



Rapport d'analyse sur les solutions et les préconisations d'utilisation de la base de données dans un outil de datavisualisation

Date : 9 août 2024

1. Introduction

1.1 Objectif du rapport

À la suite de la première mission réussie sur le nettoyage des données, Florian, le pdg de Bottleneck, souhaite aller plus loin et lance un projet de mise à disposition des données de l'entreprise pour ses équipes.

L'objectif de ce rapport est d'analyser et de valider une solution parmi 3 possibilités pour utiliser les données dans des outils de data visualisation :

- Une connexion directe à la base de données
- Une extraction des données en CSV de la base de données
- Une utilisation d'un ETL (Extraction, Transform, Load) ou Python.

1.2 Contexte de l'analyse

Dans un secteur très concurrentiel, Bottleneck doit trouver des outils pour permettre à ses équipes d'être plus réactifs et plus efficaces sur le plan commercial.

Ce rapport doit préconiser la meilleure solution possible et la plus adaptée pour Bottleneck.

2. État des lieux et évaluation de la pertinence

2.1 Description de la situation actuelle

Bottleneck est très en retard sur l'utilisation des données.

Cela risque d'impacter à terme sa compétitivité, son organisation et sa croissance s'il ne met pas en place des plans d'action pour rectifier le tir rapidement.

La mission doit choisir une méthode pour exploiter des données à partir d'une base de données SQLite en utilisant Power BI comme outil de datavisualisation.

2.2 Évaluation de la pertinence des données existantes

Les données existantes semblent moins fiables et ne permettent pas d'étude poussée de la clientèle.

Les données de stock comportent des erreurs qui engendrent des problèmes de réapprovisionnement.

De plus, l'entreprise a du mal à identifier les segments en forte croissance et rentables, faute de données fiables.

Ces données doivent être exploitables, pertinentes et de qualité pour être utilisées dans Power BI.

2.3 Identification des principaux problèmes ou enjeux

Les principaux problèmes sont :

- Sa compétitivité : leurs concurrents sont en avance et font des études plus poussées de la clientèle (ciblage, offres, promotion, etc.).
- Son organisation : la gestion de stocks et de réapprovisionnement est le nerf de la guerre. Mais si les données ne sont pas fiables, toute la chaîne de ventes peut être défaillante.
- Sa croissance : ne pas pouvoir quantifier les relais de croissance sur les nouveaux segments de marché (choisir les bons produits, les bons prix, les bonnes marges, etc.), ce qui pénalise la rentabilité de l'entreprise.

Pour fiabiliser et sécuriser les données, Bottleneck a fait appel à un consultant externe pour faire des préconisations.

À la suite des préconisations, Bottleneck a lancé un chantier de refonte de ses bases de données en interne afin de simplifier et de fiabiliser ses données.

Ce chantier est maintenant terminé.

L'ensemble des données est propre et est au sein d'une même base de données sur SQLite.

3. Besoin d'outils

3.1 Identification des outils nécessaires pour collecter les données

La collecte des données implique l'extraction des données brutes de la base de données SQLite pour les préparer à l'analyse et à la visualisation.

3.1.1 Accès direct :

➤ ODBC Driver pour connecter SQLite à Power BI.

Un pilote ODBC (Open Database Connectivity en anglais) permet à Power BI de se connecter directement à la base de données SQLite.

Pour cela, il faut installer et configurer le pilote ODBC pour permettre à Power BI d'accéder aux données de la base de données SQLite.

3.1.2 Extraction des tables en CSV :

➤ SQLite CLI (Command Line Interface) :

Un outil en ligne de commande pour interroger et exporter des données de la base de données SQLite est nécessaire.

On utilisera des commandes SQL pour exporter les tables de la base de données SQLite en fichiers CSV.

On peut également se connecter à SQLite via ODBC et importer des fichiers CSV dans Power BI.

3.1.3 Fichier "à plat" :

➤ Python:

Pour collecter les données en utilisant un ETL, via Python, qui est un outil puissant pour la collecte de données en raison de sa flexibilité et de son vaste écosystème de bibliothèque, on pourra utiliser des bibliothèques Python comme :

- ✓ « Sqlite3 » : pour interagir avec des bases de données SQLite.
- ✓ « Pandas » : pour charger les fichiers CSV.

3.2 Identification des outils nécessaires pour traiter les données

3.2.1 Accès direct :

SQLite Command Line Interface (CLI) permet de créer, gérer, et interroger les bases de données SQLite directement depuis une interface de commande.

On utilisera des requêtes SQL pour traiter et transformer les données directement dans la base de données SQLite, mais également pour des opérations comme le nettoyage des données, la conversion des types de données, les agrégations, etc.

3.2.2 Extraction des tables en CSV :

Pour traiter les données extraites de la base de données SQLite sous forme de fichiers CSV, on pourra utiliser Power Query, qui est un outil intégré dans Power BI pour la transformation et le chargement des données, le nettoyage, la conversion des types de données, la fusion de tables, etc.

On prépare des données pour l'analyse et la visualisation dans Power BI.

3.2.3 Fichier "à plat" :

Après avoir chargé les fichiers CSV, on utilisera Pandas pour nettoyer et effectuer des transformations avancées sur les données, telles que le nettoyage, la gestion des valeurs manquantes, les agrégations, etc.

Ensuite, on exportera les données transformées sous forme de nouveaux fichiers CSV dans Power BI.

3.3 Identification des outils nécessaires pour analyser les données

L'analyse des données implique l'exploration des données traitées pour en extraire des informations et générer des rapports. Voici les outils nécessaires :

3.3.1 Accès direct :

Power BI Desktop est l'outil principal pour la création de rapports et de tableaux de bord interactifs.

Il permet d'importer des données de diverses sources, y compris SQLite, et de créer des visualisations interactives.

Après avoir configuré une connexion directe à la base de données SQLite via ODBC pour importer et analyser les données directement dans Power BI, nous utiliserons les fonctionnalités de Power BI pour analyser et visualiser les données.

3.3.2 Extraction des tables en CSV :

Après avoir utilisé Power Query pour nettoyer, transformer et préparer les données pour l'analyse, on utilisera les fonctionnalités de Power BI pour créer des visualisations interactives et des tableaux de bord basés sur les données CSV.

On utilisera DAX (Data Analysis Expressions) qui est un langage de formule pour Power BI, permettant de créer des calculs personnalisés et des mesures avancées dans les rapports.

3.3.3 Fichier « à plat » :

Power BI permet l'intégration de scripts Python pour des analyses avancées.

Cela peut être utile pour effectuer des analyses statistiques ou des transformations de données complexes directement dans Power BI.

On utilisera pandas pour effectuer une analyse exploratoire des données, comme calculer des statistiques descriptives et visualiser les données.

4. Solutions d'extraction, de traitement et de visualisation

4.1 Solutions proposées pour l'extraction des données

4.1.1 Accès direct :

SQLite CLI exécutera des requêtes SQL pour l'exportation directe des données.

4.1.2 Extraction des tables en CSV :

On extraira les tables individuellement en fichiers CSV pour une utilisation flexible.

On se connectera à SQLite via ODBC et importer des fichiers CSV bruts dans Power BI.

4.1.3 Fichier "à plat" :

On utilisera Python pour combiner les CSV en un seul fichier "à plat".

4.2 Solutions proposées pour le traitement des données

4.2.1 Accès direct :

On utilisera des requêtes SQL complexes et avancées pour transformer les données directement dans la base de données SQLite avant de les importer dans Power BI.

On pourra importer ces vues dans Power BI comme une table.

4.2.2 Extraction des tables en CSV :

Dans Power Query, on appliquera des transformations telles que la modification des types de données, la division de colonnes, la fusion de tables, et l'ajout de colonnes calculées.

On peut effectuer des agrégations de données (sommations, moyennes, etc.) directement dans Power Query avant de charger les données dans le modèle de données de Power BI.

4.2.3 Fichier "à plat" :

Python est un outil puissant pour le traitement des données avant de les importer dans Power BI. On utilisera des bibliothèques comme pandas pour effectuer des transformations complexes.

4.3 Solutions proposées pour la visualisation des données

4.3.1 Accès direct :

Power BI ne prend pas en charge directement les bases de données SQLite.

Cependant, on utilisera un pilote ODBC pour établir une connexion et importer les tables.

On utilisera Power BI Desktop pour l'utilisation de graphiques standard de Power BI (tableaux, graphiques à barres, graphiques en courbes, cartes géographiques, etc.) pour visualiser les données.

On créera des dashboards interactifs permettant aux utilisateurs de filtrer et de manipuler les données en temps réel.

Utilisation des fonctionnalités avancées de Power BI

On utilisera DAX (Data Analysis Expressions) pour créer des mesures et des colonnes calculées afin d'ajouter des calculs complexes à vos visualisations.

Une mesure calculée n'est pas stockée physiquement, et n'est évaluée (calculée) que si elle est utilisée. Elle n'utilise donc pas d'espace ou de RAM, mais elle consomme de la CPU.

Créer une mesure plutôt qu'une colonne présente donc trois avantages significatifs : la formule n'étant calculée qu'au moment de son utilisation, le temps d'actualisation est plus court et le poids du fichier n'augmente pas.

4.3.2 Extraction des tables en CSV :

Après avoir transformé les données dans Power Query, on utilisera Power BI Desktop pour l'utilisation de graphiques standard de Power BI (tableaux, graphiques à barres, graphiques en courbe ou en anneau, jauges, etc.) pour visualiser les données.

On créera des dashboards interactifs permettant aux utilisateurs de filtrer et de manipuler les données en temps réel.

4.3.3 Fichier "à plat" :

On intégrera des scripts Python directement dans Power BI pour des analyses et des visualisations avancées.

Cohérence des solutions avec le besoin

5.1 Explication détaillée de chaque solution proposée

5.1.1 Accès direct :

Simple et en temps réel, cette solution nécessite un connecteur ODBC et la configuration du schéma en étoile dans Power BI.

5.1.2 Extraction des tables en CSV :

Flexible, cette solution permet de modifier les données avant importation et nécessite une étape supplémentaire d'extraction.

5.1.3 Fichier "à plat" :

Cette solution simplifie la gestion des données, adaptée pour des bases moins complexes, mais peut poser des problèmes de performance avec des volumes de données élevés.

5.2 Alignement des solutions avec les besoins identifiés

5.2.1 Accès direct :

Cette solution est adaptée pour des besoins de données en temps réel et une structure de base de données bien définie.

5.2.2 Extraction en CSV :

Cette solution convient aux besoins de flexibilité et de modification des données avant l'analyse.

5.2.3 Fichier "à plat" :

Cette solution est adaptée si la simplicité est primordiale et que la base de données n'est pas trop complexe.

5.3 Avantages et limites de chaque approche

5.3.1 Accès direct :

Avantages :

Connexion simple et directe aux données.

Accès en temps réel à la base de données.

Pas de besoin de processus d'extraction supplémentaires.

On peut effectuer des modifications directement dans la base de données SQL.

Inconvénients :

Nécessite l'ajout d'un connecteur ODBC pour utiliser SQLite avec Power BI.
Nécessité de refaire le schéma de la base de données dans Power BI pour créer un modèle en étoile.

Les visualisations récupèrent les données directement depuis la base de données, mais elles ne sont pas modifiables dans Power BI.

La capacité de la source doit être suffisamment puissante pour accepter une connexion directe.

5.3.2 Extraction en CSV

Avantages :

Toutes les tables peuvent être extraites dans des fichiers CSV séparés.

Permet la modification des données en dehors de Power BI.

Flexibilité d'utilisation des données dans d'autres outils avant importation dans Power BI.

Inconvénients :

Nécessite une étape supplémentaire d'extraction lors de la mise à jour.

Refaire le schéma de la base de données dans Power BI pour un modèle en étoile.

Le poste sur lequel est conçu et utilisé le rapport doit avoir une puissance suffisante, en terme de RAM (pour le stockage des données) et de CPU (pour les calculs).

5.3.3 Fichier "à plat" :

Avantages :

Simplifie la gestion des données en n'ayant qu'un seul fichier à manipuler.
Pas besoin de gérer le schéma en étoile ou de se connecter à la base de données.

Importer le fichier "à plat" dans Power BI.

Inconvénients :

Non optimal pour les bases de données complexes avec plusieurs relations.
Peut entraîner des problèmes de performance avec des fichiers très volumineux.

❖ Conclusion

Synthèse des principales conclusions de l'analyse

Chaque méthode d'exploitation des données présente des avantages et des limites. L'accès direct offre une solution simple et en temps réel, mais nécessite une configuration technique. L'extraction en CSV offre une flexibilité maximale, mais ajoute une étape d'extraction. La création d'un fichier "à plat" simplifie la gestion des données, mais peut poser des problèmes de performance.

Recommandations pour la mise en œuvre des solutions proposées

Accès direct :

Cette solution est recommandée si Bottleneck peut gérer la configuration technique et a besoin d'un accès en temps réel.

Dans ce cas, les données restent dans la base, qui est interrogée à chaque manipulation dans le fichier au prix d'un effort important sur la base. Une opération aussi simple que filtrer un graphique amène à réinterroger la source.

Les performances dépendent des performances du système de base de données.

Pour la lourdeur et le coût de l'accès direct, nous ne retiendrons pas cette solution, d'autant plus que les équipes n'ont pas forcément besoin de chiffres en temps réel. Une à deux actualisations quotidiennes suffiraient.

Extraction en CSV :

Cette solution est recommandée pour une flexibilité maximale et la possibilité de modifier les données avant l'analyse.

L'application sera plus performante si les données sont rapatriées localement (mode Import) : la source de données n'est interrogée qu'une fois, lors de l'exécution ou de l'actualisation de la requête, puis les données sont stockées localement ou à l'intérieur du fichier.

Ceci accélère considérablement la mise en place des visuels par la sélection des champs, l'utilisation des filtres et des interactions en général.

Cette solution paraît être la plus recommandable.

Et comme on a vu précédemment, l'actualisation quotidienne est suffisante pour la direction et les équipes commerciales.

L'investissement financier est un critère important à prendre en compte dans tout projet et si on peut minimiser les coûts tout en atteignant les objectifs de la direction, il faut privilégier cette solution, quitte à revoir la solution retenue si elle se révèle insuffisante dans la pratique.

Fichier "à plat" :

Cette solution est recommandée pour des projets moins complexes où la simplicité est essentielle.

Le projet est assez complexe avec des données et des problématiques importantes.

La deuxième solution paraît préférable à la troisième.

❖ Annexes

Liste des sources de données utilisées

Base de données SQLite : la source principale contenant les tables de données nécessaires à l'analyse.

Fichiers CSV : résulte de l'extraction des tables de la base de données SQLite pour une manipulation flexible.

Fichier "à plat" : créé à partir de la combinaison des fichiers CSV, pour simplifier la gestion des données dans Power BI.

Description détaillée des outils recommandés

SQLite ODBC Driver

Le pilote ODBC pour SQLite permet à des applications comme Power BI de se connecter directement à une base de données SQLite. Ce pilote traduit les commandes ODBC en commandes SQLite.

Caractéristiques

Compatibilité : Fonctionne avec toutes les versions de SQLite.

Installation simple : Disponible pour Windows, MacOS, et Linux.

Sécurité : Support des connexions sécurisées.

Utilisation

Télécharger et installer le pilote : Disponible sur [SQLite ODBC Driver](#).

Configurer une source de données ODBC :

Ouvrir le Gestionnaire de sources de données ODBC.

Ajouter une nouvelle source en sélectionnant le pilote SQLite.

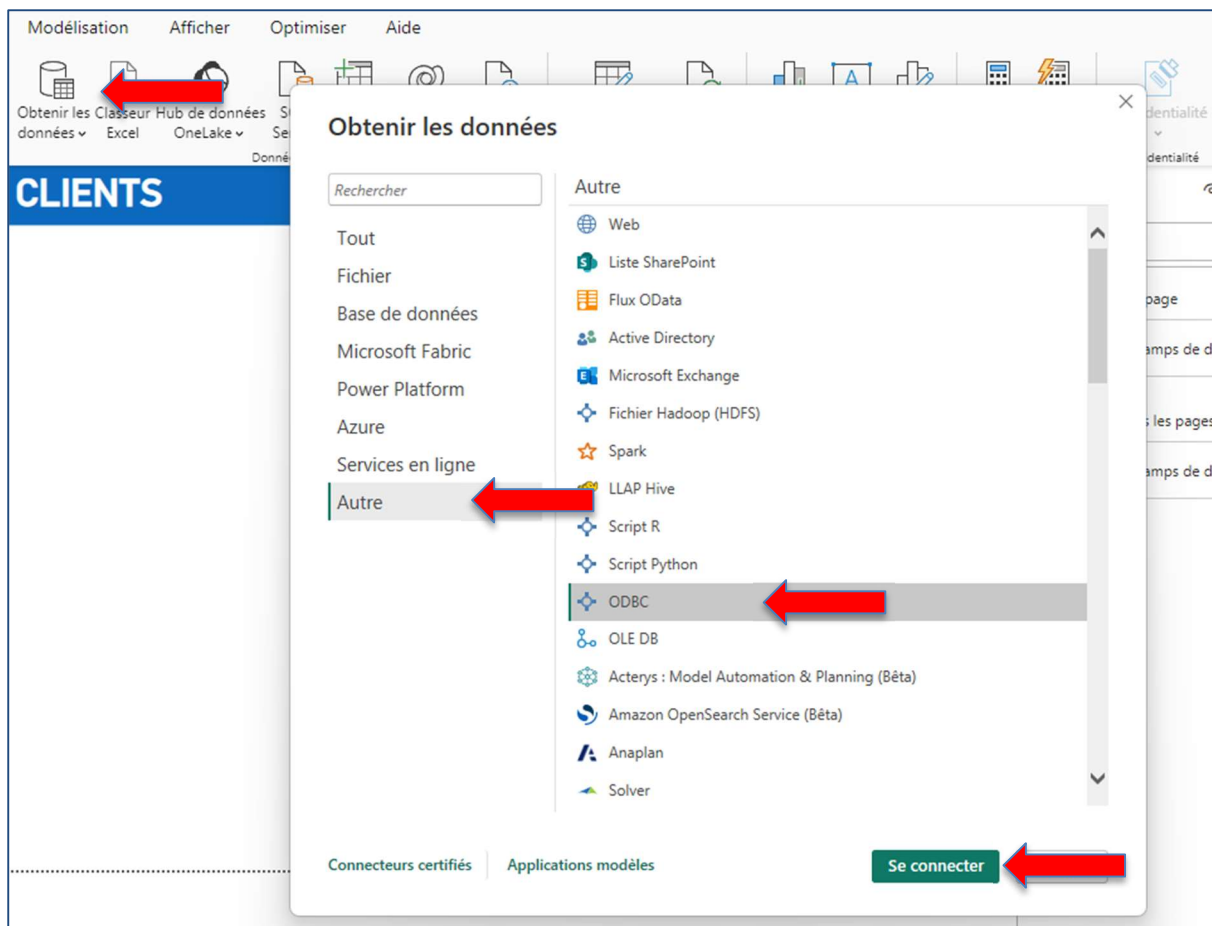
Spécifier le chemin de la base de données SQLite.

Connexion à une source de données :

Power BI se connectera à des sources de données ODBC pour importer et analyser les données.

En configurant une source de données ODBC pour SQLite, Power BI peut directement interroger la base de données et visualiser les données.

Dans **Power BI Desktop**, on cliquera sur « Obtenir des données » et sélectionnera la source de données (ODBC pour SQLite) pour importer les tables de SQLite.



Ensuite on utilisera l'éditeur **Power Query** :

On cliquera sur « Transformer les données » pour ouvrir l'éditeur Power Query.

On appliquera les transformations nécessaires comme le filtrage, l'agrégation, la fusion, etc.

Une fois les transformations appliquées, on chargera les données dans Power BI pour créer des visualisations.

Il est également possible de mettre en place des modèles composites, dont une partie des tables (ou des sources) sont exploitées en mode Import, et les autres en mode Direct.

Par exemple, des tables Produits ou Sites géographiques, qui évoluent peu, peuvent être importées, mais les tables des transactions, ou des chaînes de production, qui ont parfois besoin d'être interrogées en quasi-temps réel, peuvent rester en mode direct.

L'actualisation dans Power BI Desktop se fait à l'aide de l'outil Actualiser.

C'est dans le cadre de la publication sur Power BI Service que les possibilités d'actualisation sont les plus complètes.

❖ **Exemples de visualisations de données proposées**

✓ Principaux KPI :

Un tableau de bord interactif affichant les principaux indicateurs de gestion : segments, graphiques en secteurs/en anneau et histogramme empilé, histogramme empilé (CA, marge, stock, panier moyen, nombre d'articles vendus...).

✓ Top & Flop des ventes hors-promo :

Un rapport détaillé sur les meilleures ventes hors promo mais également les références les moins vendues par segment et par type de produit : segments, tableau simple et graphique à barres groupées.

✓ Top & Flop des ventes promo :

Comme pour les ventes hors-promo, un rapport détaillé sur les meilleures ventes promo mais également les références les moins vendues par segment et par type de produit : segments, tableau simple et graphique à barres groupées.

✓ Focus sur la gestion :

Un rapport avec les indicateurs clés de performance financière et de rentabilité : marge en valeur, taux de marge, quantités vendues et achats fournisseurs (total, hors-promo, promo) : segments, tableau simple et graphiques à barres groupées

✓ Analyse du stock :

Un rapport avec les indicateurs clés du stock (rotation, valeur, répartition et les plus grosses valeurs de stock par segment et par type de produit : segments, cartes, graphique en anneau et graphique à barres groupées.

✓ Feed-back clients

Un rapport sur les appréciations des clients (note moyenne et par secteur, références les plus notées) : segments, jauge, histogramme empilé et graphique à barres groupées.

✓ Evolution indicateurs

Un rapport pour visualiser l'évolution sur 12 mois, du prix moyen de vente hors-tax, du prix moyen d'achat hors-tax et de la marge moyenne : segments, cartes et graphique en aires.

✓ Hausse tarifaire

Un rapport pour mesurer l'inflation, la hausse des prix d'achat globale et par segment, les plus fortes hausses tarifaires en comparaison avec la hausse du prix de vente de la référence : segments, jauge, histogramme empilé et graphique à barres groupées.

✓ Recommandations business

Des recommandations business pour améliorer les performances futures de Bottleneck.