

Université de Caen Normandie

Master 1 Informatique

Bases de données avancées

*Contrôle continu 2 : Modélisation et analyse
multidimensionnelle (27/10/2023)*

Rapport réalisé par :

Nom : MESSILI

Prénom : Islem

Numéro Etudiant : 22303045

Année : 2023/2024

Nom : KHABBAZ

Prénom : Anas

Numéro Etudiant : 22302264

Année : 2023/2024

I) Données fournies :

Les données fournies dans ce TP sont :

- La base de données « operational_data.db » : Contient deux tables « Ventes » et « Clients », elle sert à alimenter la table dimension « DIMCLIENTS » et la table des faits « FAIT_VENETS »
- Le fichier « prestations.csv » : Contient des informations sur les prestations, il sert à alimenter la table de dimension « DIMPRESTATIONS »
- Le fichier « dates.csv » : Contient toutes les dates comprises de 2019 à 2021, il sert à alimenter la table de dimension « DIMDATES »
- Le fichier « geography.csv » : Contient des adresses, il sert à alimenter la table dimension « DIMLIEUX »

II) Attendus :

Après avoir construit notre Data Warehouse, on peut avec des requêtes MDX et en utilisant les schémas Mondrian, savoir au moins au moins les axes suivants :

- Prix de vente, chiffre d'affaires, nombre de ventes.
- Durée moyenne d'intervention, durée maximum d'intervention.
- Analyse par région, code postal, département, etc., du lieu de réalisation des prestations (avec la ville comme plus petite granularité).
- Analyse selon les clients, leur ville, codes postaux, départements, etc.
- Analyse par heure de la journée, par jour de la semaine, du mois, par trimestre, etc.
- Analyse par type et catégorie de prestations.

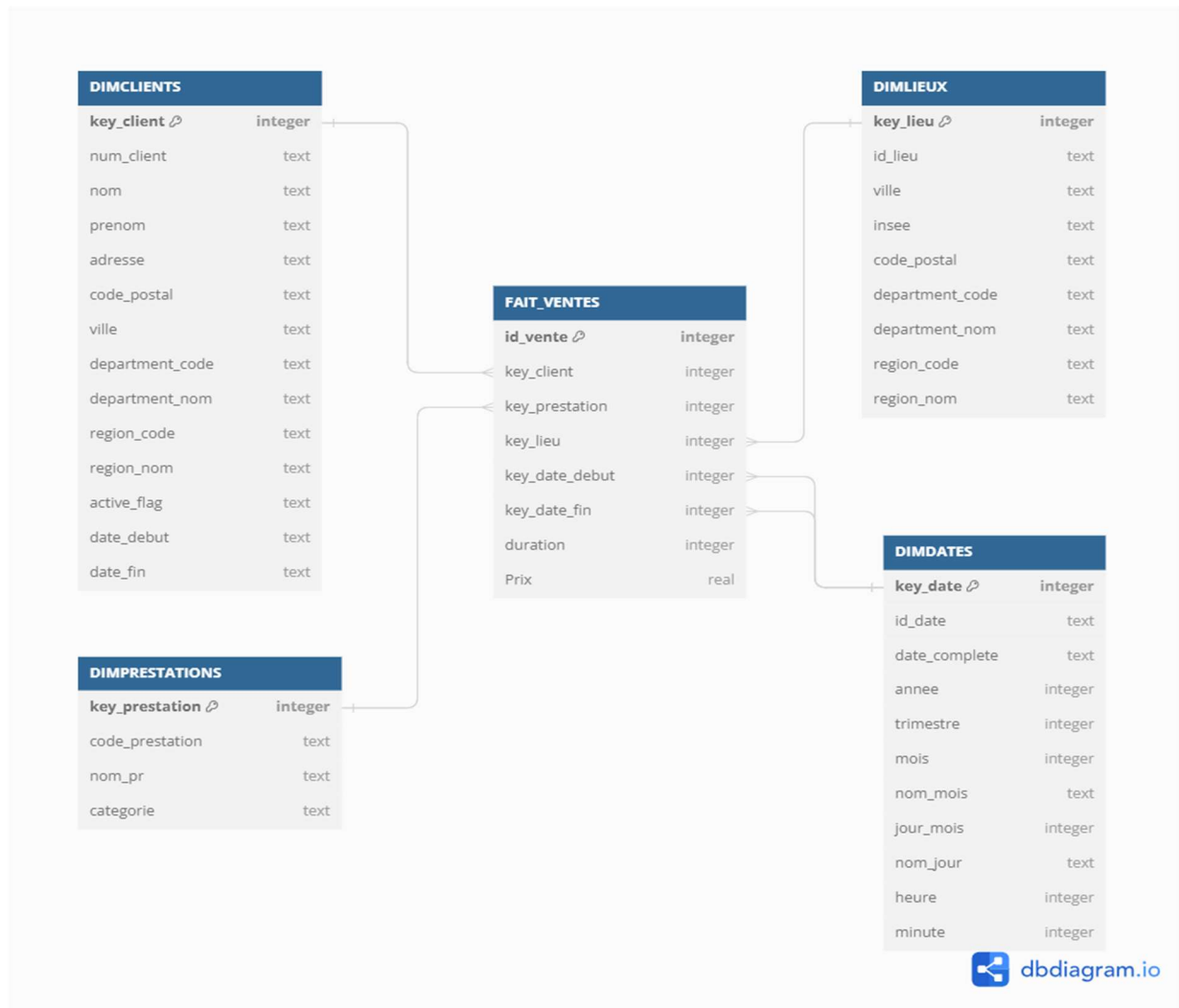
III) Création d'un entrepôt de données :

Question 1. Une modélisation en étoile des données pour notre Data Warehouse :

Le modèle en étoile, avec une table des faits « FIAT_VENTES » contenant les clés des tables dimensions et les mesures comme le prix et la durée de réalisation des prestations, ainsi que les tables de dimensions « DIMCLIENTS », « DIMPRESTATIONS », « DIMDATES », « DIMLIEUX »

Cette modélisation permet une conception simplifiée et une performance plus rapide grâce aux requêtes multidimensionnelles MDX.

Question 2. Le schéma relationnel pour notre entrepôt de données :



Pour que les données soient résilientes aux changements, on considère que la dimension clients est une dimension à évolution lente (**Slowly Changing Dimension**) et on ajoute les attributs « **active_flag** », « **date_debut** » et « **date_fin** » qui vont être mis à jour quand un client change son adresse. Cela permet de suivre l'évolution des adresses des clients au fil du temps.

- **Active_flag** : Cet attribut indique si l'adresse actuelle du client est la plus récente.
- **Date_debut** : Cet attribut représente la date à laquelle l'adresse actuelle du client a été mise en service.
- **Date_fin** : Cet attribut représente la date à laquelle l'adresse actuelle du client cesse d'être active.

Pour permettre la traçabilité des données, on utilise pour chaque table une clef de substitution (surrogate key).

Exercice 3. Les scripts SQL permettant de créer les tables correspondant à la modélisation en étoile dans une base de données SQLite, sont dans le fichier **SQLqueries.txt**

Notre entrepôt de données va être stocker dans une base de donnée SQLITE, pour la crée en lance la commande suivante :

\$ sqlite3 dataWareHouse.db

IV) Alimentation de l'entrepôt :

Exercice 4. Création d'un job Talend pour alimenter la dimension correspondant aux dates, et lieux de réalisation des prestations à partir des fichiers « dates.csv » et « geography.csv ».

- Le job Dim_Dates
- Le job Dim_Lieux

Exercice 5. Création d'un job Talend pour alimenter les autres dimensions du datawarehouse (les deux dates de validité sont à « null »)

- Pour alimenter la dimension « DIMCLIENTS », il faut lancer le job Dim_Client_Insert qui alimente la dimension quand elle est vide.
- Le job Dim_Client_Update met l'attribut « **active_flag** » a « Non » ce qui indique que l'adresse n'est plus valide.
- Le job Dim_Client_InsertUpdated insert le client avec sa nouvelle adresse mais sans le supprimer en lui donnant une nouvelle clef de substitution et avec un attribut « **active_flag** » met à « Oui ».

Exercice 6. Création d'un job Talend permettant d'alimenter la table de faits, à partir des données de la table Ventes.

- Le job Fait_Ventes

Le projet Talend est disponible dans un fichier .zip joint.