

# Rapport Global

## Introduction

Le rapport constitue une analyse détaillée des différents volets du projet, selon une organisation autour des données non structurées, semi - structurées et structurées. La finalisation de cette analyse est comparative.

## 1. Données Non Structurées

Comparaison des approches :

- Algorithmes de traitement de chaînes : plus simple mais moins flexible.
- Expressions régulières : plus puissantes mais plus lourdes à lire et à maintenir.

## 2. Données Semi-Structurées

Étude comparative des formats JSON, CSV, YAML et XML

Format	Lisibilité humaine	Facilité d'analyse	Supporte la hiérarchie	Poids des fichiers	Usage principal
JSON	Très lisible	Facile	Oui	Léger	Web, API REST
CSV	Peu lisible	Très facile	Non	Très léger	Tableaux de données
YAML	Très lisible	Facile	Oui	Léger	Configurations, DevOps
XML	Moyennement lisible	Complexe	Oui	Lourd	Services web, documents

## 3. Données Structurées

### 3.1. Stockage Relationnel vs Semi-Structuré

Les bases de données relationnelles sont construites sur une structure rigide reposant sur des tables fait de lignes é tag é es dans des colonnes. Celles - ci sont adaptées é es aux données rigoureusement organisées car elles peuvent garantir leur intégrité, permettre des requêtes complexes via SQL, etc . Les données semi-

structurées comme par exemple, en JSON ou XML, ne sont-elles pas construites sur des structures rigides, ce qui facilite leur évolution, et leur analyse peut s'avérer parfois du fait de son absence d'une forme bien précise. Au niveau performance, les bases de données relationnelles sont optimisées pour les données bien définies tandis que les bases semi - structurées devront passer par un traitement préalable pour l'analyse de ses données s'ils sont bien conçus, et bien organisés. Au niveau flexibilité, les données semi-structurées seront plus flexibles dans l'évolution de leurs formes, mais risquent de devenir plus difficiles à manipuler sans une structure solide.

### 3.2. Lien entre la POO et les Bases de Données Relationnelles

En effet, les interactions possibles entre la POO et son module sont les données correspondantes, les bases de données relationnelles, se trouvent de manière singulier au sein de moi d'un outil de conception, le mappage objet - relationnel (ORM) . Cela consiste à faire correspondre les objets du code source de l'application aux tables de la base de données : chaque classe d'objet est une table et chaque instance de la classe est une ligne dans cette table, les attributs de l'objet sont les colonnes de la table. Le traitement d'un code orienté objet est ainsi appliqué par la capacité d'interagir avec des bases de données relationnelles, ce qui rend particulièrement intéressant les applications d'entreprise dans lesquelles les types d'objets sont naturellement organisés en classes.

## 4. Analyse Finale

### 4.1 Comparaison des trois types de données

Type de données	Avantages	Inconvénients
Non structurées	Grande flexibilité, manipulation rapide	Extraction complexe, peu standardisé
Semi-structurées	Bon équilibre entre flexibilité et structuration	Peut nécessiter des conversions
Structurées	Facilité d'interrogation et d'intégrité des données	Manque de flexibilité

L'idée d ' utiliser en parallèle les trois types de représentation d ' information nous paraître la meilleure solution pour une gestion de l'information optimale. Les

informations non structurées conviennent idéalement aux situations où la souplesse est privilégiée et où les données sont fournies au fil de l'eau dans un format libre. Les données semi-structurées sont les plus adaptées lorsque nous devons garder un certain ordonnancement tout en conservant une souplesse dans l'expédition des données, comme celle que l'on trouve dans les configurations ou dans les API. Enfin, le système de gestion des données structurées convient particulièrement et remient pour des bases partageant une rigueur et une rapidité d'accès aux données avec une éprouvée logique d'intégrité et de respect de la normalisation. En fonction du contexte projet, le type de données à utiliser peut faire la différence en termes de rapidité, d'efficacité et de souplesse. La combinaison intelligente des trois types de données permettra d'accéder aux points forts de chacun d'eux pour parvenir à un résultat convenable.

Nous avons utilisé l'IA pour générer les logs