

|  |
| --- |
| Bottleneck |
| Rapport d’analyse technique |
| Pour la mise à disposition des données dans un outil de data visualisation |

Liodenot julien

01/07/2025

Table des matières

[1. Introduction 3](#_Toc205733403)

[Objectif du rapport 3](#_Toc205733404)

[Contexte de l'analyse 3](#_Toc205733405)

[2. État des lieux et évaluation de la pertinence 3](#_Toc205733406)

[2.1 Description de la situation actuelle 3](#_Toc205733407)

[2.2 Évaluation de la pertinence des données existantes 3](#_Toc205733408)

[2.3 Identification des principaux problèmes ou enjeux 3](#_Toc205733409)

[3. Besoin d'outils 4](#_Toc205733410)

[3.1 Outils pour collecter les données 4](#_Toc205733411)

[3.2 Outils pour traiter les données 4](#_Toc205733412)

[3.3 Outils pour analyser les données 4](#_Toc205733413)

[4. Solutions d'extraction, de traitement et de visualisation 4](#_Toc205733414)

[4.1 Solutions proposées pour l'extraction des données 4](#_Toc205733415)

[4.2 Solutions proposées pour le traitement des données 5](#_Toc205733416)

[4.3 Solutions proposées pour la visualisation des données 6](#_Toc205733417)

[5. Cohérence des solutions avec le besoin 6](#_Toc205733418)

[5.1 Explication détaillée de chaque solution proposée 6](#_Toc205733419)

[5.2 Alignement des solutions avec les besoins identifiés 7](#_Toc205733420)

[5.3 Avantages et limites de chaque approche 7](#_Toc205733421)

[Conclusion 8](#_Toc205733422)

# 1. Introduction

## Objectif du rapport

Nous allons analyser et comparer différentes approches d’alimentation des données d’une base SQLite locale dans un outil de visualisation de type Dashboard.

L’objectif est d’identifier la solution la plus adaptée à l’exploitation et la restitution des données de la société Bottleneck en tenant compte des besoins et des contraintes techniques.

## Contexte de l'analyse

L’entreprise Bottleneck cherche à mettre en place une solution de datavisualisation de leurs données à l’intention du PDG et des chefs produit. Il s’agit de démontrer la préparation des données, la création d’un dashboard et formuler des recommandations business.

Il faudra définir au moins trois KPI pour l’entreprise et créer des visualisations qui permettent de les suivre dans le temps (marge en valeur, prix HT, rotation des stocks, etc.).

# 2. État des lieux et évaluation de la pertinence

## 2.1 Description de la situation actuelle

• Base SQLite relationnelle contenant les ventes en ligne de vins (produits, clients, ventes, stocks, etc.).  
• Dictionnaire de données complet décrivant chaque champ et relation.  
• Données récentes, déjà nettoyées/retravaillées lors d’un précédent projet.

## 2.2 Évaluation de la pertinence des données existantes

Les tables couvrent l’ensemble du périmètre analytique (ventes, produits, stocks). Les clés primaires et étrangères garantissent une cohérence relationnelle. Aucune donnée hors période cible. Les attributs nécessaires aux KPI métier sont disponibles (dates, canaux, catégories).

## 2.3 Identification des principaux problèmes ou enjeux

• Organisationnel : Problème de suivi de stock, stocks faux et problèmes de réapprovisionnement

• Compétitivité : fiabilité des données et ciblage clientèle pour les offre et promotions

• Croissance : déterminer les segments de marchés porteurs et déterminer les bons produits, prix, marges, etc… de ces segments.

• Besoin d’automatiser le rafraîchissement des données pour éviter les manipulations manuelles.  
• Préserver la structure relationnelle dans Power BI sans dupliquer inutilement les données.  
• Garantir des transformations reproductibles et documentées pour assurer la qualité des indicateurs.

# 3. Besoin d'outils

## 3.1 Outils pour collecter les données

Il existe de nombreuses solutions pour collecter les données :

•Une connexion directe à la base de données au lancement du tableau de bord

•Exporter en CSV depuis SQLITE avec une requête SQL de mise en forme puis lecture du csv comme source de données depuis l’outil de traitement des données

• Pilote ODBC SQLite ou connecteur natif pour connexion directe depuis l’outil de traitement des données (ETL)

## 3.2 Outils pour traiter les données

• Éditeur Power Query intégré à Power BI ou Excel   
• utilisation d’un ETL (Knime,Talend), utilisation de workflow automatisables.  
• Scripts SQL/python ponctuels dans SQLite ou dans un datawarehouse pour extraction des données transformées.

## 3.3 Outils pour analyser les données

•**Power BI Desktop** pour modéliser, créer des mesures DAX et construire le tableau de bord.

• **Tableau Desktop** pour explorer, bâtir des vues interactives et publier sur Tableau Server/Cloud.

• **Excel Power Pivot** pour importer et modéliser des jeux de données volumineux, créer des mesures DAX et les analyser dans des tableaux croisés dynamiques.

•**Python + Plotly Dash** pour coder des applications web analytiques sur-mesure, avec graphiques interactifs et mises à jour en temps réel via callbacks.

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

## 4.1 Solutions proposées pour l'extraction des données

**• Connexion directe** :

* **Description :** Le Dashboard se connecte directement à la base de données de production au lancement, permettant une actualisation directe des données
* **Pertinence :** adaptée pour une base peut sollicitée, des études de performance devront tout de même être réalisées.

**• Extraction CSV** :

* **Description :** Cette méthode consiste à exécuter des requêtes SQL directement sur la base SQLite (via un outil comme DB Browser for SQLite) et à exporter les résultats sous forme de fichiers plats (CSV). Ces fichiers sont ensuite utilisés comme source de données dans l'outil de traitement ou de visualisation.
* **Pertinence :** Adaptée pour une analyse ponctuelle ou une preuve de concept (PoC), mais inadaptée à un besoin de reporting récurrent en raison de son caractère manuel.

**• Connecteur ETL** :

* **Description :** L'outil ETL de traitement des données se connecte directement à la base de données SQLite en utilisant soit son connecteur natif, soit un pilote ODBC (Open Database Connectivity). Cette connexion permet d'interroger la base et de planifier des actualisations de données de manière intégrée.
* **Pertinence :** Solution simple et efficace pour des besoins où les transformations de données peuvent être gérées directement dans l’outil de data visualisation ensuite.

## 4.2 Solutions proposées pour le traitement des données

**•** **Traitement dans Power Query**: l'éditeur Power Query devient l'outil de traitement principal. Il permet d'effectuer des jointures, des filtrages, et de créer des colonnes calculées. Bien que puissant, il lie la logique de transformation au rapport lui-même et peut donc prendre de la ressource et du temps à l’actualisation.

**•** **Traitement via un ETL** : En utilisant l'approche ETL , tout le traitement lourd est externalisé dans KNIME. Le workflow graphique de KNIME permet de construire une chaîne de traitement documentée et réutilisable, assurant une qualité et une cohérence maximales des données avant même leur chargement dans l'outil de visualisation.

**• Traitement via scripts SQL/Python** : des requêtes SQL ciblées (SQLite ou entrepôt de données) combinées à des scripts Python exécutent les transformations ponctuelles ou complexes qui dépassent les capacités visuelles de Power Query ou de l’ETL ; cette approche offre une flexibilité maximale, mais exige une gouvernance stricte du code (versioning, revue, documentation) pour éviter la dérive et garantir la reproductibilité.

## 4.3 Solutions proposées pour la visualisation des données

**•**Power BI Desktop :

Le Dashboard sera mis en forme dans Power BI et intégré aux outils Microsoft de l’entreprise. Il sera publié sur le service Power BI. Chaque utilisateur sui le le lancera aura des droits adaptés.

**•**Tableau Desktop : Les vues interactives sont construites dans Tableau Desktop et publiées sur Tableau Server/Cloud, ce qui offre un rafraîchissement centralisé, un contrôle d’accès fin (projet, classeur, source) et une gouvernance complète des contenus.

**•**Excel Power Pivot : Les modèles Power Pivot alimentent des tableaux croisés ou graphiques dynamiques directement dans Excel ; le fichier est partagé via SharePoint/OneDrive et se rafraîchit via passerelle, conservant la syntaxe DAX et l’intégration bureautique familière des utilisateurs.

**•**Python + Plotly Dash : Pour les besoins d’applications analytiques sur mesure (workflow complexe, temps réel, simulation), une app Dash conteneurisée est déployée sur serveur interne ; la logique de traitement est en Python, les visuels interactifs en Plotly, l’accès sécurisé par authentification d’entreprise.

# 5. Cohérence des solutions avec le besoin

## 5.1 Explication détaillée de chaque solution proposée

|  |  |  |  |
| --- | --- | --- | --- |
| Scénario | Chaîne technique | Fonctionnement | Points-clefs |
| A – Connexion directe | SQLite → Power BI (ODBC) | Power BI lit la base en temps réel, transformations dans Power Query, refresh programmé. | Implantation express ; zéro intermédiation ; dépendance forte à Power Query ; charge de calcul au rafraîchissement. |
| B – Workflow ETL | SQLite → KNIME → Power BI | KNIME extrait, nettoie, enrichit ; génère fichiers propres ou pousse dans un dataset Power BI ; refresh orchestré côté KNIME. | Traçabilité, versioning, reprise sur incident ; transformations lourdes hors BI ; extensible à d’autres sources. |
| C – Export manuel | SQLite → CSV →  Power BI | Opérateur exécute la requête, exporte CSV, recharge le rapport. | Simple démonstration ou POC uniquement ; non-automatisé ; risque d’erreur et d’oubli. |
| D – Application Python + Plotly Dash | SQLite → Scripts Python → Dash (Docker/K8s) | Scripts Python (pandas) chargent et transforment les données ; Dash sert une web-app interactive ; actualisation via scheduler ou callbacks temps réel. | Personnalisation totale (UI, logique) ; déploiement conteneurisé ; versioning Git. |

## 5.2 Alignement des solutions avec les besoins identifiés

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Besoin clé | A | B | C | D |
| Actualisation automatisée | **Oui** | **Oui** (optimal) | Non | **Oui** (scheduler) |
| Traçabilité des transformations | Faible | **Élevée** | Non | **Élevée** (code) |
| Performance gros volumes | Moyenne | **Élevée** | Moyenne | Moyenne |
| Maintenance & gouvernance | Moyenne | **Élevée** | Faible | Complexe |
| Mise en route rapide | **Élevée** | Moyenne | **Élevée** | Faible |
| Interactions complexes / simulation | Moyenne | Moyenne | Non | **Élevée** |

**Légende :**

* **Oui / Non** → capacité présente ou absente
* **Élevée / Moyenne / Faible** → niveau d’efficacité ou de pertinence

## 5.3 Avantages et limites de chaque approche

Voir tableau comparatif synthétique dans la version longue du rapport (cf. analyse détaillée).

|  |  |  |
| --- | --- | --- |
| Approche | Avantages | Limites |
| Connexion directe | • Déploiement rapide • Moins d’outils à maintenir | • Transformations cachées dans PBIX • Temps de refresh dépendant de la volumétrie |
| Workflow ETL | • Qualité et reproductibilité • Scalabilité • Centralisation des règles métier | • Courbe d’apprentissage KNIME • Coût serveur/maintenance supplémentaire |
| Export manuel | • Zéro licence/infra supplémentaire • POC en quelques minutes | • Non viable en production • Risque humain • Aucune gouvernance |
| Python + Dash | • UI et logique sur-mesure • Temps réel possible • Versioning classique (Git) | • Besoin dev Python/JS • Hébergement et sécurité à gérer • Perf limitée si volumétrie très élevée |

# Conclusion

La **connexion directe** est acceptable pour un prototype ou un besoin sporadique avec faible volumétrie.  
Pour un reporting durable, automatisé et évolutif, le **workflow KNIME + Power BI** s’aligne le mieux sur les exigences de fiabilité, de traçabilité et de performance.  
L’**export manuel** est exclu du périmètre cible et l’utilisation du scénario python est à envisager pour des cas où l’entreprise requiert une application très spécifique, mais nécessite un effort de développement et maintenance plus important.

**Recommandation :** industrialiser le scénario B. Conserver le scénario A comme solution de secours ou de prototypage rapide.