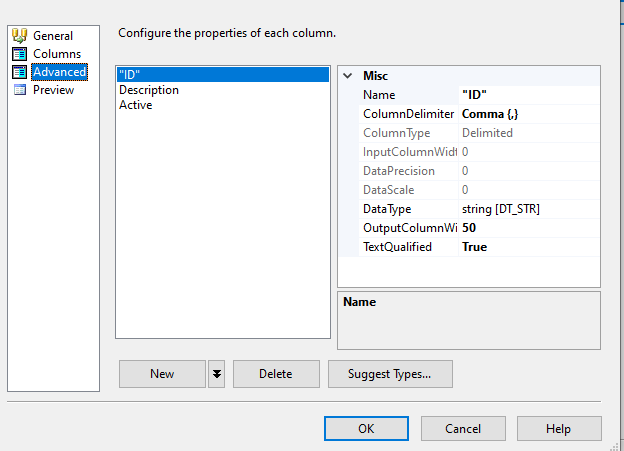
A l’intérieur de cette tâche, on va créer les flux de données : *source de fichier plat / destination OLE DB*



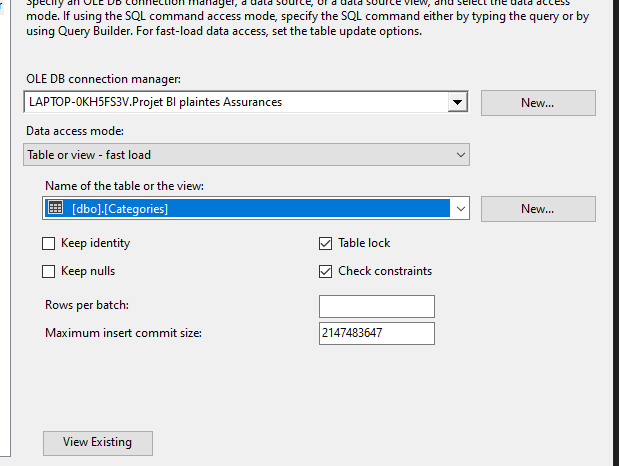
Passons à la configuration des flux de donnés !

* On ouvre la configuration du fichier plat :
* On crée un nouveau Connection manager et on choisit le fichier csv correspondant !

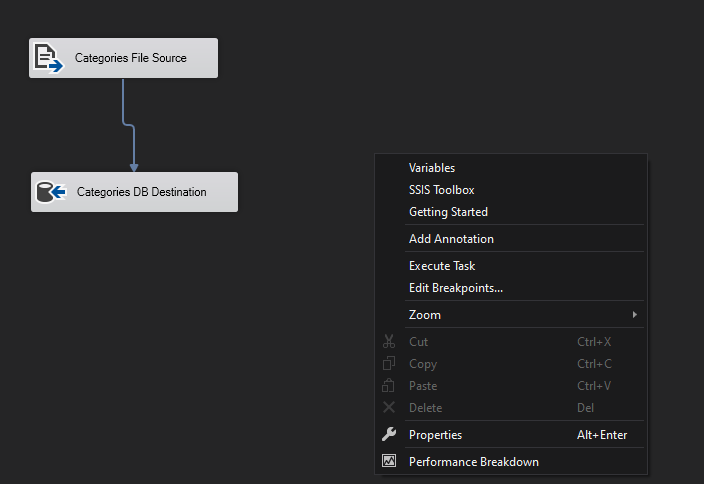
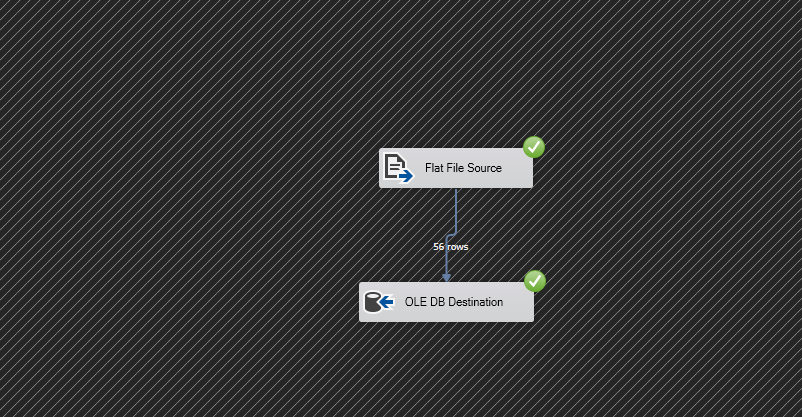


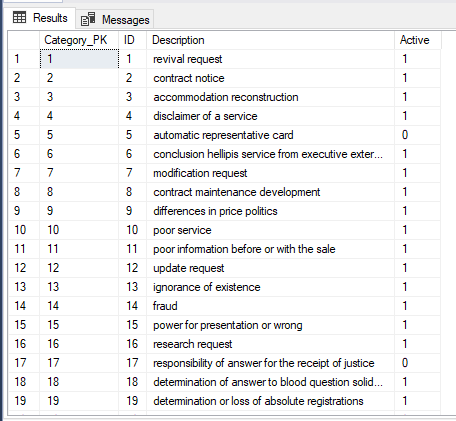
* On configure les types de données et leur taille :
* On relier le fichier plat à l’OLE DB Destination :

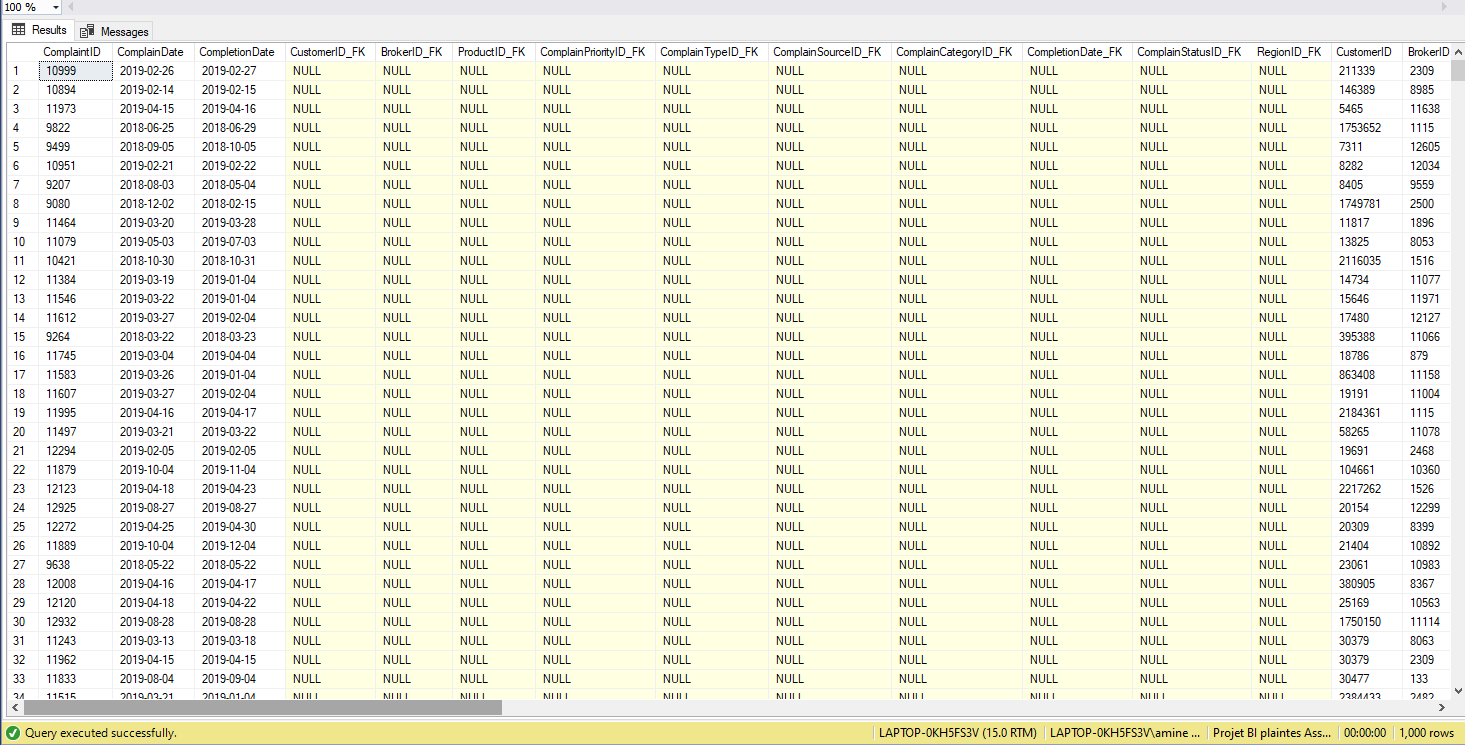
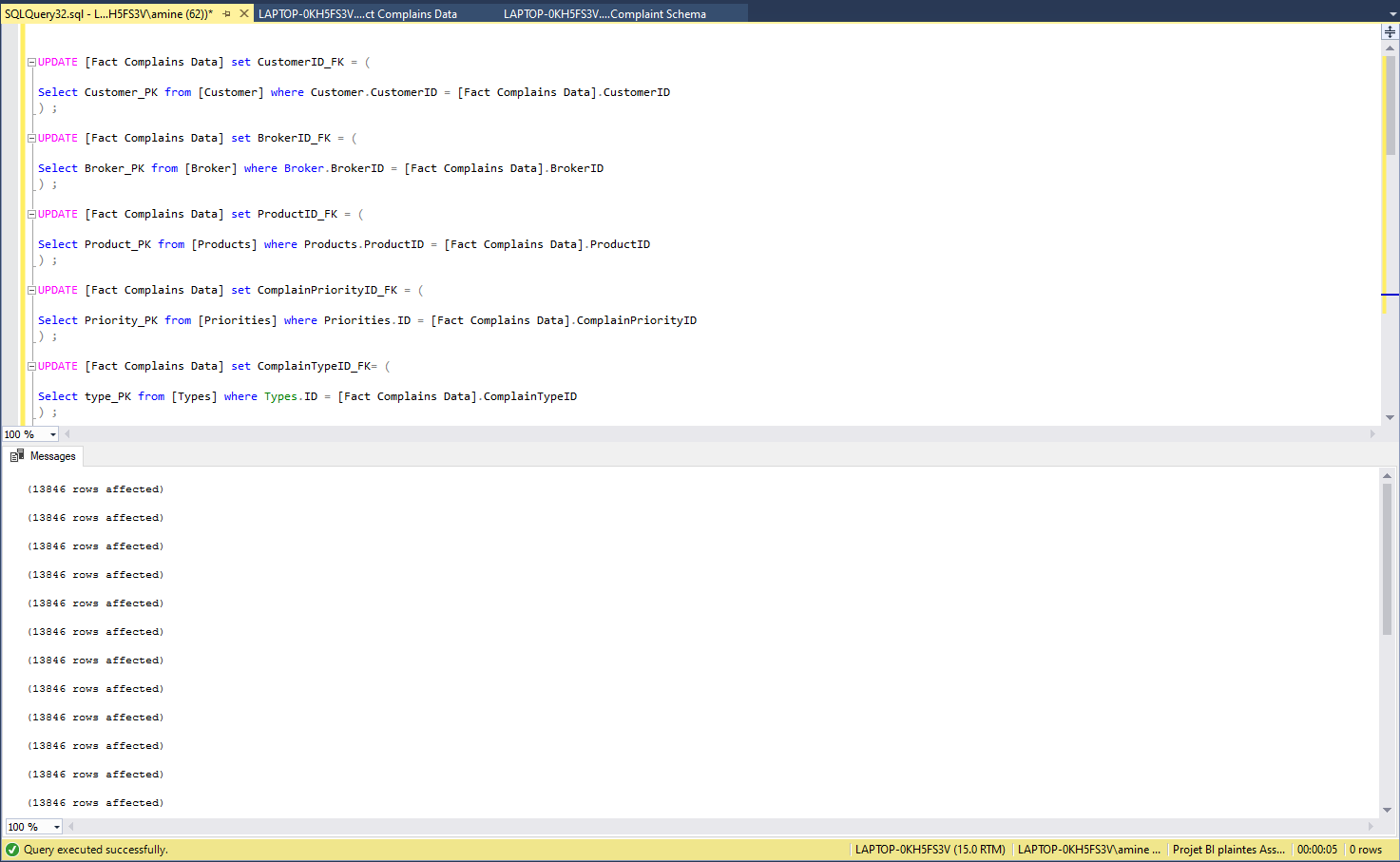


* On choisit la base de données et la table correspondante
* Puis on fait le mapping des données :



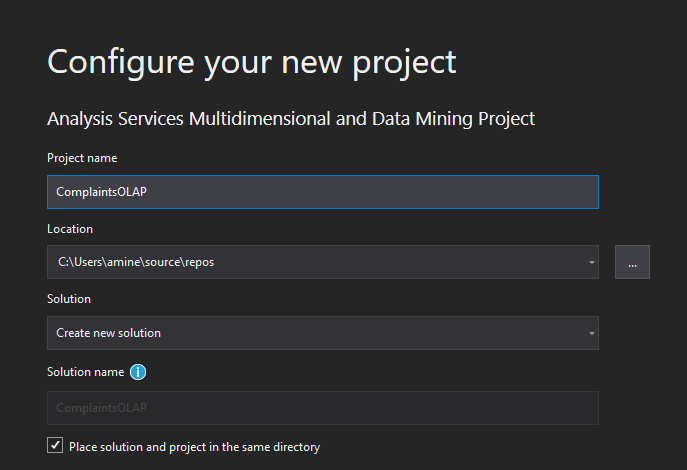
* On rennome maintenant les flux de données et on exécute la tâche
* Résultat et vérification dans la base de données !

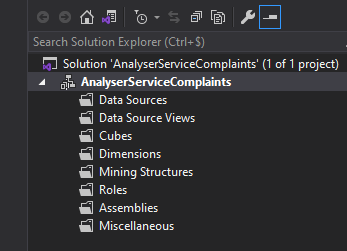


* On refait la même opération pour toute les tables depuis les fichiers csv correspondants à chaque table.
* Par contre, on remarque que les clés étrangères de la table des faits contiennent des valeurs NULL
* Pour remédier à ce problème on exécute le script suivant :
* Vérification :
* Ainsi, notre Data Warehouse est bien créer et bien alimenter en termes de données.
* Maintenant on passe à la création du cube OLAP.

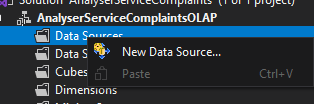
## Création du cube – Analyses Services multidimensionnelles

On commence par créer un projet SSAS sur Visual Studio



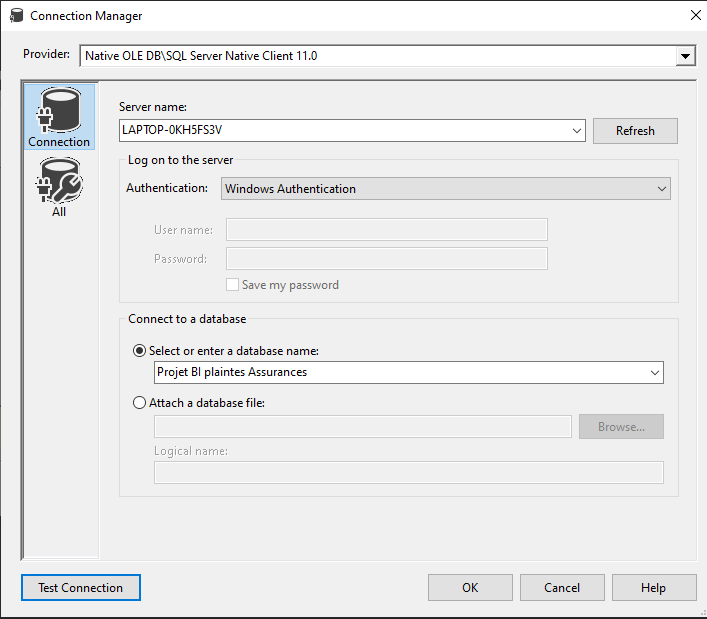


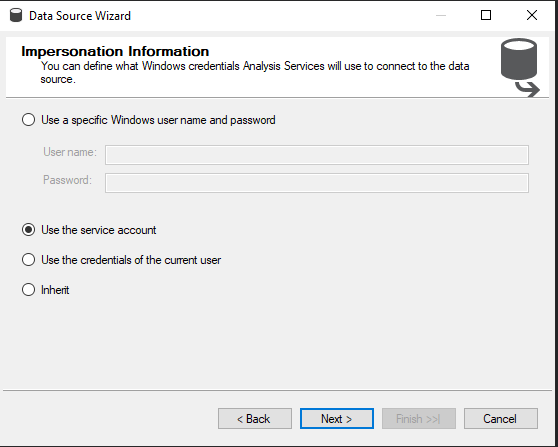
* Une fois le projet est créé, on commence par créer une nouvelle Data Source



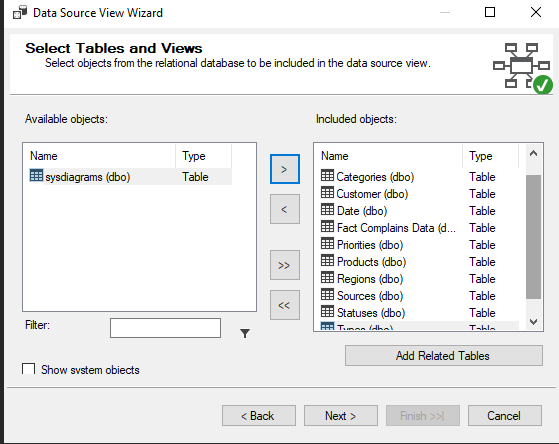
* En utilisant l’assistant, on choisit notre serveur de base de données et on utilise le compte service !

On choisit la base de données qu’on a créé au paravent et on se connecte au même serveur.

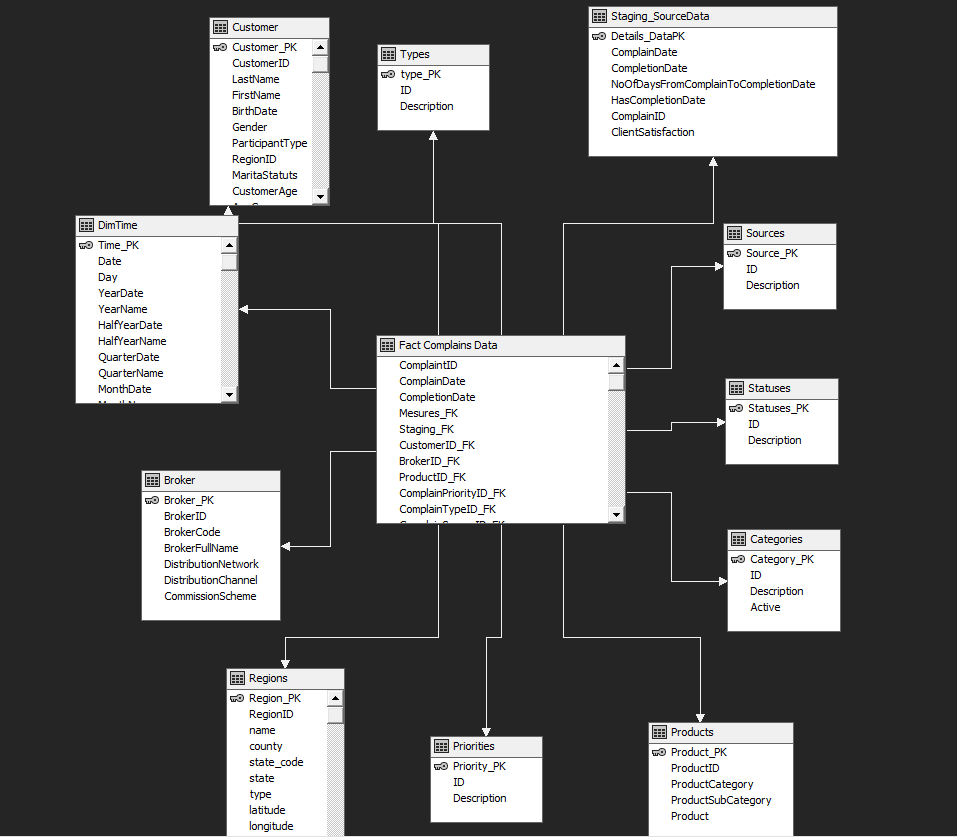


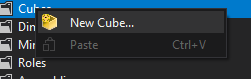


On choisit le compte service pour plus de sécurité.

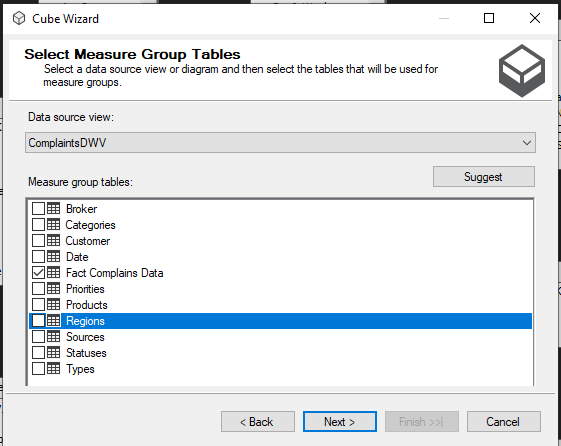
* Après, on procède à la création d’une vue de notre data source et on choisit les tables.



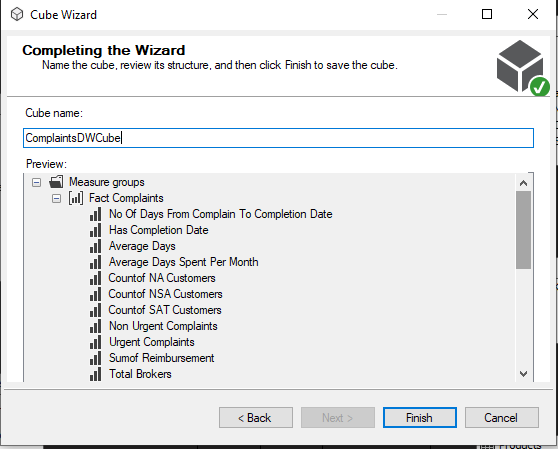
* On peut maintenant visualiser le modèle étoile de notre Data Warehouse.
* Après confirmation de notre schéma, on peut créer le cube sans problème et on doit choisir les tables existantes depuis l’assistant de création.

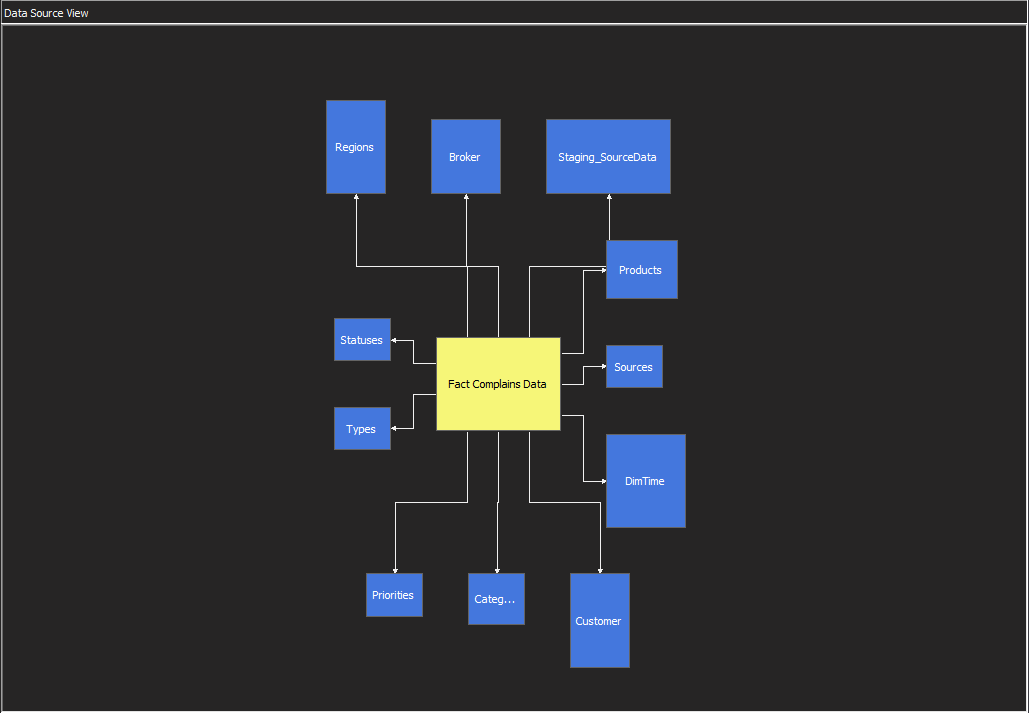


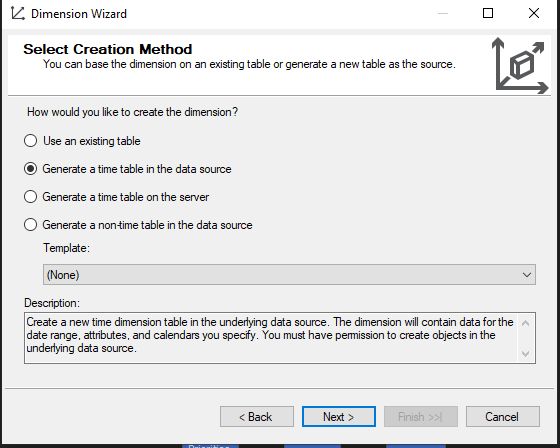
* On choisit maintenant la table de fait qui contient les mesures.



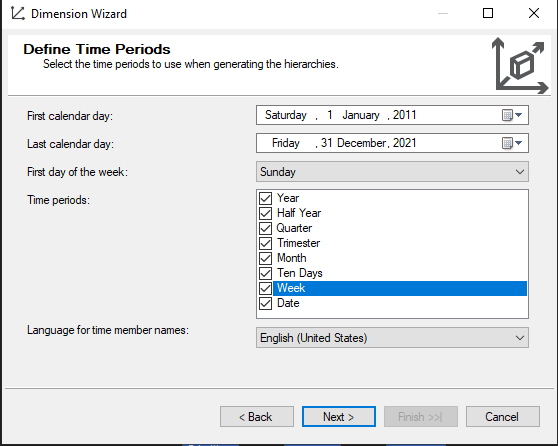
* On confirme que le cube est créé.

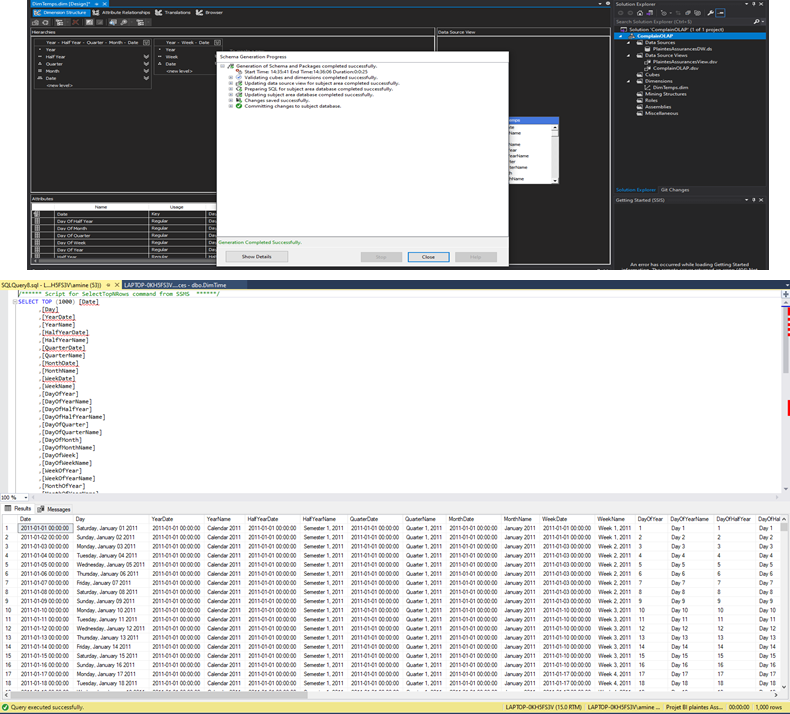


* Voici donc notre cube !
* Comme on avait mentionner auparavant, cet ici où on va créer la dimension du temps grâce à l’assistant :

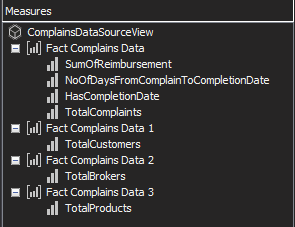
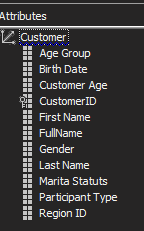


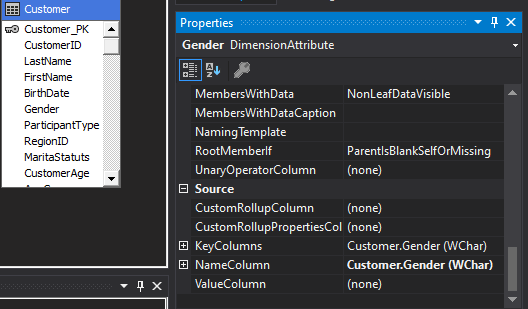
On génére une table de temps directement dans la base de données et la data source.

* On choisit la plage des dates et les champs désirés ! Et au final la table est automatiquement créée !
* 



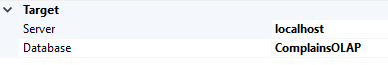
* On peut maintenant créer nos mesures, et aussi choisir les champs souhaités pour chaque dimension ! (Hiérarchie si disponible ou possible)



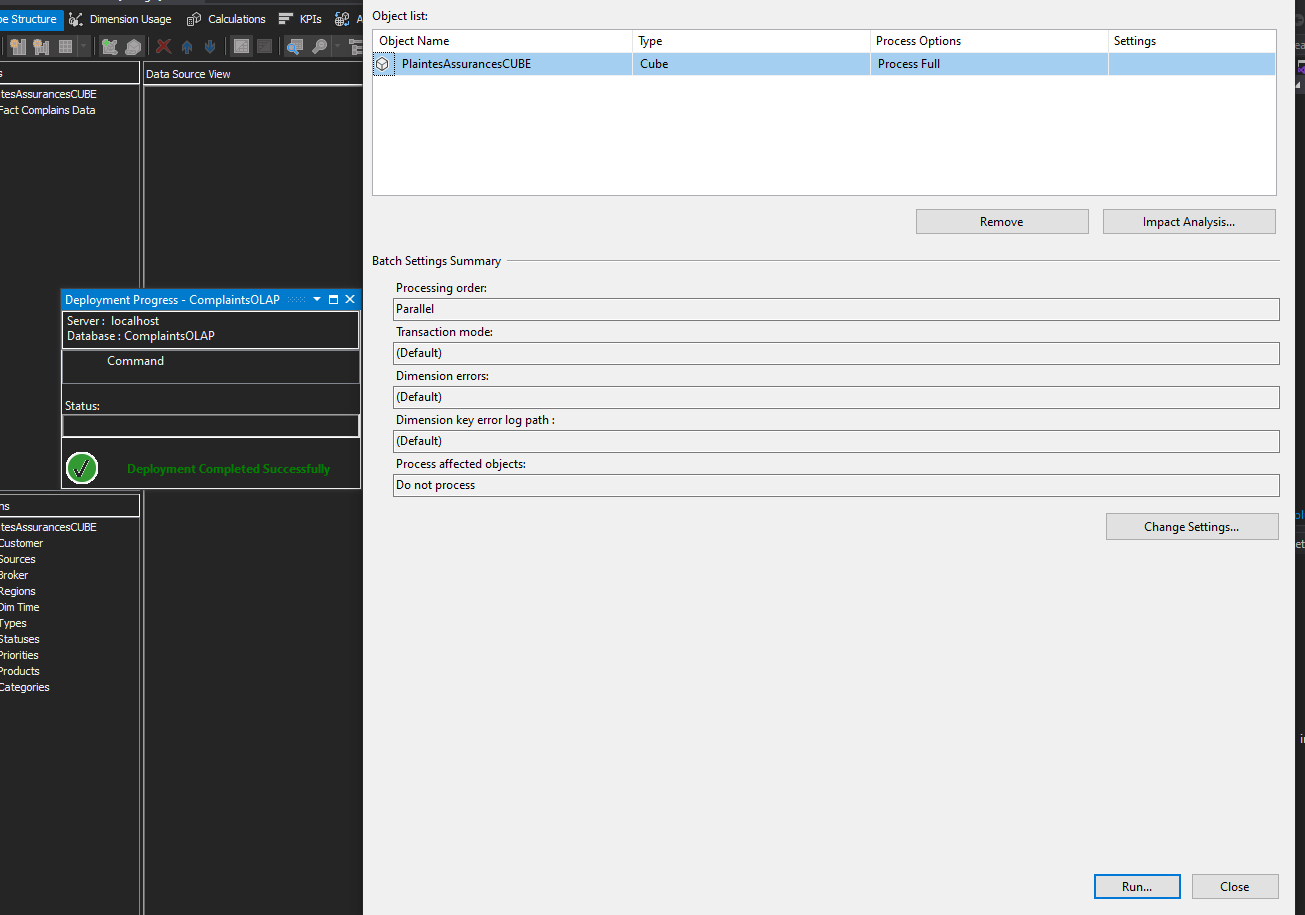


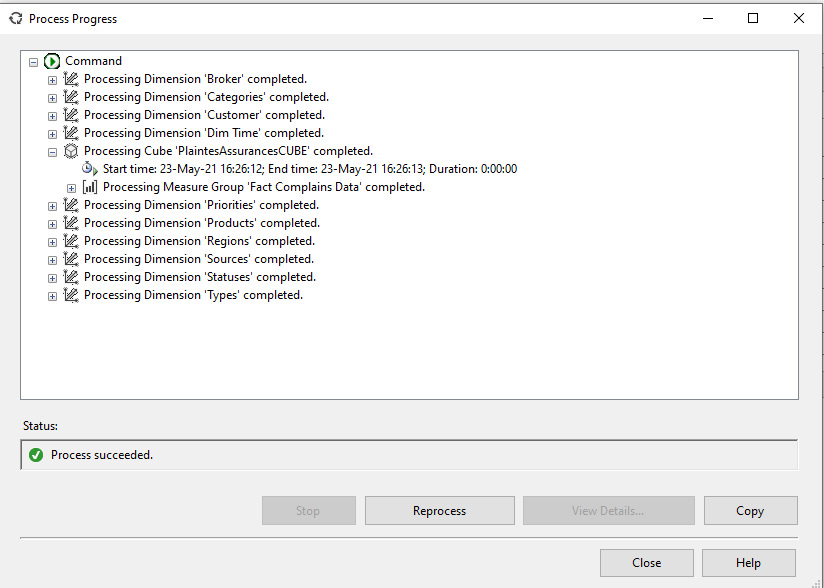
On Configure les propriétés des champs ainsi que leur type de donnés, taille, nom du champs… De même pour toutes les dimensions.

* Après les modifications nécessaires, on spécifie le serveur d’instance de SSAS :



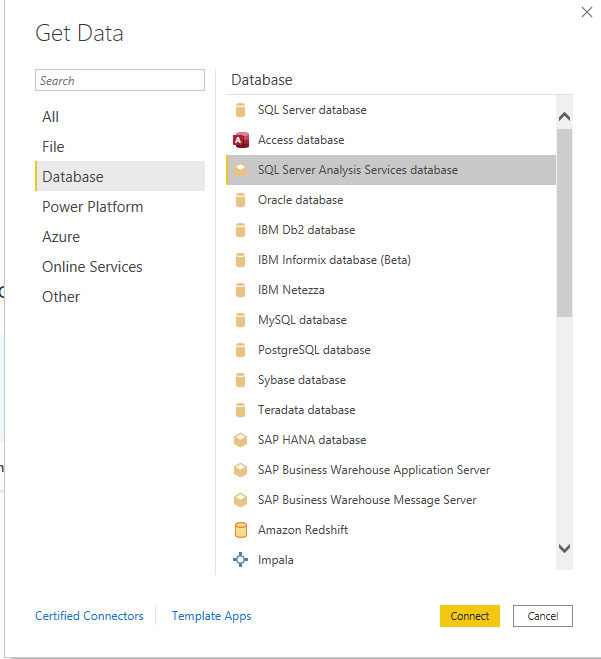
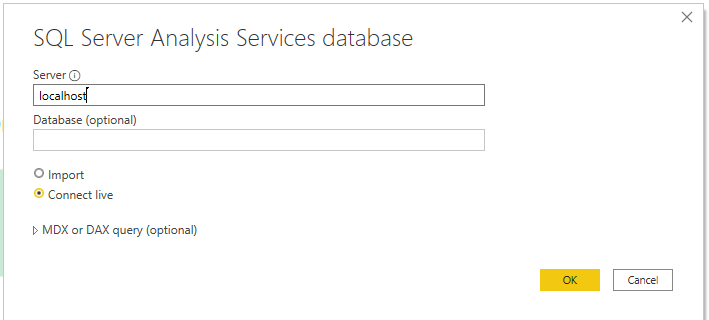
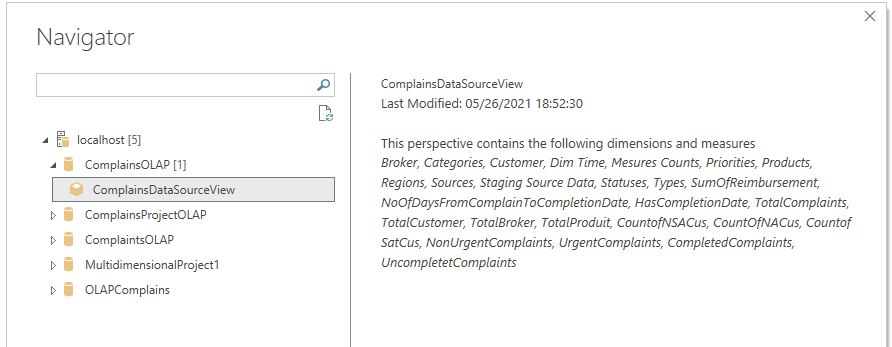
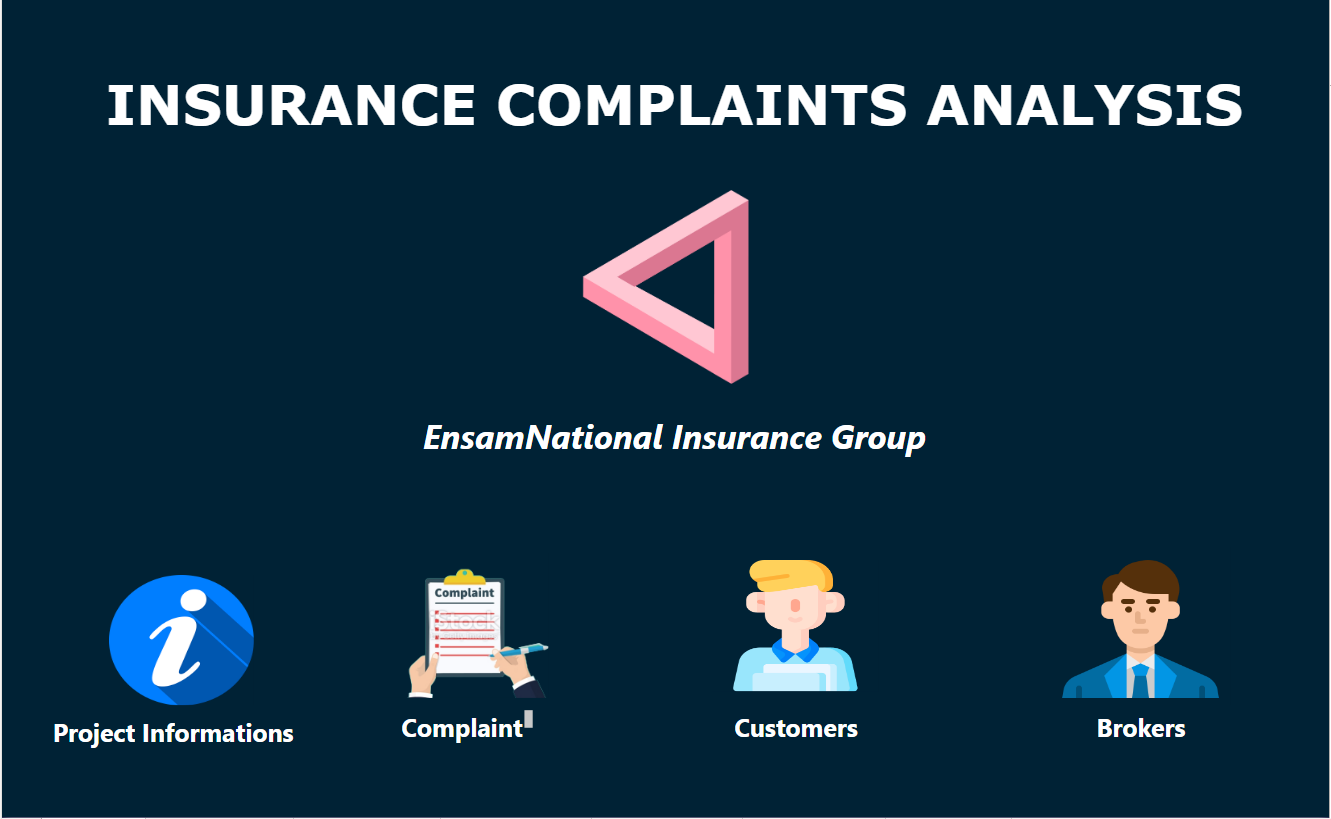
* Et on déploie notre cube vers le serveur.





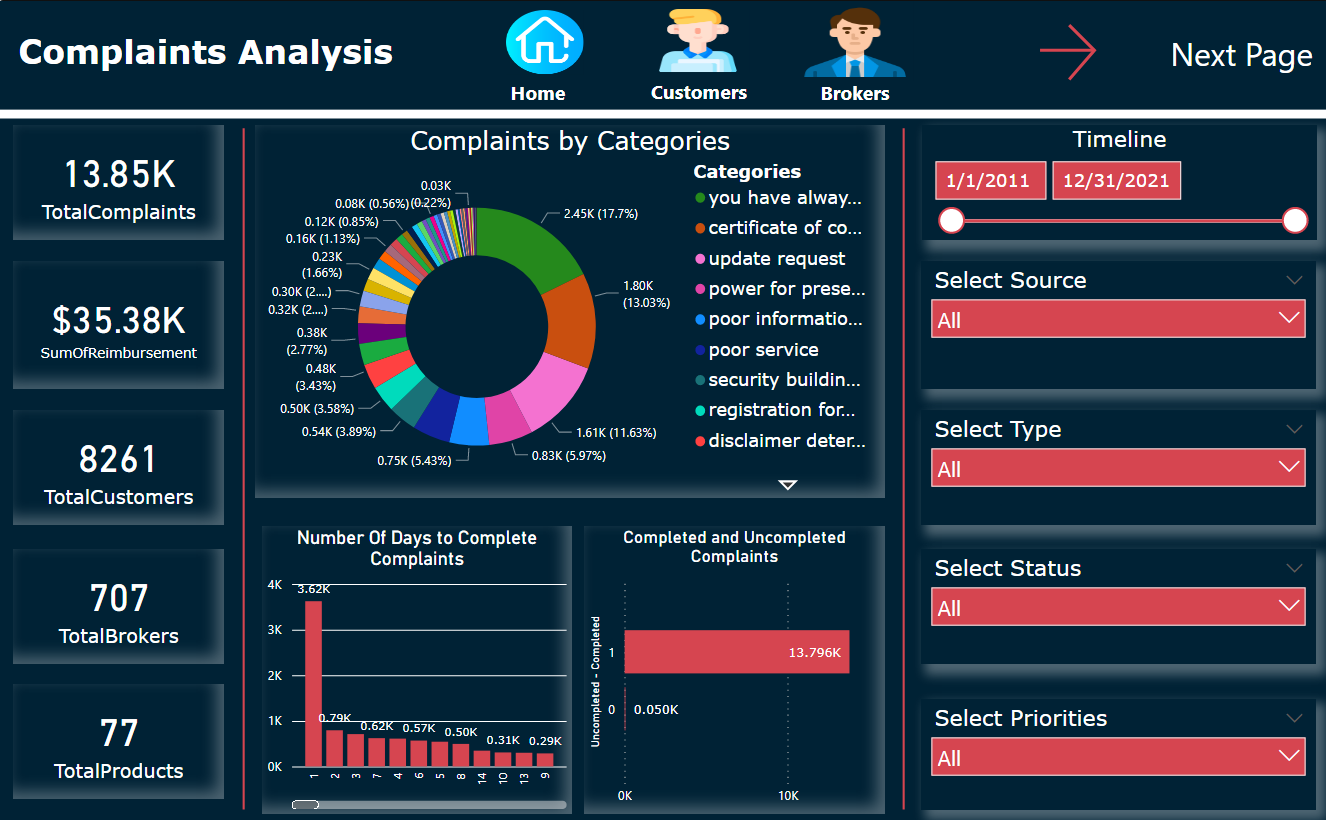
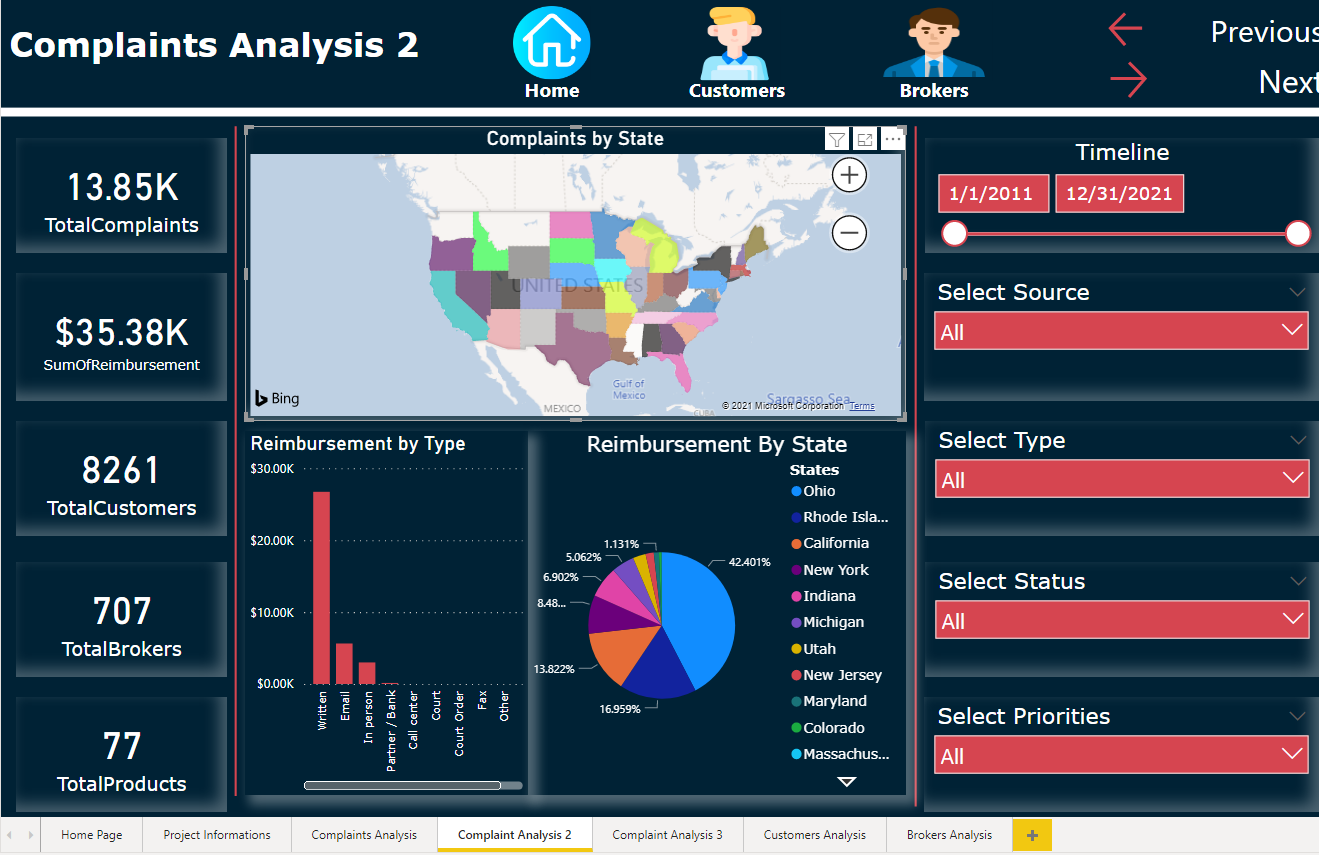
## Visualisation sur Power BI

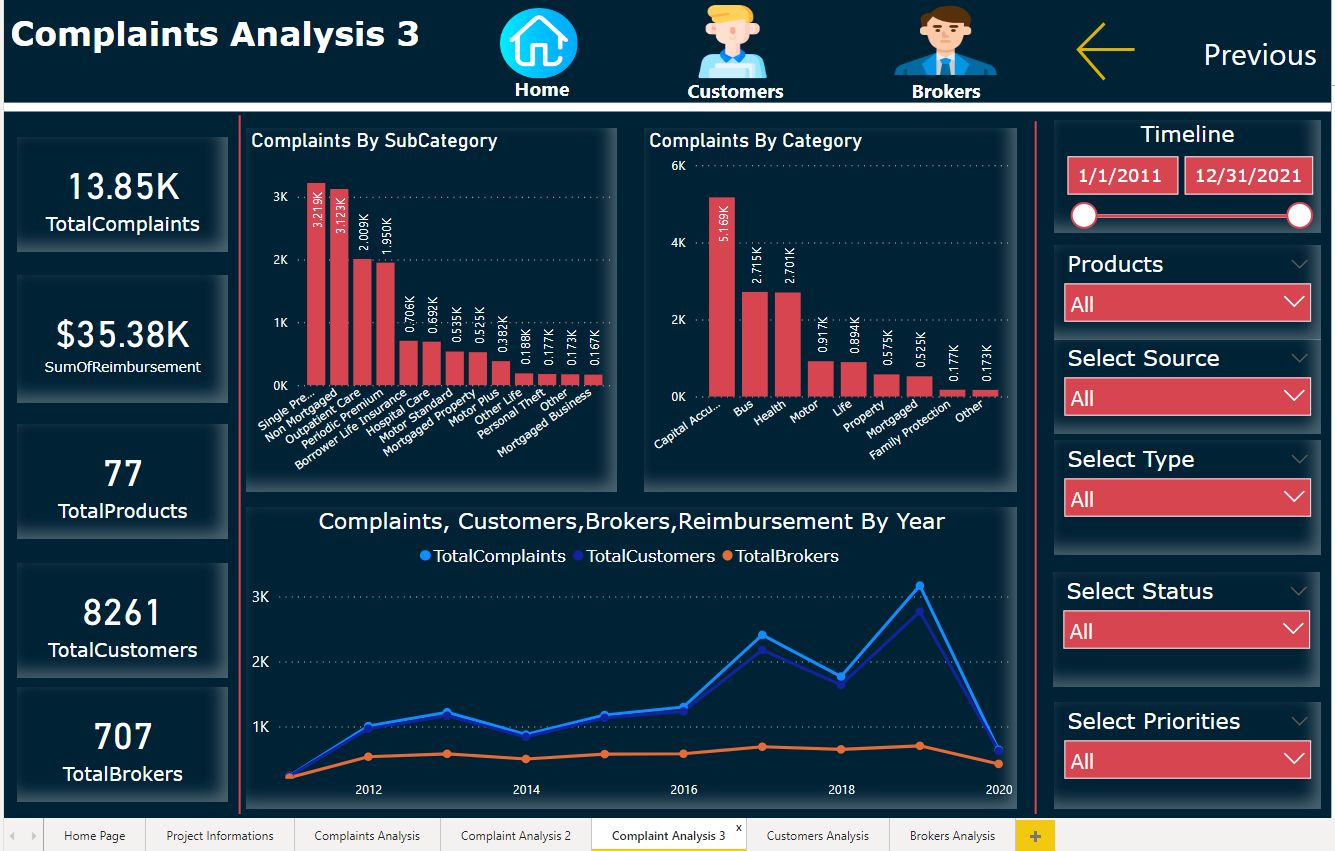
Maintenant on peut créer des tableaux de bords pour visualiser nos données !

* Importation des données !
* On se connecte au serveur, et choisi le mode ‘Connect Live’ pour que les données analysées soient connectées à tout changement dans notre Data Warehouse.
* On choisit le cube à importer !
* Et on trouve tous les mesures/dimensions de notre cube !
* Maintenant on peut créer n’importe quel type de visualisation

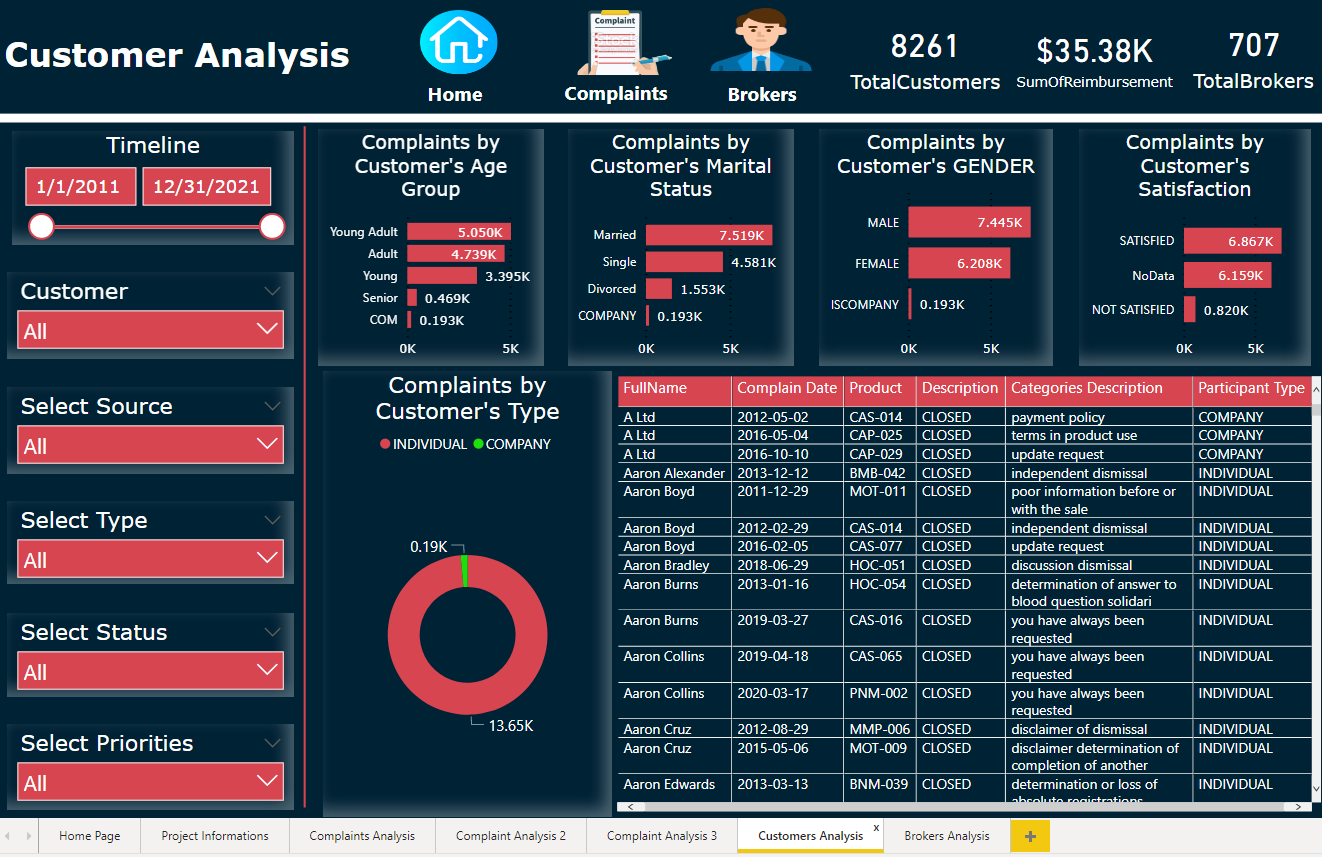
**Page d’accueil**, contient le titre et 4 buttons pour naviguer.

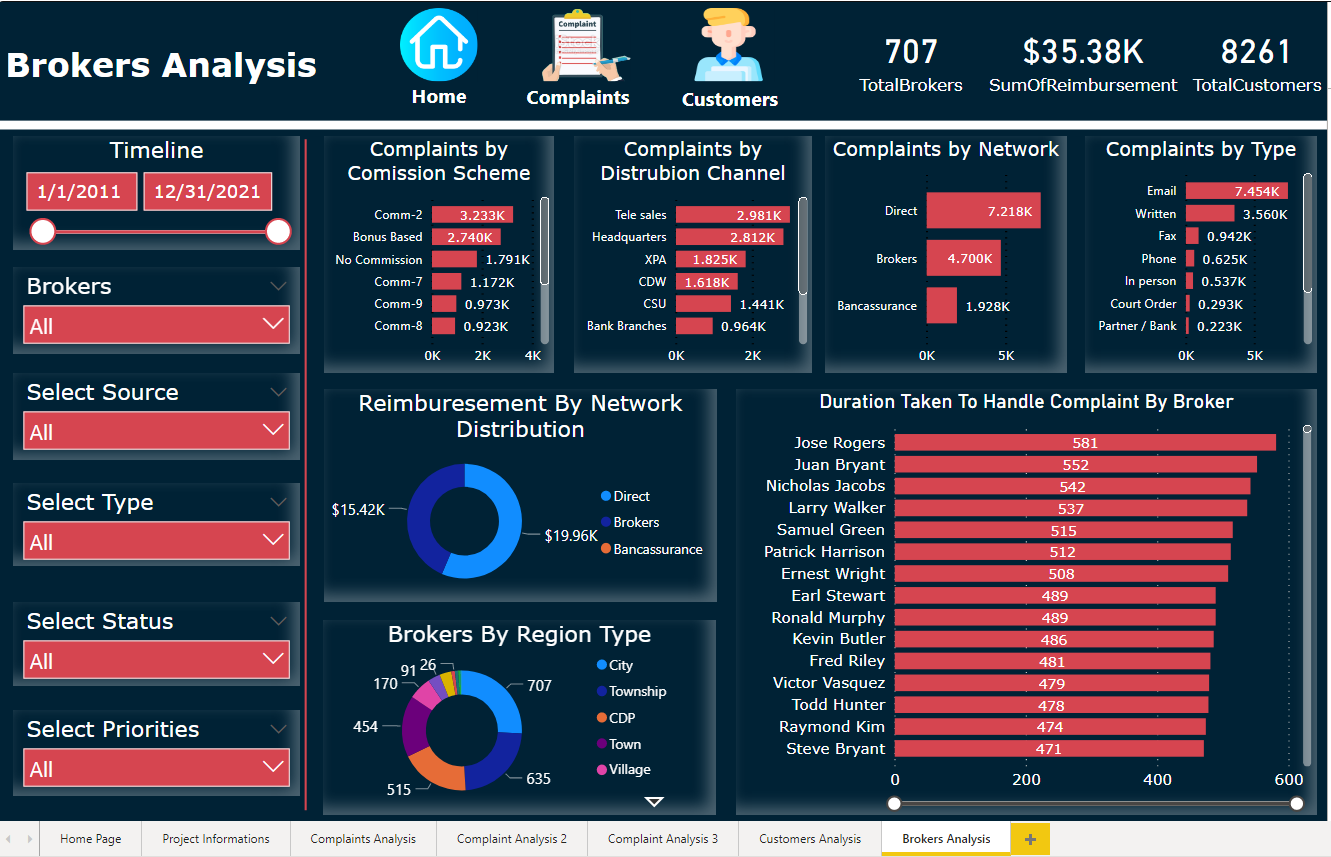
* Page d’informations contient une description du projet





* 3 pages ont été réalisé pour l’analyse des plaintes selon plusieurs critères (Région, Catégories, Types, Priorités, Plage des dates…)



* Analyse des plaintes selon les champs de la dimension des clients.
* Analyse des plaintes selon les champs de la dimension des courtiers.

**REMARQUE IMPORTANTE**

Le fichier power BI qu’on a crée est un fichier ‘connected live’ au cube et la base de données SSAS. C’est-à-dire qu’un autre utilisateur ne peut pas ouvrir le fichier s’il n’est pas connecté au même serveur et puisque le serveur sur lequel est déployé est simplement un ‘localhost’, il est impossible d’ouvrir le fichier sur une autre machine ! Et Malheureusement Microsoft Power BI ne supporte toujours pas le switch du mode Live vers le mode Import comme expliqué ici : [Microsoft IDEA · Support switching SSAS connect model from "Connect Live" to "Import" (powerbi.com)](https://ideas.powerbi.com/ideas/idea/?ideaid=d770b60b-8202-4db8-848d-90450923d0bb)

Donc exceptionnellement pour ce projet, la procédure pour lire le fichier ipbix est d’attacher les fichiers base de données (mdf et ldf) sur SSMS et ouvrir le projet SSAS sur Visual Studio et connecter le projet au serveur localhost de la machine et redéployer le cube. En attendant que Microsoft développe et trouve une solution rapide à ce problème et pour ne pas refaire les tableaux de bord à 0, ceci est la seule solution disponible pour l’instant.

# Conclusion

Dans le but de découvrir le monde d’analyses des plaintes d’une assurance, le choix de données est très important pour bien extraire des informations pertinentes en se basant sur des outils de stockage (SQL Server), analyse (SSIS-SSAS) et Reporting (Power BI).

Au but de jouer avec les diagrammes, cartes, filtres sur un tableau de bord qui résume et facilite la lecture de notre rapport. C’est pour cela on a passé par plusieurs étapes :

* Importation de données
* Création d’ETL
* Création de cube multidimensionnels.

Dans les modèles multidimensionnels, il s’agit d’une analyse par cube OLAP plus classique. À partir des tables de faits, il faut construire les différentes mesures et dimensions ou axes qui vont permettre l’analyse multidimensionnelle, créer les hiérarchies….

Il existe aussi les modèles tabulaires 🡺 les tables s’organisent d’une manière équivalente aux bases de données relationnelles. L’avantage d’un modèle tabulaire c’est qu’il est beaucoup moins dépaysant car son interface ressemble beaucoup à Excel et repose principalement sur le langage DAX.