



Ministère de l'Enseignement
Supérieur et de la Recherche
Scientifique

Année Universitaire
2023/2024



Ecole Internationale Supérieure
Privée Polytechnique de Sousse

RAPPORT DE PROJET DE FIN D'ÉTUDES

Pour l'obtention du Diplôme National
d'ingénieur en Informatique

Spécialité : Génie Logiciel

Intitulé

La mise en place d'une solution de
Business Intelligence pour l'application
Thunder Express

Lieu du stage

webify

Réalisé par

Inès Belkhiria

Encadré par

Dr. Adel Dahmen
Ing. Maha Kchaich

Table des matières

Introduction générale	1
1 Présentation générale	3
Introduction	3
1.1 Présentation de la société d'accueil	3
1.2 Présentation du projet	4
1.2.1 Problématique	4
1.2.2 Prospection des solutions existantes	5
1.2.3 Objectifs du projet	8
1.2.4 Processus de développement	9
1.3 Étude théorique	10
1.3.1 Business Intelligence	10
1.3.2 Architecture BI	11
1.3.3 Modélisation BI	12
1.3.4 Méthodologies de entrepôts de données	15
Conclusion	18
2 Spécification des besoins et Planification	19
Introduction	19
2.1 Collecte des exigences	19
2.1.1 Exigences fonctionnelles	19
2.1.2 Définition des indicateurs clés de performance (KPI)	20
2.1.3 Exigences non fonctionnelles	22
2.1.4 Faisabilité technique du projet BI	22
2.2 Spécification des exigences	24
2.2.1 Identification des acteurs	24
2.2.2 Identification des cas d'utilisation	24
2.3 Planification du projet	27
Conclusion	29
3 Conception des données et des applications	30
Introduction	30
3.1 Modélisation dimensionnelle	30
3.1.1 Source de données	30
3.1.2 Description des données	32
3.1.3 Qualité et Accessibilité des données	33

3.2	Conception du Processus ETL	34
3.2.1	Extraction (Extract)	34
3.2.2	Transformation (Transform)	35
3.2.3	Chargement des données (Load)	36
3.2.4	Scripts ETL	37
3.3	Modèle des données	37
3.3.1	Approche Dimensionnelle : Dimensions et Faits	37
3.3.2	Shémas des données	40
3.3.3	Measures DAX	42
3.4	Conception des applications	43
3.4.1	Conception des Rapports et des Tableaux de Bord	43
3.4.2	Développement de prototypes BI	44
3.4.3	Tests de prototypes (Ajustements basés sur les retours des utilisateurs)	45
3.5	Environnement technique	46
3.5.1	Environnement matériel	46
3.5.2	Environnement logiciel	46
	Conclusion	51

Introduction générale

Dans le monde actuel axé sur les données, les entreprises évoluent dans un paysage hautement concurrentiel où les décisions doivent être prises rapidement et de manière stratégique. La capacité de rassembler, analyser et exploiter efficacement les données est devenue un facteur essentiel pour parvenir à une croissance durable et conserver un avantage concurrentiel. Ce défi a conduit à l'émergence de la gestion commerciale en tant que discipline à part entière. Tirant parti d'outils technologiques avancés, les entreprises s'efforcent d'analyser les tendances des ventes, d'évaluer la viabilité des produits et de comprendre le comportement des clients. C'est ici que la Business Intelligence (BI) entre en jeu, offrant des représentations intelligentes d'informations à partir de bases de données grâce à des outils spécialisés.

La Business Intelligence est un outil puissant qui permet aux organisations de débloquent tout le potentiel de leurs données. En fournissant une vision globale des performances, en permettant une prise de décision basée sur les données, en améliorant l'expérience client et en favorisant l'innovation, la BI est devenue indispensable pour les entreprises modernes.

Dans ce contexte, notre projet se concentre sur la conception et le développement d'une solution de Business Intelligence spécifiquement destinée à la gestion des commandes pour l'application de livraison de repas Thunder Express. En exploitant l'entreposage de données et les capacités de reporting avancées, notre projet vise à mieux gérer et structurer les données, permettant à l'entreprise de générer des informations exploitables, d'améliorer l'efficacité et de réaliser une croissance durable dans un marché hautement compétitif.

Le présent rapport s'articule autour de cinq chapitres :

Le premier chapitre, intitulé «**Cadre Général du projet**», commence par une présentation de l'organisme d'accueil. Nous y abordons ensuite l'étude de l'existant et la solution proposée. Nous concluons ce chapitre par une description de la méthodologie de gestion de projet adoptée. Ce chapitre offre également une compréhension approfondie de la solution en introduisant les concepts clés de la Business Intelligence, avant d'entrer dans les détails spécifiques.

Le deuxième chapitre, intitulé «**Analyse et spécification des besoins**», définit les acteurs de notre application et précise les exigences fonctionnelles et non fonctionnelles. Il inclut également un diagramme de cas d'utilisation illustrant les interactions prévues.

Le troisième chapitre, intitulé «**Conception des données et des applications**», fournit une description détaillée de l'architecture adoptée pour la solution proposée, l'approche de modélisation utilisée ainsi que les indicateurs de performance clés (KPI) calculés.

Le quatrième chapitre, intitulé «**Intégration, Développement et Déploiement**», présente l'environnement de développement et les outils utilisés pour la mise en œuvre de notre solution. Il détaille également les étapes impliquées dans le processus de mise en œuvre et les défis rencontrés.

Finalement, le rapport est clôturé par une conclusion générale qui résume notre travail et nous proposons quelques perspectives.

Présentation générale

Introduction

Nous commençons ce chapitre par une présentation de l'organisme d'accueil. Nous détaillerons ensuite la problématique de notre stage et la solution proposée pour remédier aux limitations des solutions existantes. Le calendrier du stage sera présenté via un diagramme de Gantt, et nous préciserons la méthodologie de gestion de projet utilisée.

Nous aborderons également les concepts clés de l'informatique décisionnelle, tels que l'architecture décisionnelle, le processus ETL (Extraction, Transformation, Chargement) et le reporting, ainsi que les outils utilisés tout au long du projet.

1.1 Présentation de la société d'accueil

— **Présentation** : WEBIFY TECHNOLOGY est une agence tunisienne fondée en 2019 à Sousse, formée d'une équipe d'ingénieurs doués et ardents. Cette société est spécialisée dans la conception et la réalisation de sites / applications Web et Mobiles. Elle est ouverte aux dernières tendances, afin d'offrir un service adéquat, suite à des connaissances diverses et un engagement à long terme. L'innovation est la valeur sûre de WEBIFY TECHNOLOGY, toute l'équipe aime tout ce qu'elle fait, la satisfaction et le respect des engagements est leur point fort.

L'approche d'affaires adoptée par WEBIFY TECHNOLOGY est axée sur l'écoute et des besoins de ses clients et partenaires dans le but de : fournir un conseil personnalisé bien déterminé, aider à atteindre les objectifs tout en respectant le budget ainsi que l'échéance prévu et accompagner, superviser, conseiller et épauler les clients.

— **Services** : Les services proposés par Webify sont organisés, essentiellement selon trois axes à savoir :

— Management : Gestion de projets, Stratégies, Développement d'affaires à l'international.

— Technologies innovantes : Référencement et web analytics, Marketing digital, Web responsive, Métadonnées, Géolocalisation, Automatisation.

— Marketing interactif : Stratégie marketing, Identité visuelle, Conception et design emballage, Conception et design graphique, Production de tout type de contenu multimédia,

Recherche et veille stratégique, Communications interactives, Campagnes de publicité et accompagnement marketing, Marketing par Emailing.

- Coordonnées : Les coordonnées de l'entreprise sont détaillées ci-dessous :
Webify Tecchnology a deux branches, une à Sousse et l'autre à France.
Adresse : Immeuble jaziri Boulevard 14 janvier, HammamSousse
Téléphone : +216 25 23 43 69
LinkedIn : <https://www.linkedin.com/company/webify-technology/>
Facebook : <https://www.facebook.com/Webify.Tech>

Ce travail fait partie du projet de fin d'études visant à obtenir le diplôme d'ingénieur en génie logiciel. Il a été réalisé au sein de la startup Webify.



Logo de Webify

1.2 Présentation du projet

Dans cette section, nous mettons notre travail dans son contexte général. Tout d'abord, nous abordons la problématique qui nous a conduit à sélectionner ce sujet, ainsi que les raisons sous-jacentes. Ensuite, nous exposons la solution proposée afin de résoudre la problématique. À la fin, nous détaillons les objectifs de ce projet.

1.2.1 Problématique

L'application Thunder Express, développée par l'entreprise Webify, est une plateforme de livraison de repas en service depuis deux ans, qui génère une multitude de données. Le principal défi pour l'entreprise est d'extraire des informations significatives et exploitables de ces données. Les décideurs ont besoin d'une vue d'ensemble de l'état général de l'application afin de prendre des décisions stratégiques qui favoriseront sa croissance et son succès. Cette compréhension repose sur l'analyse des données, qui fournit des insights précieux pour orienter leurs actions et améliorer le service de livraison.

Cependant, en raison de l'ampleur et de la diversité des données, les décideurs de l'application rencontrent des difficultés à les analyser et à les traiter pour évaluer les performances et l'efficacité de Thunder Express. Ce manque d'outils d'aide à la décision efficaces et de visibilité claire sur les résultats passés se traduit par une dépendance à des rapports créés dans Excel ou d'autres plateformes similaires, qui s'avèrent inadéquats et ne répondent pas aux besoins spécifiques des décideurs.

Pour surmonter ces défis, plusieurs questions se posent :

- Comment exploiter les données existantes pour améliorer la prise de décision stratégique ?
- Quels indicateurs clés de performance (KPI) devrions-nous mettre en place pour mesurer efficacement la performance de Thunder Express ?
- Comment l'intégration d'un système de Business Intelligence (BI) peut-elle optimiser les processus opérationnels de Thunder Express ?
- Quelles sources de données internes et externes devrions-nous intégrer pour obtenir une vue à 360 degrés de notre activité ?
- Quelles fonctionnalités de visualisation des données seraient les plus bénéfiques pour faciliter l'analyse par les décideurs ?
- Quelles mesures devrions-nous mettre en place pour garantir la sécurité et la confidentialité des données dans notre système BI ?
- Comment pouvons-nous former et accompagner les utilisateurs pour une adoption réussie du nouveau système BI ?
- Comment pouvons-nous former et accompagner les utilisateurs pour une adoption réussie du nouveau système BI ?
- Comment pouvons-nous utiliser l'analyse prédictive pour anticiper les tendances du marché et ajuster notre stratégie en conséquence ?

1.2.2 Prospection des solutions existantes

La phase d'étude de l'existant est essentielle. Elle consiste d'une part à analyser les solutions similaires existantes dans le but d'identifier leurs points faibles pour les améliorer. D'autre part, elle facilite la spécification d'une nouvelle solution optimale et efficace.

Dans ce paragraphe, nous présentons les différentes solutions existantes étudiées dans le cadre de notre projet.

1. **InetSoft** : propose un tableau de bord analytique puissant et adaptable pour la livraison de nourriture, intégrant des KPIs et des analyses pour optimiser les services et augmenter les revenus grâce aux retours clients et aux performances des restaurants.



FIGURE 1.1 – Les interfaces de tableau de bord InetSoft

2. **Otter** : centralise les commandes de plusieurs plateformes de livraison dans un tableau de bord unique, offrant des fonctionnalités de gestion des menus et de rapports analytiques pour optimiser les opérations et l'efficacité des livraisons.

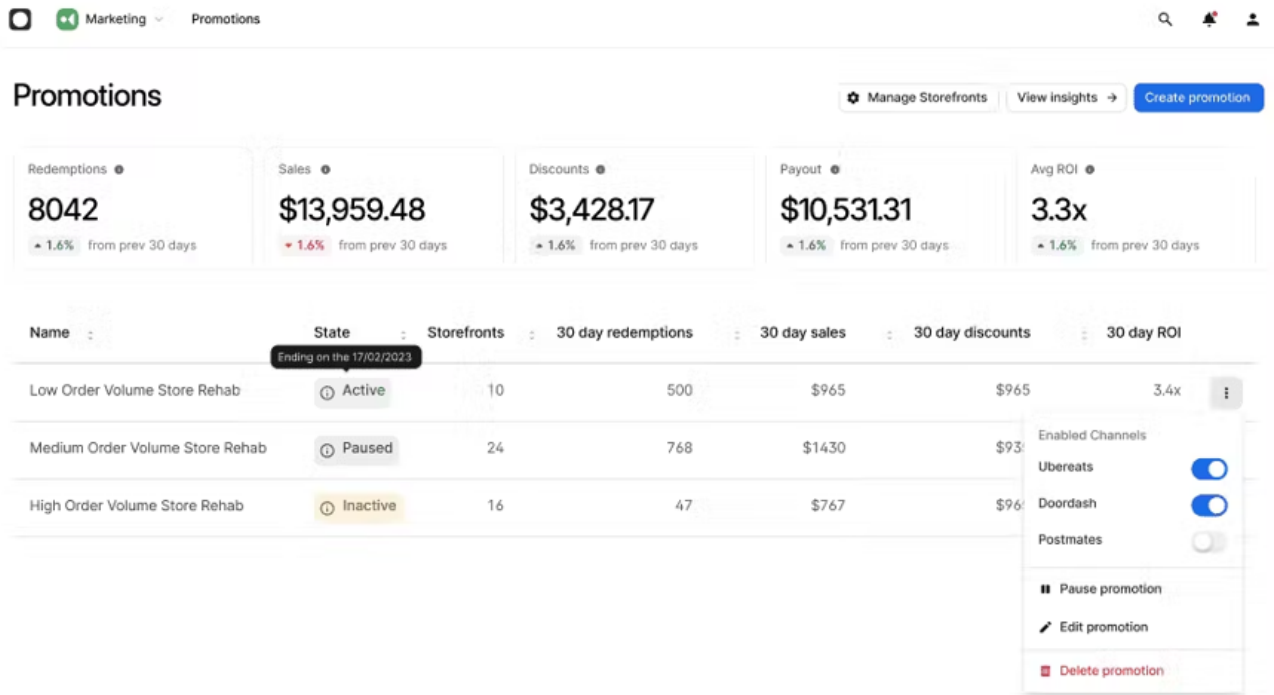


FIGURE 1.2 – Les interface de tableau de bord Otter

3. **Tookan** : Conçu pour les entreprises de livraison à la demande, Tookan offre un tableau de bord centralisé pour créer et gérer les missions de livraison.

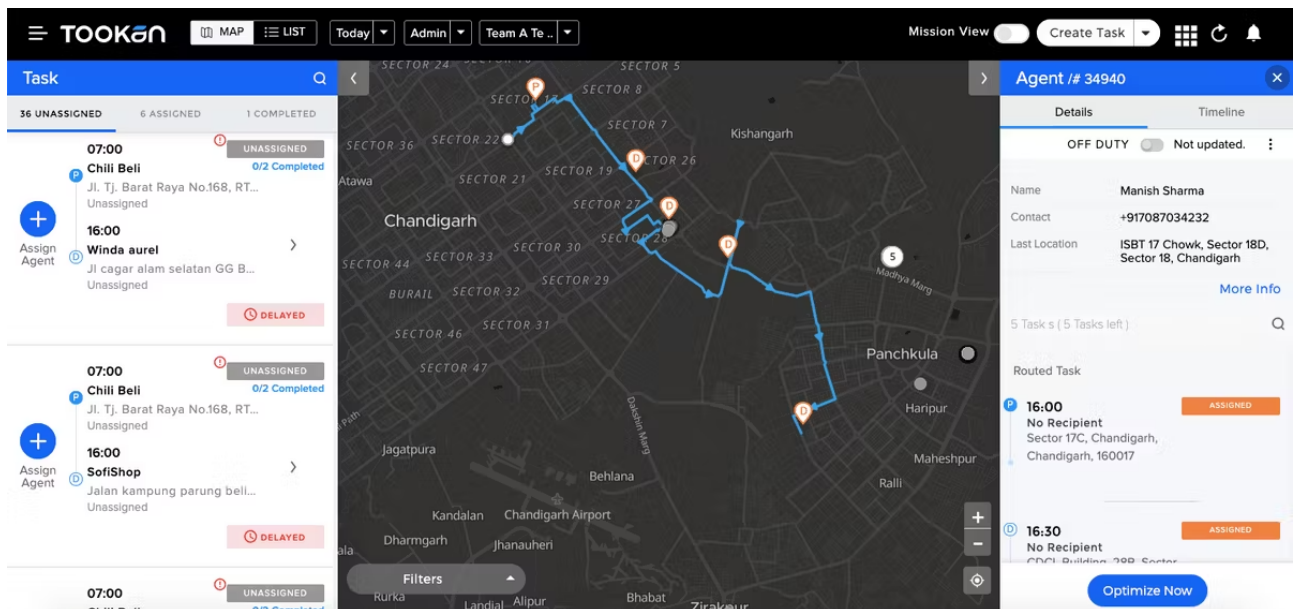


FIGURE 1.3 – Les interfaces de tableau de bord Tookan

4. **Thunder Express** : est une application de livraison de repas développée par l'entreprise Webify, en fonctionnement depuis deux ans. Elle dispose d'un tableau de bord intégré à l'interface de son back-office administrateur.

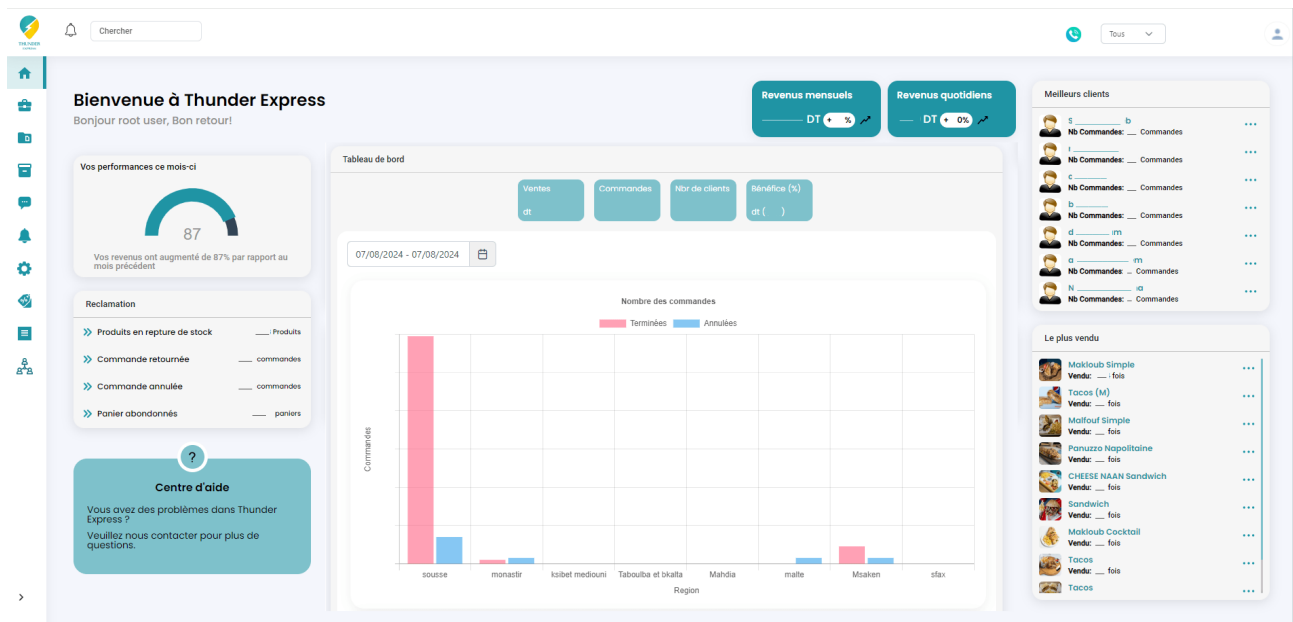


FIGURE 1.4 – Les interfaces de tableau de bord Thunder Express

Nous analysons et critiquons ces solutions à travers le tableau 3.10 :

Applications	Avantages	Inconvénients
InetSoft[1]	<ul style="list-style-type: none"> ● Visualisation puissante et adaptabilité : InetSoft offre des fonctionnalités de visualisation de données robustes et une adaptabilité multi-appareils. ● Indicateurs clés de performance (KPIs) : Le tableau de bord inclut des KPIs qui fournissent une vue consolidée et quantifiable des performances globales. 	<ul style="list-style-type: none"> ● Complexité initiale : La mise en place et la familiarisation avec les fonctionnalités avancées peuvent nécessiter un certain temps et des compétences techniques, ce qui peut être un défi pour les utilisateurs moins expérimentés. ● Coût élevé : Les solutions avancées d’InetSoft peuvent être coûteuses.
Otter[2]	<ul style="list-style-type: none"> ● Centralisation des commandes : Otter permet de centraliser les commandes provenant de plusieurs plateformes de livraison dans un tableau de bord unique. ● Rapports analytiques détaillés : Otter offre des fonctionnalités avancées de suivi des performances et des rapports analytiques. 	<ul style="list-style-type: none"> ● Courbe d’apprentissage : La familiarisation avec toutes les fonctionnalités d’Otter peut prendre du temps, surtout pour les utilisateurs non techniques. ● Coût élevé : Les solutions d’Otter peuvent être coûteuses.
Tookan[3]	<ul style="list-style-type: none"> ● Suivi en temps réel : Tookan offre un suivi fiable en temps réel des livraisons. ● Interface conviviale : L’interface utilisateur de Tookan est intuitive. 	<ul style="list-style-type: none"> ● Service client : Certaines critiques indiquent que le service client de Tookan peut être lent.

ThunderExpress	<ul style="list-style-type: none"> • Global overview : Le tableau de bord donne une vue d'ensemble claire et concise des performances de l'entreprise. Il permet de visualiser rapidement les revenus, les commandes, le nombre de clients et les produits les plus vendus. 	<ul style="list-style-type: none"> • Manque de détails : Le tableau de bord n'offre qu'une vue globale des performances et ne fournit pas de détails sur les données présentées. • Focus sur les ventes uniquement : Le tableau de bord se concentre principalement sur les ventes, et pourrait bénéficier de l'inclusion d'autres indicateurs clés de performance (KPI).
----------------	--	---

TABLE 1.1 – Étude de l'existant

L'objectif de notre solution est d'intégrer tous les points forts des solutions existantes et en même temps compenser leurs points faibles.

1.2.3 Objectifs du projet

Pour surmonter les limites des systèmes existants et répondre efficacement aux problématiques de Thunder Express, une solution de Business Intelligence (BI) a été adoptée pour l'application de livraison. Cette solution prend la forme d'un tableau de bord intégré au back-office administrateur, offrant une plateforme fiable et complète pour l'intégration des données et la génération de rapports, afin de soutenir les décideurs.

L'architecture de la solution BI inclut un outil d'intégration de données robuste qui facilite les processus d'extraction, de transformation et de chargement (ETL). De plus, un outil de reporting permet l'exploration et l'analyse des données. Grâce à cet outil, les utilisateurs peuvent générer des rapports interactifs et détaillés, améliorant ainsi la prise de décision grâce à des insights pertinents et actualisés.

Pour mener à bien notre solution BI et pouvoir répondre à toutes les questions énoncées dans la section problématique, nous devons, dans la mesure du possible :

- Mettre en place un tableau de bord exécutif qui synthétise les données clés et permet de prendre des décisions éclairées.
- Définir et implémenter un ensemble de KPI critiques pour chaque service ou acteur impliqué.
- Réduire le temps de traitement des rapports mensuels grâce à l'automatisation des processus BI.
- Créer des rapports de visualisation interactifs adaptés aux besoins spécifiques de chaque service ou acteur.
- Intégrer les sources de données pertinentes et nécessaires dans le système BI.
- Implémenter un système de contrôle d'accès basé sur les rôles.
- Atteindre un taux d'adoption élevé du nouveau système BI parmi les utilisateurs cibles grâce à un programme de formation complet.

- Développer un modèle prédictif capable de prévoir les ventes avec une grande précision.

Cette approche vise à optimiser les opérations de Thunder Express, en rendant les informations plus accessibles et en améliorant l'efficacité des processus décisionnels.

1.2.4 Processus de développement

Pour assurer une gestion efficace du projet, une bonne organisation est nécessaire. Nous avons choisi d'adopter la méthode agile SCRUM pour la conception et le développement de notre solution.

- **Méthode agile**

La méthode agile est une approche de gestion de projet qui favorise la flexibilité, la collaboration et l'adaptation aux changements tout au long du processus de développement. Elle se caractérise par des cycles itératifs et un fort engagement des parties prenantes.

- **SCRUM**

Scrum est un cadre de gestion de projet agile qui favorise la collaboration, l'adaptabilité et la livraison itérative de produits. Il se caractérise par des cycles de travail appelés "sprints" et une approche centrée sur l'équipe, l'auto-organisation et l'amélioration continue.

- **Rôles**

SCRUM s'articule autour de trois rôles particuliers, chacun ayant une fonction bien précise :

- **Scrum Master** : est le facilitateur agile qui guide l'équipe de développement dans l'application de Scrum et favorise la collaboration et l'amélioration continue.
- **Product Owner** : est le responsable du produit, définissant la vision et les priorités, et collaborant avec l'équipe de développement pour assurer sa création et son amélioration continue.
- **L'équipe de développement** : est une équipe auto-organisée et pluridisciplinaire, responsable de la réalisation des éléments du backlog du produit lors des sprints.

Le tableau 1.2 présente les acteurs du projet conformément à la méthodologie Scrum :

Rôle	Acteur
Scrum Master	Islem Dahmen
Product Owner	Maha Kchaich
Équipe	<ul style="list-style-type: none"> — Ines Belkhiria — Abir Dahmen

TABLE 1.2 – Acteurs du projet selon "Scrum"

- **Pratiques**

Les artefacts Scrum sont des éléments permettant de faire fonctionner le cadre de travail Scrum :

- **Le Product Backlog** : est une liste ordonnée de toutes les fonctionnalités, les exigences et les améliorations souhaitées pour le produit.
- **Le Sprint Backlog** : est une liste des éléments du Product Backlog qui ont été sélectionnés pour être réalisés pendant un Sprint spécifique.
- **L'Increment ou Le sprint** : est une itération de développement, aboutissant à un incrément de fonctionnalité potentiellement livrables.
- **Les User Stories** : sont des descriptions courtes des fonctionnalités du produit, du point de vue de l'utilisateur.
- **Les Reunions Scrum** : sont des réunions régulières et structurées qui font partie du cadre Scrum. Ces réunions permettent à l'équipe de se synchroniser, de planifier et d'adapter son travail.

1.3 Étude théorique

1.3.1 Business Intelligence

La Business Intelligence (BI) combine analytique métier, data mining, visualisation de données, outils et infrastructure de gestion des données et meilleures pratiques pour aider les entreprises à prendre des décisions basées sur les données. En pratique, on parle de BI lorsque vous disposez d'une vue d'ensemble des données de votre entreprise et utilisez ces données pour favoriser le changement, gagner en efficacité et vous adapter aux évolutions du marché.[4]

1.3.2 Architecture BI

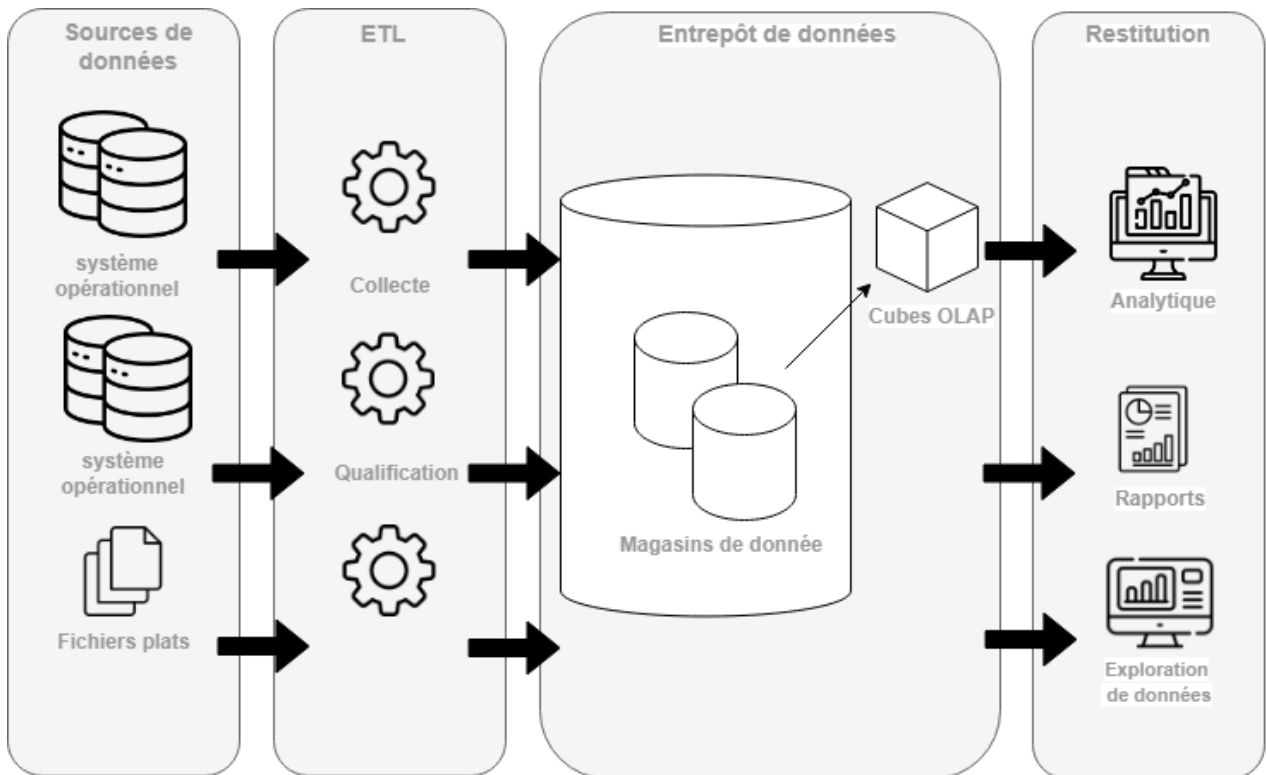


FIGURE 1.5 – Architecture de Business Intelligence

L'architecture de Business Intelligence (BI) fournit la structure essentielle pour atteindre les objectifs de BI. Elle englobe diverses méthodologies, concepts et outils permettant la collecte, le traitement et la visualisation des données.

— Entrepôt de données (Data Warehouse)

Un entrepôt de données (Data Warehouse) est un système de gestion de base de données centralisé conçu pour stocker, intégrer et analyser des données provenant de sources diverses. Il permet de consolider de grandes quantités de données historisées et transformées, rendant ainsi possible une analyse approfondie et une prise de décision éclairée.

— Magasins de données (Data mart)

Un datamart est un entrepôt de données plus petit et ciblé, conçu pour servir les besoins spécifiques d'un département ou d'une fonction particulière au sein d'une organisation. Il extrait et agrège les données de l'entrepôt de données global ou directement des sources opérationnelles pour fournir des analyses spécialisées et rapides.

— Processus ETL (Extraction, Transformation et Chargement)

Le processus ETL (Extraction, Transformation et Chargement) est une série d'étapes utilisées pour intégrer des données provenant de diverses sources, les transformer en un format approprié, et les charger dans un entrepôt de données ou un autre système de stockage. C'est

une composante essentielle des architectures de Business Intelligence (BI) et des systèmes d'entreposage de données, permettant de préparer les données pour l'analyse et le reporting.

1. **Extraction** : Cette étape consiste à collecter des données à partir de diverses sources d'origine, qui peuvent inclure des bases de données relationnelles, des systèmes transactionnels, des fichiers plats, des API, et d'autres sources de données.
2. **Transformation** : Les données extraites sont nettoyées, formatées et transformées pour répondre aux besoins de l'analyse et pour être compatibles avec le modèle de données de l'entrepôt de données.
3. **Chargement** : Cette étape consiste à insérer les données transformées dans le système de stockage cible, typiquement un entrepôt de données, où elles sont disponibles pour l'analyse et le reporting.

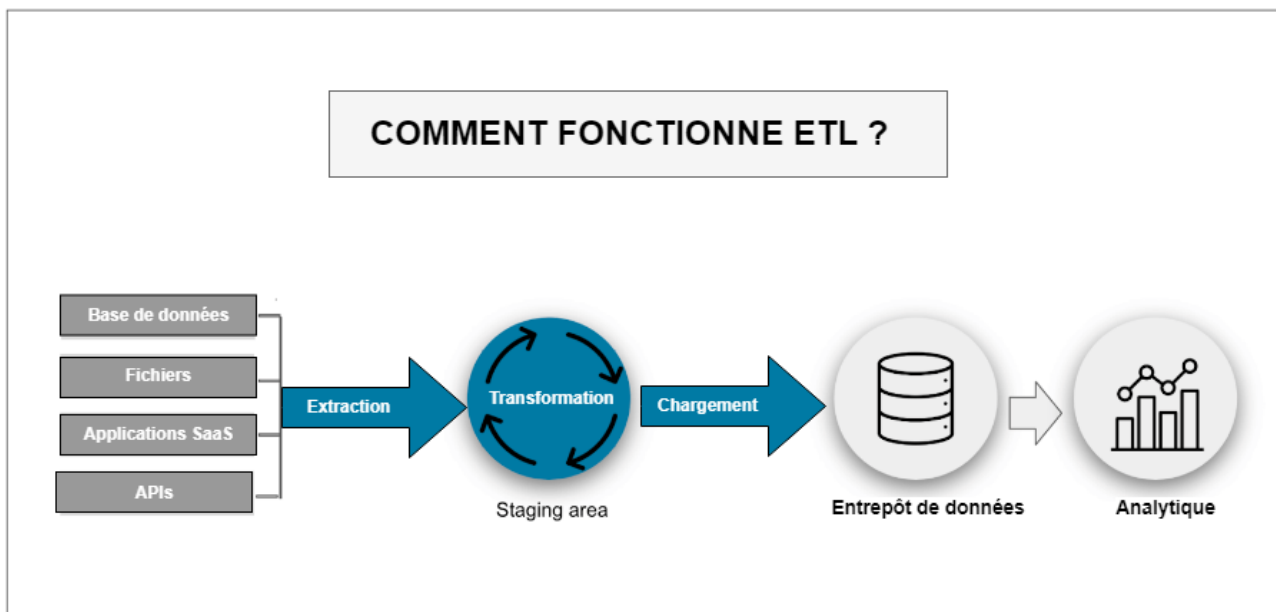


FIGURE 1.6 – Processus ETL

— Reporting

Le reporting est le processus de collecte, de structuration, d'analyse et de présentation de données pour fournir des informations pertinentes aux décideurs au sein d'une organisation. Il permet de transformer des données brutes en informations exploitables, facilitant ainsi la prise de décision et la gestion des performances.

1.3.3 Modélisation BI

Les projets BI nécessitent un entrepôt de données (DWH). La modélisation du DWH est essentielle pour éviter des erreurs futures. Les modèles de données sont conçus selon des besoins et des règles métiers spécifiques, ce qui peut entraîner des représentations différentes même avec les mêmes données. Ces règles, qui définissent le système d'information et les relations entre les groupes de données, se traduisent en schémas spécifiques : étoile, flocon de neige ou constellation.

1. Schéma en étoile

Le schéma en étoile est considéré comme le type de schéma le plus simple pour relier les dimensions et les faits dans un entrepôt de données. Il se caractérise par une connexion directe où une table de faits centrale sert de noyau au schéma, entourée de tables de dimensions. Dans le schéma en étoile, la table de faits contient les clés primaires des tables de dimensions qui y sont liées. La simplicité et les relations directes du schéma en étoile en font une option privilégiée pour la mise en œuvre de solutions d'entreposage de données.

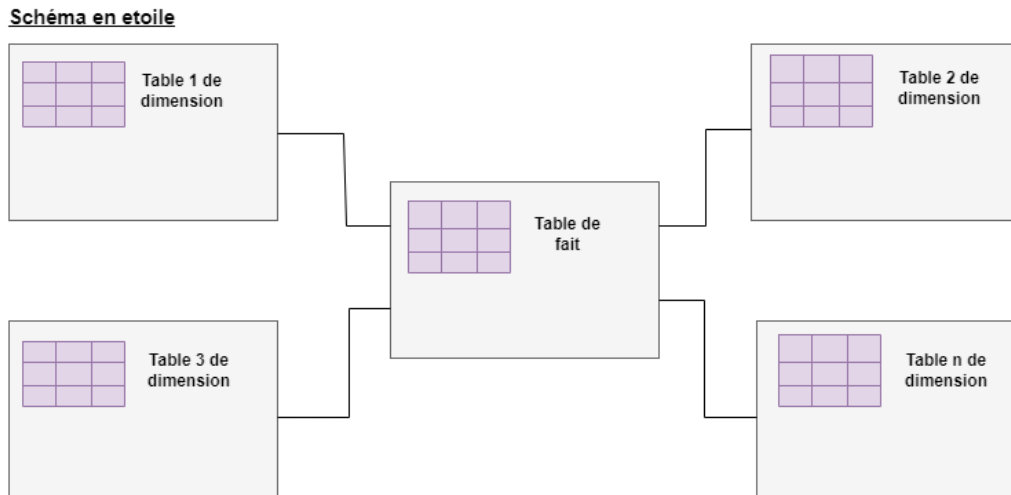


FIGURE 1.7 – Schéma en étoile

2. Schéma en flocon

Le schéma en flocon de neige comprend une table de faits centrale reliée à plusieurs tables de dimensions, dont certaines peuvent également se connecter à d'autres dimensions, créant ainsi une structure hiérarchique. Ce schéma permet d'intégrer un nombre illimité de dimensions, ce qui offre la possibilité de capturer divers aspects des données. Chaque dimension peut inclure plusieurs niveaux, permettant une représentation détaillée des informations. Cette structure hiérarchique offre un modèle de données flexible et évolutif. La Figure 2.4 illustre un schéma en flocon de neige avec deux dimensions, chacune comportant trois niveaux.

Schéma en flocon de neige

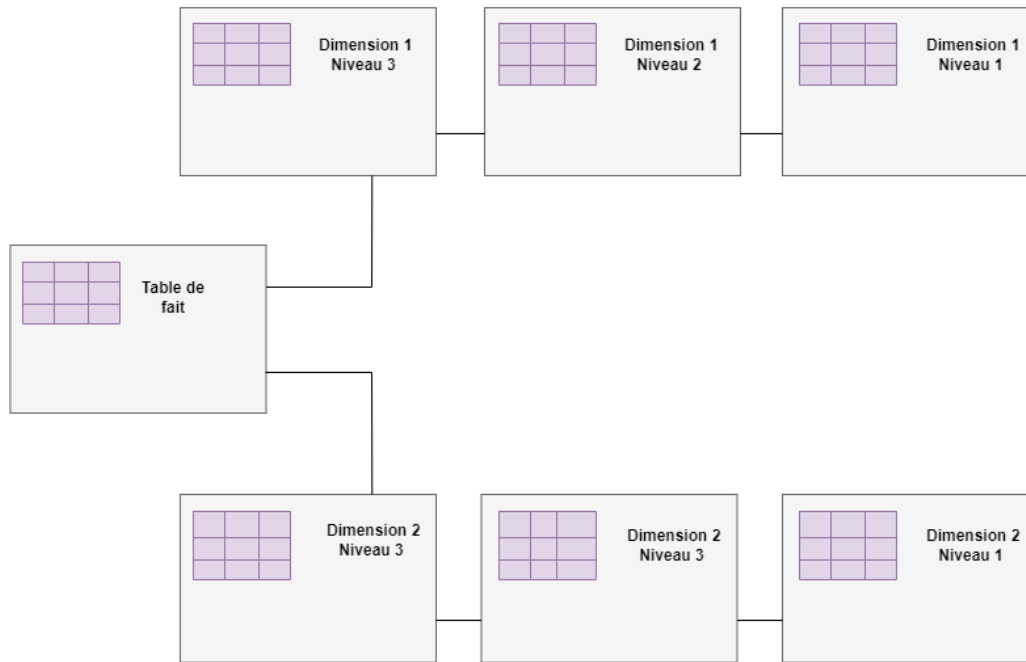


FIGURE 1.8 – Schéma en flocon de neige

3. Schéma en constellation

La constellation est souvent considérée comme l'un des schémas les plus complexes. Elle permet l'utilisation de plusieurs tables de faits, chacune représentant un schéma en étoile distinct, tout en partageant des tables de dimensions communes. Cela permet de représenter des relations complexes et d'offrir diverses perspectives d'analyse au sein de l'architecture de l'entrepôt de données. La figure 2.5 illustre la structure de base d'un schéma en constellation.

Schéma en constellation

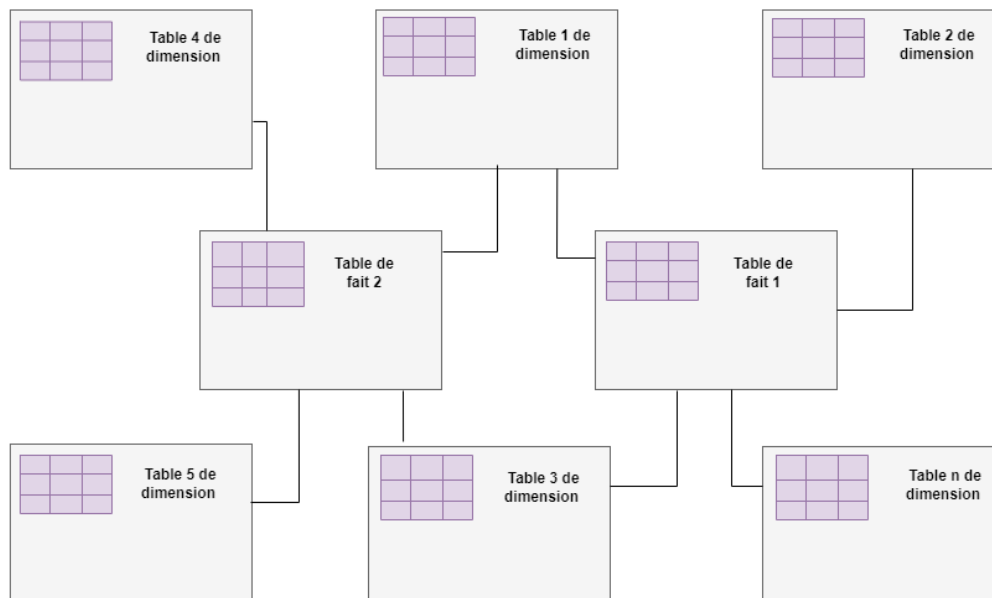


FIGURE 1.9 – Schéma en constellation

1.3.4 Méthodologies de entrepôts de données

Pour réussir les projets de BI, il est essentiel de sélectionner une méthodologie de conception appropriée. Ainsi, choisir la bonne méthodologie nécessite une compréhension approfondie de chaque option disponible et de sa capacité à répondre aux contraintes du projet. En matière de conception, deux approches architecturales couramment utilisées sont celles de Kimball et d’Inmon. Nous présenterons ci-dessous les différentes méthodologies.

1. Méthodologie d’ Inmon

Bill Inmon, le père de l’entrepôt de données, a proposé le concept de développement d’un entrepôt de données qui identifie les principaux domaines et entités avec lesquels l’entreprise travaille, tels que les clients, les produits, les fournisseurs, etc. La définition de Bill Inmon d’un entrepôt de données est qu’il s’agit d’une "collection de données orientée sujet, non volatile, intégrée et variant dans le temps à l’appui des décisions de la direction".[5]

L’approche top-down de Bill Inmon pour la conception d’un entrepôt de données commence par la création d’un entrepôt de données centralisé et normalisé, souvent appelé Data Warehouse. Cette méthode préconise une collecte des données exhaustive et leur intégration avant toute utilisation dans des data marts ou des applications spécifiques.

La figure 2.6 présente une représentation visuelle de la méthodologie d’Inmon, mettant en valeur son cadre conceptuel :

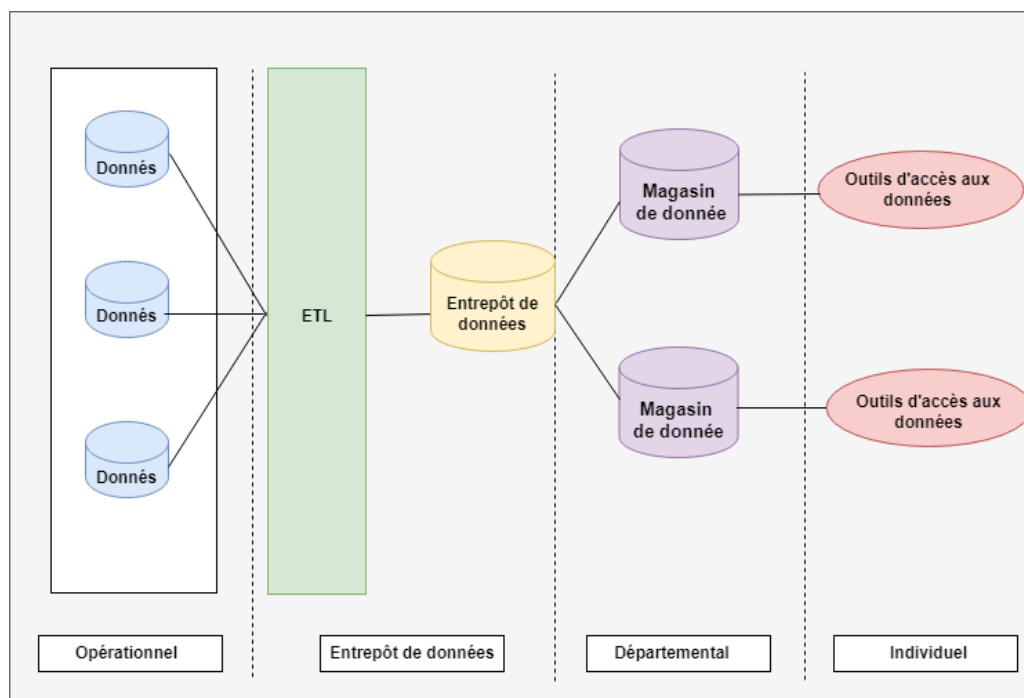


FIGURE 1.10 – Approche Inmon

2. Méthodologie de Kimball

Initié par Ralph Kimball, le modèle de données Kimball suit une approche ascendante pour conception d’architecture d’entrepôt de données dans lequel les magasins de données sont d’abord formés en fonction des besoins de l’entreprise.

Les principales sources de données sont ensuite évaluées et un Outil d'extraction, de transformation et de chargement (ETL) est utilisé pour récupérer des données à partir de plusieurs sources et les charger dans une zone de transit du serveur de base de données relationnelle. Une fois les données téléchargées dans la zone de préparation de l'entrepôt de données, la phase suivante consiste à charger les données dans un modèle d'entrepôt de données dimensionnel qui est par nature dénormalisé. Ce modèle divise les données en table de faits, qui sont des données transactionnelles numériques ou une table de dimensions, qui sont les informations de référence qui prennent en charge les faits.[6]
 La figure 2.6 présente une représentation visuelle de la méthodologie de Kimball, mettant en valeur son cadre conceptuel :

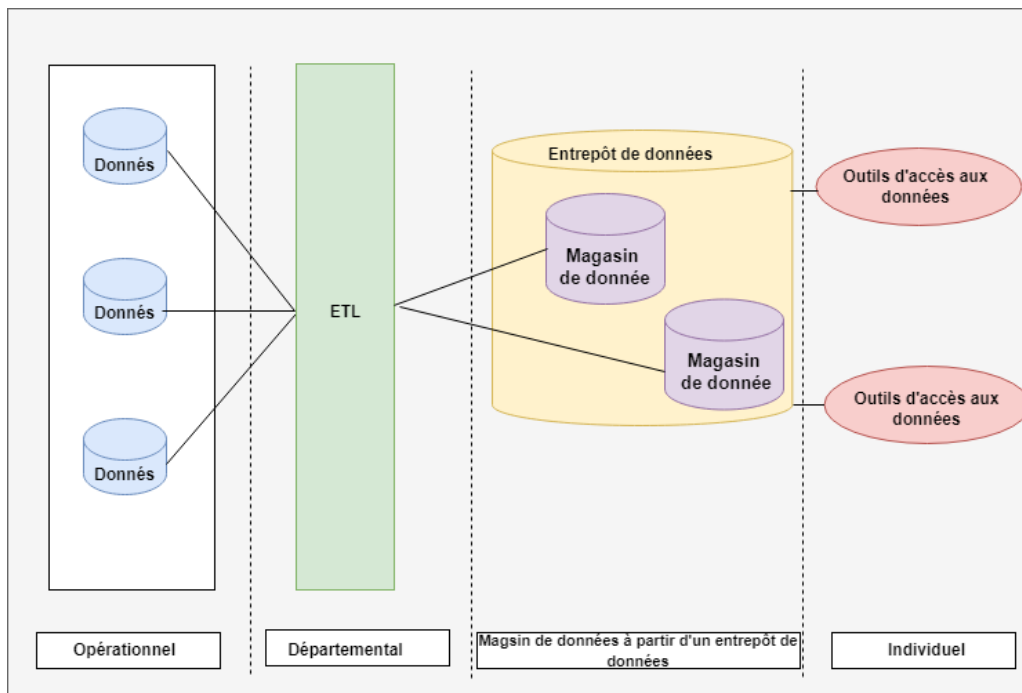


FIGURE 1.11 – Approche Kimball

3. Inmon vs Kimball

Critère	Approche Inmon	Approche Kimball
Auteur	Bill Inmon	Ralph Kimball
Concept clé	Entrepôt de données d'entreprise	Datamarts dimensionnels
Philosophie	Top-Down	Bottom-Up
Conception initiale	Entrepôt de données centralisé, puis datamarts	Niveau des datamarts
Intégration des données	Niveau de l'entrepôt de données	Datamarts spécifiques aux départements
Évolutivité	Plus facile à étendre pour de nouveaux domaines	Rapide à mettre en œuvre, mais peut être plus complexe à gérer à grande échelle

Redondance des données	Minimisée grâce à la normalisation	Potentiellement élevée en raison de la dénormalisation
Performance	Peut être moins performant pour les requêtes complexes	Généralement performant pour les requêtes analytiques
Complexité de mise en œuvre	Initialement plus complexe et coûteux	Initialement plus simple et moins coûteux
Flexibilité	Très flexible, mais nécessite une planification rigoureuse	Flexible et adaptable aux changements rapides
Utilisateurs cibles	Grandes entreprises avec des besoins analytiques complexes	Organisations cherchant des résultats rapides et faciles à mettre en œuvre

TABLE 1.3 – Comparaison entre les approches Inmon et Kimball

En résumé, l’approche Inmon est plus orientée vers une intégration globale et normalisée des données, offrant une flexibilité et une consistance élevées au prix d’une complexité initiale. L’approche Kimball, en revanche, se concentre sur des résultats rapides et des datamarts spécifiques, privilégiant la simplicité et la performance des requêtes analytiques au prix d’une potentielle redondance des données et d’une gestion plus complexe à grande échelle.

4. Méthodologie adoptée

Le choix de la méthodologie pour développer un entrepôt de données dépend fortement de l’objectif final du projet. Les approches de Kimball et d’Inmon ont leurs forces et faiblesses, et aucune ne satisfait pleinement toutes les exigences. L’approche de création d’un entrepôt de données doit être guidée par les objectifs commerciaux de l’entreprise, la nature de l’activité, le temps et l’argent disponibles, ainsi que le niveau de dépendance entre les différentes fonctions.

L’approche préconisée par Ralph Kimball est privilégiée lorsqu’il existe une forte focalisation sur l’optimisation locale et l’obtention de succès rapides, car la mise en œuvre de l’entrepôt de données est relativement rapide par rapport à d’autres approches. Dans notre cas, le choix de la méthode Kimball est avantageux car il permet de segmenter l’entrepôt de données en plusieurs datamarts logiques et cohérents, plutôt que d’opter pour un modèle vaste, centralisé, général et souvent complexe.

L’approche de Kimball est idéale pour des projets nécessitant des solutions agiles et conviviales, avec une livraison rapide et une capacité d’adaptation aux besoins commerciaux évolutifs. Elle permet de représenter les données de manière intuitive et simple, offrant une vue claire et concise des relations entre les données. C’est précisément pour ces raisons que nous avons choisi cette méthode pour notre projet, car elle répond parfaitement à nos exigences en matière de flexibilité, de clarté et de rapidité de mise en œuvre.

Pour mettre en pratique la méthodologie Kimball, voici les étapes clés à suivre :

1. Spécification des besoins et Planification
 - Collecte des exigences
 - Spécification des exigences

2. Conception des données et des applications

- Modélisation dimensionnelle
- Développement ETL
- Environnement technique

3. Intégration, Développement et Déploiement

- Intégration des Données
- Développement des applications BI
- Sécurité et gestion d'accès
- Évaluation des performances
- Déploiement

Conclusion

Dans ce chapitre, nous avons commencé par présenter le contexte général du projet, en introduisant d'abord l'organisme d'accueil, puis en abordant la problématique. Ensuite, nous avons réalisé une analyse des applications existantes, suivi d'une description de la solution proposée. Nous avons également exposé la méthodologie de gestion de projet et établi la planification.

Nous avons ensuite présenté les notions et techniques utilisées dans le projet, fournissant ainsi une explication complète des concepts fondamentaux de la Business Intelligence, qui sont au cœur des objectifs de notre projet.

Dans le chapitre suivant, nous nous concentrerons sur l'analyse détaillée de la solution, en identifiant les besoins spécifiques.

Spécification des besoins et Planification

Introduction

Après avoir présenté en détail les concepts de business intelligence dans le chapitre précédent, nous allons maintenant nous concentrer sur une analyse approfondie des besoins pour les comprendre davantage. Ensuite, ces exigences seront décrites et communiquées à l'aide d'un diagramme de cas d'utilisation.

2.1 Collecte des exigences

2.1.1 Exigences fonctionnelles

Le processus des exigences fonctionnelles décrit le comportement attendu et les fonctionnalités du système, permettant aux utilisateurs d'atteindre avec succès leurs objectifs. Ces exigences fonctionnent comme des lignes directrices pour le développement du système, garantissant que les fonctionnalités requises sont implémentées et que le système répond aux besoins des utilisateurs. Voici les exigences obligatoires pour notre solution :

- **Gestion des Sources de Données** : La gestion des sources de données inclut la mise à jour et la maintenance des sources de données connectées au système BI. Cette tâche est réalisée par l'Analyste de Données, qui s'assure que les informations disponibles sont toujours à jour et précises. De plus, le système permet aux utilisateurs, notamment l'Administrateur, d'exporter des données dans différents formats tels que CSV ou Excel, facilitant ainsi l'analyse externe ou le partage d'informations avec d'autres parties prenantes.
- **Gestion des Rapports et Tableaux de Bord** : La gestion des rapports et des tableaux de bord est un aspect essentiel pour l'analyse des données. L'Analyste de Données a la capacité de gérer des rapports personnalisés, en fonction de critères spécifiques, pour répondre aux besoins variés des utilisateurs. Par ailleurs, il est aussi responsable de la génération de tableaux de bord interactifs, qui permettent de visualiser les données en temps réel. Les utilisateurs, incluant l'Administrateur, le Dispatcheur et l'Analyste de Données, peuvent consulter ces tableaux de bord pour obtenir des informations sur les performances et les analyses. Ils ont également la possibilité d'appliquer des filtres pour personnaliser les vues selon des critères spécifiques. En outre, les tableaux de bord financiers sont accessibles aux Administrateurs, Directeurs Financiers et Analystes de Données,

leur offrant une vue détaillée sur les performances financières de l'organisation.

- **Gestion des Rôles** : La gestion des rôles se concentre sur la gestion des utilisateurs du système BI. L'Administrateur est responsable de la gestion des accès et des permissions des utilisateurs, ce qui implique de définir qui peut voir ou modifier quelles données. Cette fonctionnalité est essentielle pour assurer la sécurité des informations et garantir que seuls les utilisateurs autorisés accèdent à des données sensibles ou critiques.

2.1.2 Définition des indicateurs clés de performance (KPI)

1. KPIs liés aux Clients :

- **Satisfaction client** : Utilisation des mesures telles que les évaluations post-commande pour évaluer la satisfaction globale des clients, offrant ainsi des insights précieux sur la perception de notre service.

$$TauxSatisfaction = \frac{\sum \text{commands}[AvisPositifs]}{\sum \text{commands}[TotalAvis]}$$

- **Taux de fidélité clients** : Calcul du pourcentage de clients existants qui reviennent et passent des commandes répétées, un indicateur essentiel pour mesurer la satisfaction et la rétention des clients.

$$TauxFidelite = \frac{\sum \text{commands}[CommandesReussies]}{\sum \text{commands}[TotalCommandes]}$$

- **Taux de conversion** : Calcul du pourcentage de visiteurs de l'application qui finissent par passer commande, fournissant des informations sur l'efficacité de notre processus de conversion et d'engagement.
- **Croissance des nouveaux clients** : Suivi de la croissance du nombre de nouveaux clients, un indicateur vital pour évaluer l'attraction et l'expansion de notre base clientèle.

2. KPIs liés aux Commandes :

- **Chiffre d'affaires des commandes (mesure de rentabilité), valeurs des commandes** : Analyse du chiffre d'affaires total généré par les commandes ainsi que de la valeur moyenne des commandes, offrant un aperçu de la rentabilité.

$$\text{Pourcentage gain} = \frac{\sum \text{commands}[products_gain]}{\sum \text{commands}[total_price]}$$

- **Durée moyenne de traitement de commande** : Évaluation du temps moyen nécessaire pour traiter une commande de bout en bout, permettant d'identifier les goulets d'étranglement et d'optimiser l'efficacité opérationnelle.

$$\text{Durée Traitement (heures)} = \frac{1}{n} \sum_{i=1}^n \text{DATEDIFF}(\text{creationDate}_i, \text{lastUpdateDate}_i, \text{HOUR})$$

- **Les KPIs liés à la répartition géographique** : Identification des zones géographiques avec les commandes les plus élevées et les opportunités de croissance.

3. KPIs liés aux Fournisseurs et produits :

- **Performance des partenaires** : Suivi des indicateurs de performance individuels pour chaque restaurant partenaire, tels que le temps de préparation des commandes, le taux de satisfaction client, etc., afin de garantir la qualité et la fiabilité des services.

$$\text{Pourcentage Commandes} = \frac{\text{Nombre Commandes Fournisseur}}{\text{Total commandes}} \times 100$$

- **KPIs individuels pour chaque restaurant** : Évaluation de la performance de chaque restaurant partenaire en fonction de critères spécifiques tels que la rapidité de service, la précision des commandes, etc., permettant une gestion proactive.

4. KPIs liés aux livreurs :

- **Taux de livraison à temps** : Mesure du pourcentage de commandes livrées dans les délais impartis par rapport au nombre total de livraisons. Cela permet d'évaluer l'efficacité et la ponctualité des livreurs.

- **Taux de satisfaction des clients lié à la livraison** : Évaluation de la satisfaction des clients concernant la qualité de la livraison, la politesse du livreur, etc., à travers des sondages ou des évaluations post-livraison.

- **Nombre de livraisons par heure ou par jour** : Suivi du volume de livraisons effectuées par chaque livreur sur une période donnée, permettant de mesurer leur productivité et leur charge de travail.

$$\text{Livraisons par Jour} = \sum_{j=1}^n \delta(\text{commands}[\text{DayOfWeek}]_j = [\text{Date Sélectionnée}])$$

$$\text{Livraisons par Heure} = \sum_{h=1}^m \delta(\text{commands}[\text{HourOfDay}]_h = [\text{Heure Sélectionnée}])$$

- **Taux de réussite des livraisons** : Mesure du pourcentage de livraisons réussies par rapport au nombre total de livraisons attribuées à chaque livreur, prenant en compte les livraisons manquées ou les commandes annulées.

$$\text{Commandes Réussites} = \sum_{i=1}^n \delta(\text{commands}[\text{cycle}]_i = \text{"SUCCESS"})$$

$$\text{Taux de Réussite} = \frac{\text{Commandes Réussites}}{\text{Total Commandes}} \times 100$$

- **Nombre d'avis positifs reçus par livreur** : Suivi du nombre d'évaluations positives ou de commentaires favorables attribués à chaque livreur, reflétant leur professionnalisme et leur qualité de service.

- **Taux de retour des commandes** : Évaluation du pourcentage de commandes retournées en raison de problèmes liés à la livraison, tels que des erreurs de commande ou des retards, permettant d'identifier les domaines d'amélioration pour les livreurs.

$$\text{Commandes Retournées} = \sum_{i=1}^n \delta(\text{commands}[\text{cycle}]_i = \text{"REJECTED"})$$

$$\text{Taux de Retour} = \frac{\text{Commandes Retournées}}{\text{Total Commandes}} \times 100$$

2.1.3 Exigences non fonctionnelles

Les exigences non fonctionnelles définissent les caractéristiques essentielles du système pour améliorer sa qualité globale. Notre prochain système doit remplir les critères suivants :

- **Efficacité** : capacité de la solution à fournir les résultats escomptés tout en optimisant l'utilisation des ressources telles que le temps, la mémoire et la puissance de calcul.
- **Évolutivité** : à mesure que les ressources système augmentent, notre solution doit pouvoir s'adapter à la charge de travail accrue, garantissant ainsi sa capacité à gérer la charge supplémentaire. Cela est essentiel pour maintenir les performances de l'application lors d'une utilisation intensive.
- **Fiabilité** : capacité du système à produire de manière cohérente des résultats précis et fiables en fonction des entrées reçues. Étant donné que des défauts ou des dysfonctionnements pourraient entraîner des pertes financières et des pénalités pendant le processus de génération de rapport, il est primordial que le système affiche un haut niveau de fiabilité.
- **Convivialité** : la solution doit posséder des interfaces claires et intuitives pour les utilisateurs finaux, garantissant ainsi une expérience agréable.
- **Sécurité** : Le système doit être capable de protéger les données sensibles et les informations confidentielles contre les accès non autorisés, les fuites de données, et autres menaces de sécurité. Il est essentiel que le système respecte les normes de sécurité en vigueur pour garantir l'intégrité et la confidentialité des informations.
- **Disponibilité** : En cas de problème avec nos serveurs internes, le système est capable de basculer automatiquement vers des serveurs en ligne hébergés sur le cloud. Cette solution de secours garantit que le système reste opérationnel, même en cas de défaillance matérielle ou de maintenance des serveurs principaux. La haute disponibilité est essentielle pour minimiser les interruptions de service et assurer une continuité des activités sans faille.

2.1.4 Faisabilité technique du projet BI

La faisabilité technique évalue si un projet peut être réalisé avec les technologies actuelles, en considérant les ressources, la compatibilité et les compétences nécessaires. Elle détermine la viabilité technologique du projet.

1. **Qualité des Données** : Les données provenant de l'application Thunder Express sont complètes et cohérentes. L'application, présente sur le marché depuis deux ans, a généré une quantité importante de données, avec 25 000 commandes et un nombre significatif de clients. Cela offre une base solide pour le travail en BI. Toutefois, les données nécessitent des processus de nettoyage et de transformation pour répondre aux exigences de la BI.
2. **Infrastructure Technologique** : Les serveurs existants sont suffisants pour les besoins actuels, mais des investissements dans le stockage et les outils BI sont nécessaires pour assurer la scalabilité.
3. **Compétences Techniques** : L'équipe interne de Webify possède des compétences de base en gestion de données et en développement BI. Cependant, une formation complémentaire sur les outils choisis, notamment Power BI, est recommandée pour renforcer les compétences de l'équipe et maximiser l'efficacité des projets BI.
4. **Risques Techniques** : Les principaux risques identifiés incluent la qualité des données et la performance des processus ETL. Des tests de performance réguliers sont recommandés. Les données disponibles dans l'application ne permettent pas de trouver tous les KPI identifiés, et certaines données manquent, comme les heures de travail des livreurs. Pour remédier à cela, une suggestion a été faite d'inclure un système de notification pour rappeler à l'administrateur de remplir les heures de travail des livreurs à la fin de la journée.
De plus, d'autres données manquent, telles que le genre et la date de naissance. Pour résoudre ce problème, il a été proposé d'ajouter une interface incitant les clients à compléter leurs informations personnelles en gagnant des points bonus. Cela permettra d'améliorer la qualité des données et, par conséquent, la pertinence des tableaux de bord.
5. **Prototypage** : Le prototype développé a validé la faisabilité technique, et les retours des utilisateurs sont positifs. Nous avons présenté plusieurs prototypes à la partie administrative, qui nous a donné son accord. L'un de ces prototypes est illustré dans la figure 2.1.

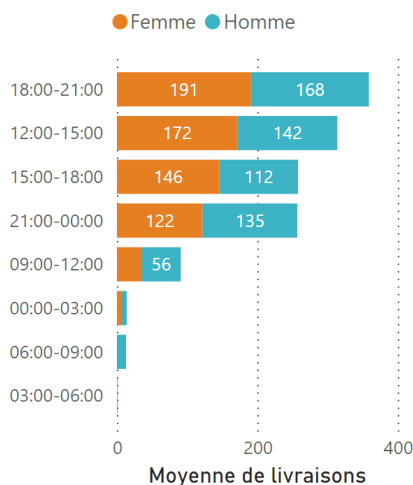


FIGURE 2.1 – Prototype : Plages horaires de livraisons

6. Feuille de Route :

Phase 1 : Formation et préparation (1 mois)

Phase 2 : Développement de l'entrepôt de données et des processus ETL (2 mois)

Phase 3 : Développement des tableaux de bord et des rapports (1 mois)

Phase 4 : Tests et optimisation (1 semaine)

Phase 5 : Déploiement et support (1 semaine)

En suivant ces étapes et en prenant en compte ces aspects, on peut s'assurer que notre projet BI est techniquement faisable et bien préparé pour la mise en œuvre.

2.2 Spécification des exigences

2.2.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.[7]

Nous définissons deux types d'acteurs dans notre projet : Administrateur et dispatcheur.

- **Administrateur** : Cet acteur joue le rôle de responsable système. Il peut réaliser la majorité des opérations de gestion, telles que la gestion des clients, la gestion des partenaires, l'analyse des données et le suivi des commandes. Il s'agit de l'employé responsable de l'administration et de la surveillance du système, assurant son bon fonctionnement et la gestion de ses divers composants.
- **Dispatcheur** : Cet acteur est chargé de la répartition et de la coordination des ressources et des tâches. Le dispatcher est responsable de la distribution et de la gestion des ressources pour assurer une efficacité optimale et un service fluide. Il joue également le rôle de décideur. Il suit la commande de sa création jusqu'à sa finalisation.
- **Analyste de Données** : Cet acteur est chargé de créer des rapports, de générer des tableaux de bord, d'explorer des données et de gérer les utilisateurs.
- **Directeur Financier** : Cet acteur joue un rôle clé dans la santé financière de l'entreprise, en assurant une gestion efficace des ressources financières, en élaborant des stratégies financières solides et en garantissant la conformité aux réglementations financières.
- **Job Scheduler** : Il jouera un rôle secondaire. Le système utilise un processus Cron pour automatiser des tâches comme la mise à jour des données, l'exécution des processus ETL ou la génération de rapports.

2.2.2 Identification des cas d'utilisation

Un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. La création de ce type de diagramme UML requiert un ensemble de symboles et de connecteurs spécifiques.[8]

La figure 3.22 suivante présente le diagramme du cas d'utilisation global illustre les principales fonctionnalités de notre système.

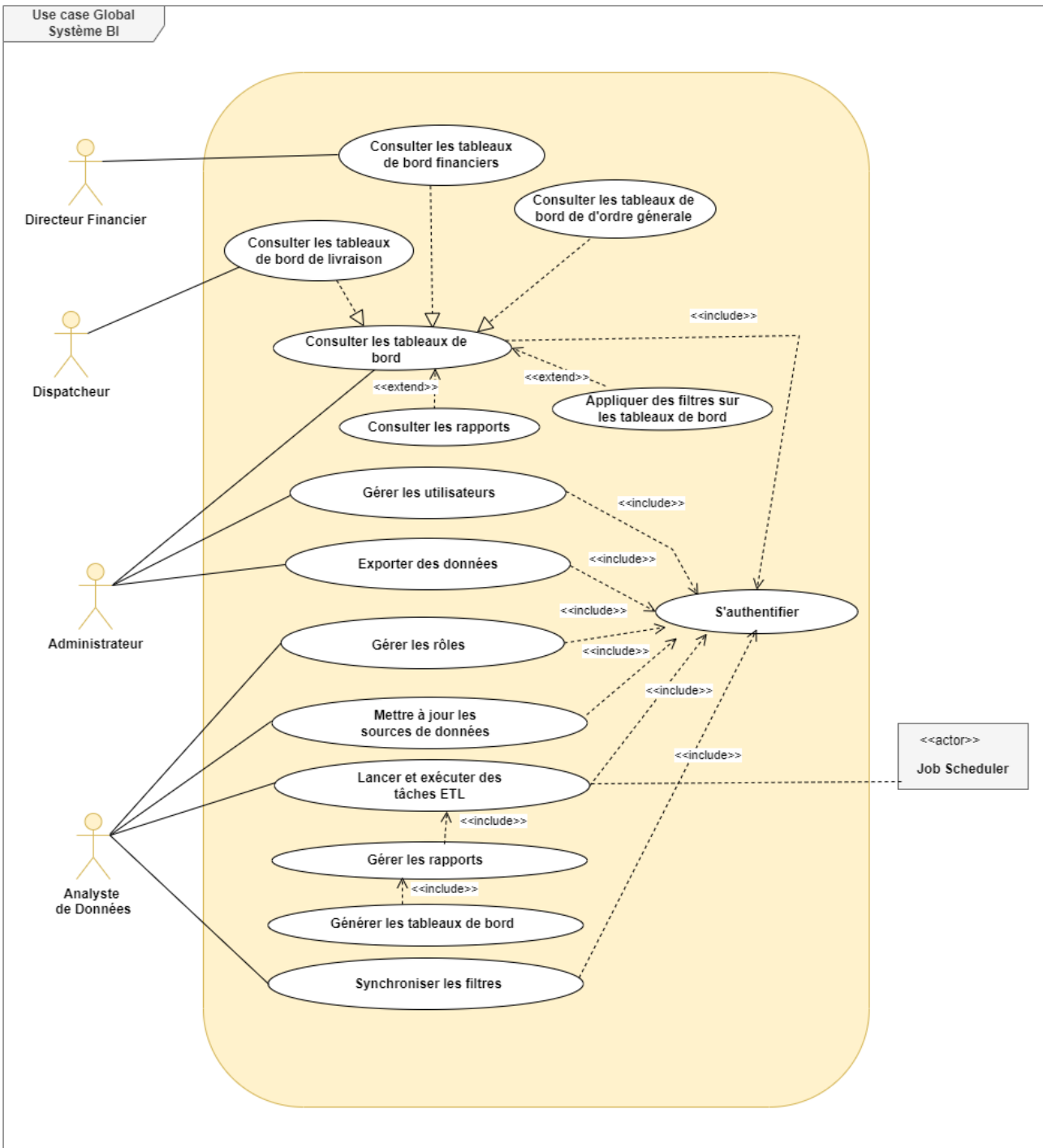


FIGURE 2.2 – Diagramme du cas d'utilisation général

La description de scénarios du cas d'utilisation : « Exporter des Données » est donnée par le tableau 2.1 :

Titre	Exporter des Données
Acteurs	Administrateur

Objectif	Ce cas d'utilisation permet aux administrateur d'exporter les données depuis le système de Business Intelligence (BI) vers divers formats de fichiers (PDF, Excel, CSV, etc.). Cela inclut la sélection des données à exporter, l'application de filtres et de formats spécifiques, et le téléchargement du fichier généré.
Précondition	<ol style="list-style-type: none"> 1. L'administrateur doit être authentifié et autorisé à exporter des données. 2. Les données doivent être disponibles et accessibles pour l'exportation.
Post-condition	<ol style="list-style-type: none"> 1. Les utilisateurs obtiennent un fichier contenant les données exportées au format choisi. 2. Les utilisateurs peuvent télécharger et utiliser le fichier exporté.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne l'option "Exporter des Données" dans le menu principal ou à partir d'un rapport/tableau de bord. 2. Le système affiche une interface de sélection des données et des filtres. 3. Le client aime ce box en cliquant sur le cœur.[A1] 4. Le système ajoute ce box à la liste des favoris des boxes. <p>[S2] : Ajouter partenaire aux favoris :</p> <ol style="list-style-type: none"> 1. Le client fait son choix parmi les boxes affichées. 2. Le système affiche les détails de box sélectionnée. 3. Le client consulte les informations du partenaire de ce box. 4. Le système affiche les détails de ce partenaire. 5. Le client aime ce partenaire en cliquant sur le cœur.[A1] 6. Le système ajoute ce partenaire à la liste des favoris des partenaires.
Scénario alternatif	<p>— [S1.A1] [S2.A1] : Ce box ou ce partenaire se trouve déjà dans la liste des favoris.</p>

Exigences supplémentaires	— Le cœur sur lequel le client clique pour exprimer son appréciation est rouge lorsque le box ou le partenaire est en favoris, et il est gris dans le cas contraire.
---------------------------	--

TABLE 2.1 – Description textuelle du cas « Gérer les favoris »

2.3 Planification du projet

Pour assurer le succès de notre projet, nous avons mis en place une planification détaillée, prenant en compte chaque étape clé et les ressources nécessaires pour atteindre nos objectifs.

Le tableau 3.11 offre une vue d'ensemble des différentes phases et tâches du notre projet, ainsi que les livrables.

N° sprint	Description	Backlog du sprint	Livrables
0	Préliminaire	— Formation et préparation	— Documents de formation
1	Développement de l'entrepôt de données et des processus ETL	— Définition des sources de données	— Modélisation de l'entrepôt de données
2	Développement des tableaux de bord et des rapports	— Conception des tableaux de bord et des rapports — Implémentation des tableaux de bord et des rapports	— Prototypes des tableaux de bord et des rapports — Tableaux de bord et rapports fonctionnels
3	Tests et optimisation	— Tests des fonctionnalités — Optimisation des performances	— Rapport de tests — Améliorations du code et des processus
4	Déploiement et support	— Installation et configuration de l'entrepôt de données — Formation des utilisateurs	— Environnement de production configuré — Documentation utilisateur

TABLE 2.2 – Backlog du notre projet

Le diagramme de gantt est un outil graphique qui représente la gestion du projet dans le temps ce qui facilite sa réalisation. En effet, la figure 2.3 représente l'avancement des activités et des tâches exécutées tout le long de notre projet.

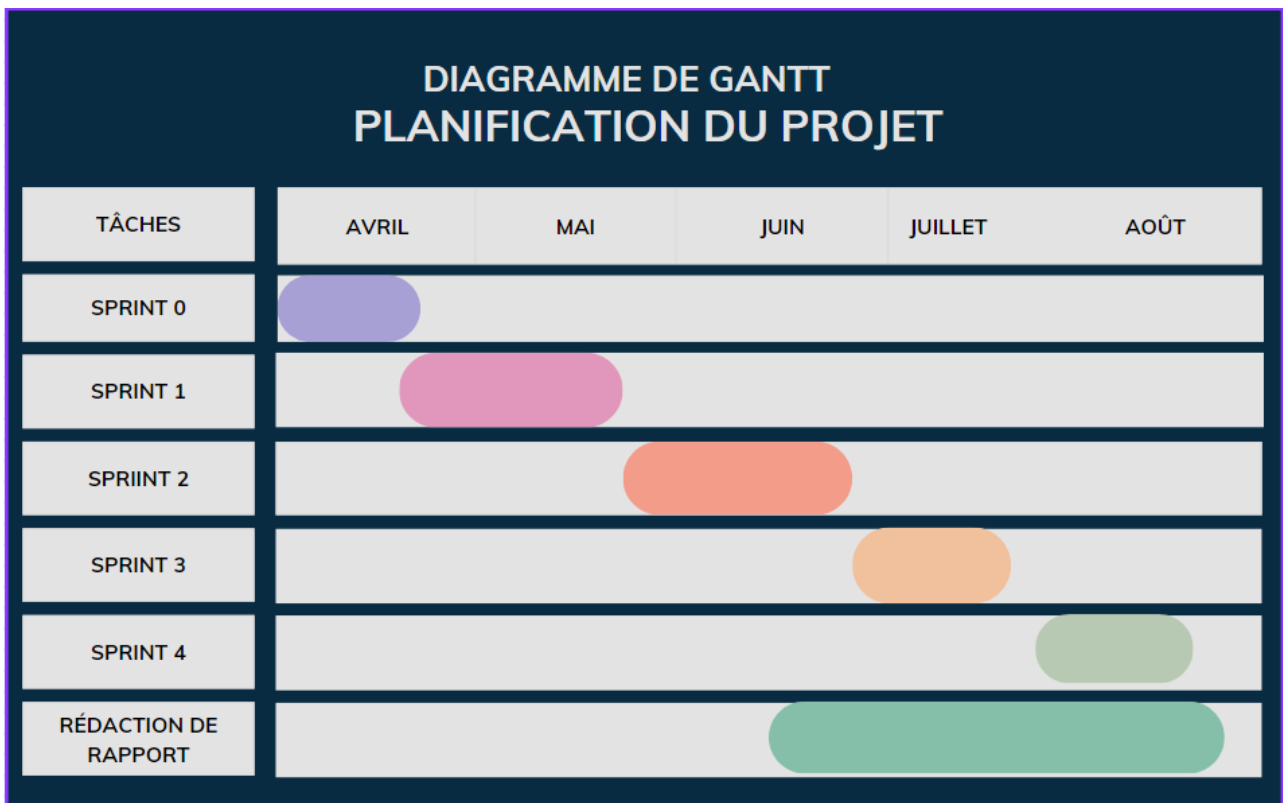


FIGURE 2.3 – Diagramme de GANTT

Conclusion

Conception des données et des applications

Introduction

3.1 Modélisation dimensionnelle

L'analyse des sources de données dans le cadre d'un projet de Business Intelligence (BI) est une étape essentielle pour comprendre les types de données disponibles, évaluer leur qualité et déterminer leur accessibilité.

3.1.1 Source de données

- **Source Interne** : Notre principale source de données provient de l'application Thunder Express, une application de livraison de repas active sur le marché depuis deux ans. Elle est développée par l'équipe Webify. J'obtiens les données tout en respectant la confidentialité des informations de l'application.

la figure 3.1 montre le diagramme de classes de la base de données Thunder de notre source de données

- **Description des classes** :

Dans le tableau 3.9 ci-dessous on a fait la description des classes présentées dans le diagramme de classes de la base de données Thunder :

Classe	Description
Commande	La classe Commande représente la commande passé par le client ainsi que son statut, son fournisseur, et son livreur .
Fournisseur	La classe Fournisseur représente un fournisseur avec des attributs tels que l'identifiant, le nom, l'adresse et les produits fournis. Elle inclut des méthodes pour gérer les informations du fournisseur, ajouter ou supprimer des produits, et évaluer sa performance
Produit	La classe Produit représente un produit avec des attributs tels que l'identifiant, le nom, le prix le poid et les dimensions ainsi que le type du produit. Elle inclut des méthodes pour gérer les informations du produit , ajouter ou supprimer des produits.
Livreur	La classe Livreur représente un livreur avec des attributs tels que l'identifiant, le nom, le numéro de téléphone et la région. Elle inclut des méthodes pour gérer les informations du livreur, assigner des livraisons et suivre l'état des livraisons effectuées.
Client	La classe Client représente un client avec des attributs tels que l'identifiant, le nome et l'adresse. Elle inclut des méthodes pour gérer les informations du client, passer des commandes et consulter l'historique des achats.
Région	La classe Région représente les régions des clients et fournisseurs avec des attributs tels que l'identifiant, le nom , latitude et longitude.
Catégorie	La classe Catégorie représente les les domaines d'activité des fournisseurs avec des attributs tels que l'identifiant, le nom , description et image.
Menu	La classe Menu représente les menu préparés par chaque fournisseur avec des attributs tels que l'identifiant, le nom , description.
Options	La classe Options représente les options possibles pour chaque produits avec des attributs tels que l'identifiant, le nom et la description.
Favoris	La classe Favoris représente les fournisseurs favoris pour un client.
Type	La classe Type représente les types des produits avec des attributs tels que l'identifiant, le nom et la description.
Evaluation	La classe Évaluation représente l'évaluation du commande elle contient l'évaluation du commande et l'évaluation du livreur avec des attributs tels que l'identifiant et le rating.
Bonus	La classe Bonus représente le solde du bonus pour chaque client et inclut la source du bonus, le montant et la commande dans laquelle il a été appliqué
Historique	La classe Historique représente l'historique de changement de statut pour chaque commande.

TABLE 3.1 – Description de diagramme de classe

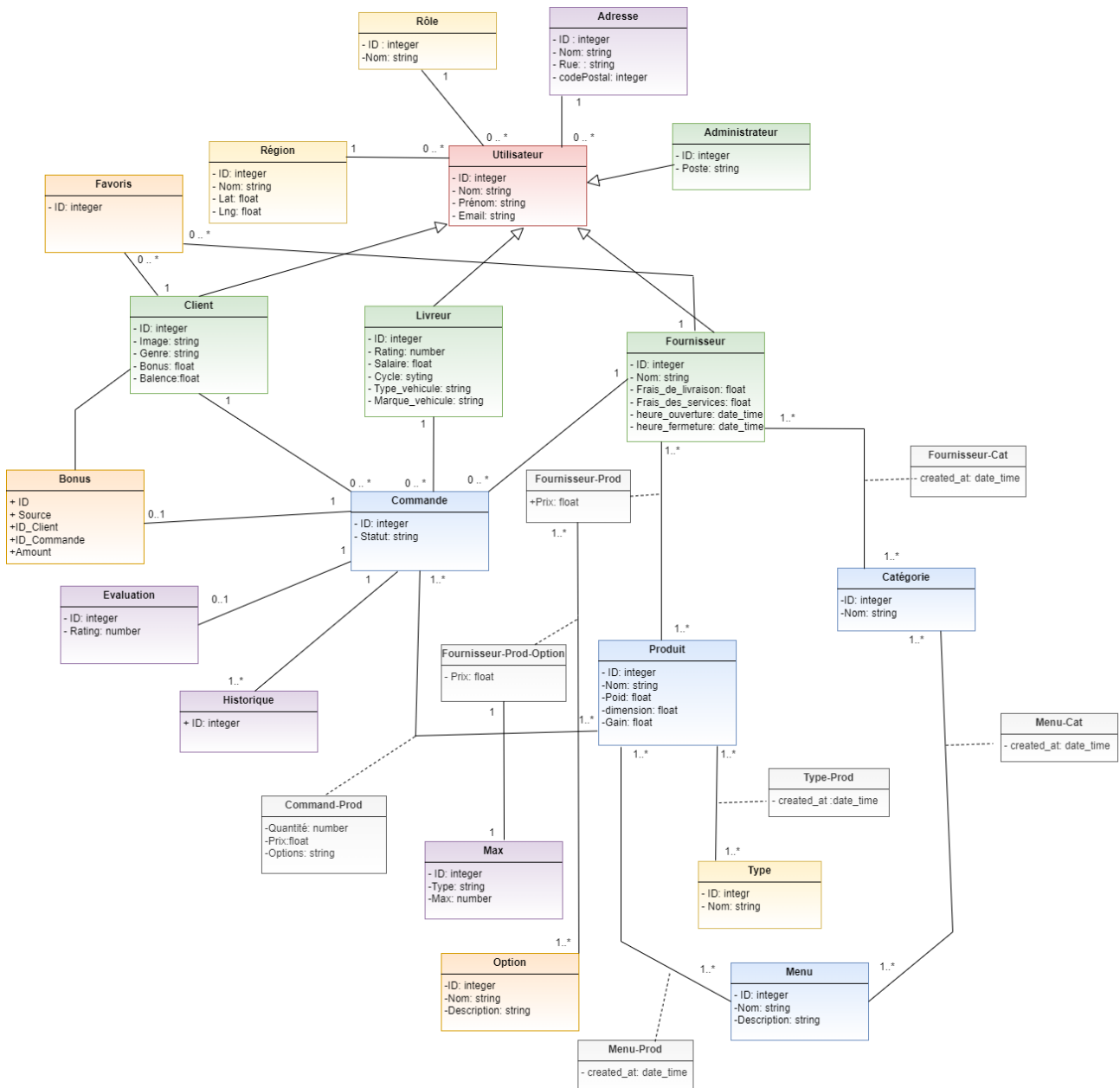


FIGURE 3.1 – Diagramme de classes Thunder Express

3.1.2 Description des données

— **Types de Données** : Données structurées à partir de bases de données relationnelles MySQL.

— **Granularité des Données** :

1. **Niveau de détail des données** : Nous avons utilisé les deux types de données.

Données Agrégées pour fournir une vue d'ensemble des performances de l'application, comme les ventes totales, les revenus moyens par jour, et les tendances des commandes.

Données Transactionnelles pour permettre des analyses approfondies, telles

que le suivi des commandes spécifiques, l'identification des catégories fréquentes, et la compréhension des comportements d'achat.

2. **Fréquence de mise à jour des données** : La fréquence de mise à jour des données varie : certaines sont collectées en temps réel et d'autres quotidiennement. Nous avons lancé un cron périodique pour intégrer ces nouvelles données.

— **Volume des Données** : Le volume de données utilisé est considérable. Nous avons utilisé une base de données contenant 8 000 clients, 25 000 commandes, 35 875 produits vendus, et 9 099 produits disponibles. Ce nombre augmente chaque jour, car l'application est en fonctionnement continu.

3.1.3 Qualité et Accessibilité des données

— **Précision** : Nos données sont exactes car elles proviennent de sources réelles. Cependant, nous rencontrons des problèmes tels que des informations manquantes sur le genre et l'âge des clients, ainsi que des avis sur les commandes et les livraisons, puisque tous les clients ne laissent pas leur avis. De plus, certains champs sont manquants, surtout pour les utilisateurs qui se connectent via Google et Apple. Nous constatons également des doublons dans les données pour les clients qui se connectent à chaque fois en utilisant une méthode différente, ainsi que pour les commandes annulées et relancées en raison d'erreurs.

— **Complétude** : La quantité de données manquantes est faible par rapport aux données attendues. Nous avons rencontré certaines lacunes dans notre travail, comme mentionné dans les risques, mais nous œuvrons activement à les corriger.

— **Actualité** : Le degré de fraîcheur des données est maintenu quotidiennement, car l'application est en fonctionnement continu. Pour nos KPI, nous avons besoin d'une actualisation périodique des données. C'est pourquoi nous utilisons des tâches cron pour automatiser cette mise à jour régulière.

— **Méthodes d'Accès** : Nous avons reçu les données et les avons extraites d'une base de données SQL fournie par la société Webiy, qui est responsable du développement de l'application Thunder Express.

— **Sécurité des Données** : La société nous a donné l'autorisation d'extraire les données sous des conditions juridiques précises, afin de ne pas accéder aux informations confidentielles des utilisateurs.

Une analyse complète des sources de données fournit une base solide pour la conception et le développement d'un système BI efficace, en garantissant que les bonnes données sont disponibles au bon moment pour prendre des décisions éclairées.

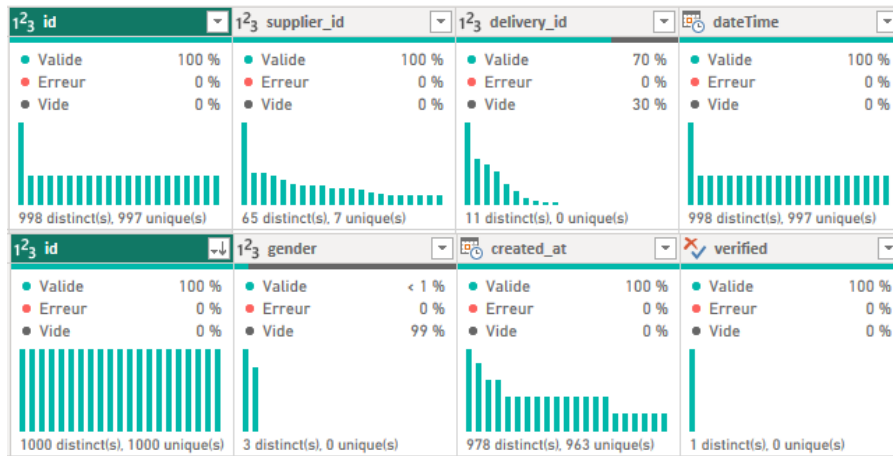


FIGURE 3.2 – Qualité des colonnes dans la base Thunder

3.2 Conception du Processus ETL

Le processus ETL (Extract, Transform, Load) est un élément clé de notre projet de Business Intelligence, permettant d'assurer une intégration efficace des données provenant de diverses sources. Cette section détaille chaque étape du processus ETL que nous avons mis en œuvre pour alimenter notre projet.

3.2.1 Extraction (Extract)

L'étape d'extraction consiste à collecter des données depuis la source de données, dans notre cas la première étape est la préparation de la source de données qui est la base de données MySQL Thunder qui contient tout les tables de l'applications Thunder Express. La deuxième étape à préparer est la connexion à la source de données. L'ajout de cette connexion se fait par l'ajout de la connexion Mysql via les outils d'administration et la connexion à la source Mysql via le gestionnaire de connexion dans Power Bi Desktop Figure 3.3. Enfin l'extraction des données brutes des sources dans un format intermédiaire ou temporaire, souvent appelé zone de staging.

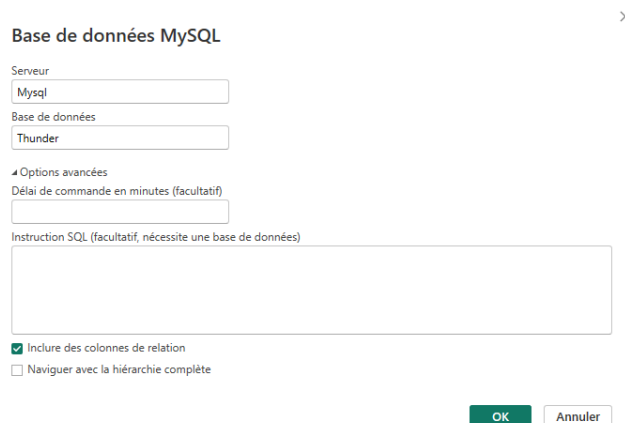


FIGURE 3.3 – Ajout de source de données Mysql

La troisième étape est la sélection des données pertinentes en fonction des besoins de notre projet BI et des spécifications de l'entrepôt de données Figure 3.4.

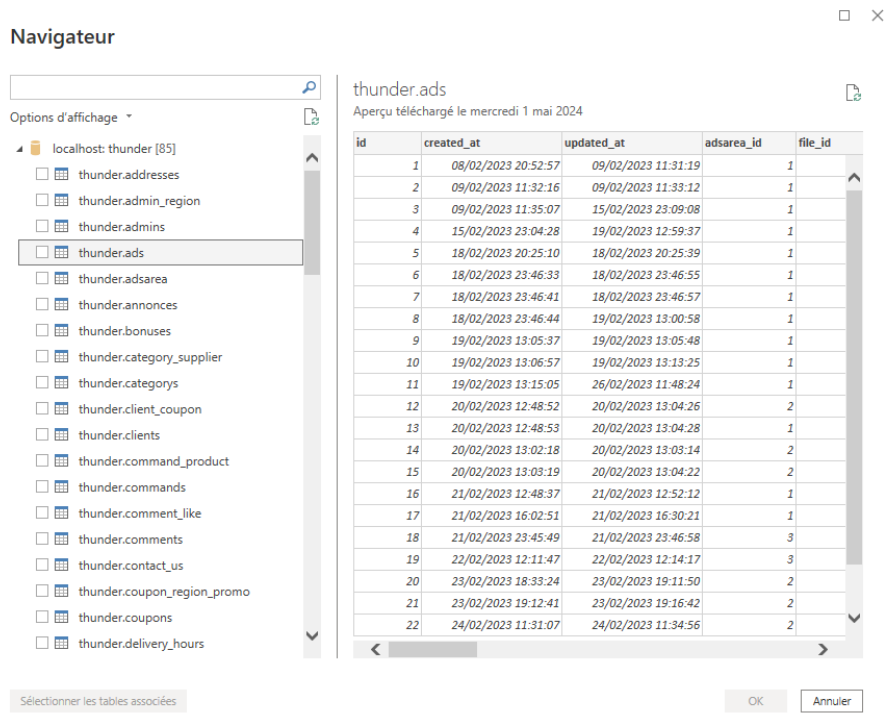


FIGURE 3.4 – Sélection des Données

3.2.2 Transformation (Transform)

La phase de transformation vise à préparer les données extraites pour l’analyse. Cela inclut plusieurs sous-étapes essentielles :

La première transformation que nous avons effectuée est le nettoyage des données, qui implique d’identifier et de corriger les erreurs, les valeurs manquantes et les doublons..

Figure 3.5

1	7	sousse	sousse	1
2	10	monastir	monastir	1
3	11	region de test sur p...	region_de_test_sur_p...	1
4	12	region	region	1
5	17	Hergla	Hergla	1
6	38	gjhkjk	gjhkjk	1
7	39	csqsc	csqsc	1
8	40	ssss	ssss	1
9	41	test	test	1
10	42	t01	t01	1
11	43	region040124	region040124	1
12	44	Tunis	Tunis	1
13	45	test01	test01	1
14	46	test2	test2	1
15	47	m	m	1
16	48	l	l	1
17	49	test danger	test_danger	1

FIGURE 3.5 – Transformation de données

La deuxième transformation est la normalisation des formats de données (par exemple, format de date) pour garantir l’homogénéité ensuite l’enrichissement tel que l’ajout de nouvelles colonnes dérivées. Figure 3.6

```

= Table.AddColumn("#Colonnes supprimées", "nameGender", each if [gender] = 1 then "Homme" else if [gender] = 2 then "Femme" else null)
= Table.ExpandTableColumn("#Requêtes fusionnées", "regions_delivery", {"name"}, {"regions_delivery.name"})
= Table.AddColumn("#delivery_rating développé", "TimeSlot", each if Time.Hour([date]) >= 0 and Time.Hour([date]) < 3 then "00:00-03:00"
= Table.AddColumn("#Personnalisée ajoutée", "DayOfWeek", each Date.DayOfWeekName([date]))
= Table.AddColumn("#Personnalisée ajoutée1", "HourOfDay", each Time.Hour(DateTime.Time([date])))
= Table.AddColumn("#Personnalisée ajoutée2", "total_price_without_delivery", each [total_price] - [delivery_price])
= Table.AddColumn("#Personnalisée ajoutée3", "total_price_with_gain", each [products_gain] + [total_price])
= Table.AddColumn("#Personnalisée ajoutée4", "profit", each [delivery_price] + [products_gain])
= Table.RenameColumns("#Personnalisée ajoutée5",{"date", "dateTime"})

```

FIGURE 3.6 – Normalisation de donnés

La troisième étape consiste à regrouper les données selon le niveau requis pour les rapports et analyses, que ce soit pour des périodes journalières, mensuelles ou trimestrielles. Enfin, on utilise le filtrage pour sélectionner ou exclure des données en fonction de critères spécifiques.

```

Saison =
SWITCH(
    TRUE(),
    MONTH([date]) IN {12, 1, 2}, "Hiver",
    MONTH([date]) IN {3, 4, 5}, "Printemps",
    MONTH([date]) IN {6, 7, 8}, "Été",
    MONTH([date]) IN {9, 10, 11}, "Automne",
    "Inconnue"
)

```

FIGURE 3.7 – Agrégation de donnés

3.2.3 Chargement des données (Load)

Une fois les données transformées, elles sont chargées dans l'entrepôt de données, prêtes à être utilisées pour les rapports, les analyses et les visualisations dans le système BI.

Pour la table commande nous avons choisis le chargement Incrémental. Ce mode consiste à charger uniquement les données qui ont changé depuis la dernière opération de chargement. Cela peut inclure les nouvelles données ajoutées ou les données modifiées. Ce type de chargement est généralement plus efficace que le chargement complet, car il réduit le volume de données traitées.

Pour les autres tables on a choisis le chargement Complet (Full Load) qui est le mode de chargement par défaut dans power Bi. Dans ce mode, l'ensemble des données est chargé dans l'entrepôt de données à chaque exécution. Cela signifie que toutes les données, même celles déjà présentes, sont extraites et insérées à nouveau.

FIGURE 3.8 – Chargement incrémental de la table commandes

3.2.4 Scripts ETL

Le processus ETL est généralement itératif et cyclique, s'exécutant régulièrement pour maintenir la cohérence et la fraîcheur des données dans l'entrepôt de données. Il peut être planifié selon un calendrier prédéfini ou déclenché par des événements spécifiques, tels que la mise à jour des données sources.

En résumé, le processus ETL permet de collecter, nettoyer, transformer et charger les données de manière efficace et fiable, garantissant ainsi la qualité et l'intégrité des données dans l'entrepôt de données, prêtes à être exploitées pour l'analyse et la prise de décision.

3.3 Modèle des données

3.3.1 Approche Dimensionnelle : Dimensions et Faits

Dans cette section notre objectif est de proposer une méthode de modélisation faits/dimensions répondant aux mieux aux besoins des utilisateurs pour leur faciliter l'exploitation des données stockées dans un entrepôt de données.

Tout le long de notre conception nous suivrons la démarche de « Ralf Kimball » qui est la plus adaptées à notre cas et qui se résume en Quatre étapes :

1-Sélection du processus d'activités à modéliser

Un processus est une activité opérationnelle effectuée dans une organisation qui est généralement pris en charge par une source de gestion et collecte de données. Pour notre cas Le processus clé est : le suivi des commandes

2-Choix de grain du processus d'activités

La définition de la granularité de la table de fait, revient à spécifier exactement ce que

représente une ligne de cette table. Et ce, en déclarant l'expression la plus fine qui contiendra le besoin de l'utilisateur de manière plus détaillée

3-Choisir les dimensions participantes à ce processus

Les dimensions fournissent les questions suivantes : "qui fait quoi, quand, pourquoi, et comment" aux alentours du contexte d'un événement particulier d'un processus donné. Les tables de dimensions contiennent les attributs descriptifs pour définir ce processus. Après le choix de la granularité, l'identification des dimensions est assez simple. Selon Kimball : « Le temps est la seule dimension qui figure systématiquement dans tout un entrepôt de données , et plus souvent la première dimension dans le classement sous-jacent de la base de données ». C'est pour cela que la table de dimension est appelée quelquefois l' "âme" d'un entrepôt de données .

4-Identifier les faits numériques

Les faits sont les mesures qui résultent d'un processus d'un événement particulier. Les faits sont généralement des quantités numériques additives telles que le chiffre d'affaire qui répondent à la question qu'est-ce qu'on mesure?". Les décideurs sont vivement intéressés par l'analyse de ces mesures afin de déterminer la performance du processus concerné

Dimensions recueillies :

Dimension CLIENT :

Attribut	Description
ID_Client	Clé de dimension client.
Gender	Genre du client.
Name	Nom du client.
Created_at	Date d'ajout du client.
Bonus	Total bonus du client.
Region	Region du client.

TABLE 3.2 – Description de la dimension client

Dimension DELIVERY :

Attribut	Description
ID_Delivery	Clé de dimension delivery.
Name	Nom du client.
Region	Region du client.
Start_worktime	Temps de début du travail.
End_worktime	Temps de fin du travail.
Vehicule	Type de vehicule de livreur.
Rating	Evaluation du livreur (entre 0 et 5).

TABLE 3.3 – Description de la dimension delivery

Dimension DELIVERY_HOURS :

Attribut	Description
ID_Delivery_hours	Clé de dimension delivery_hours.
ID_Delivery	Nom du livreur.
Hours	Total heures travaillés dans la journée.
Start_hour	Heure de début du journée.

End_hour	Heure de fin du journée.
Date	Date de la journée.

TABLE 3.4 – Description de la dimension delivery_hours

Dimension SUPPLIER :

Attribut	Description
ID_SUPPLIER	Clé de dimension fournisseur.
Region	Région du fournisseur.
Start_time	Temps d'ouverture.
Close_time	Temps de fermeture.
Name	Nom du fournisseur.
Stars	Évaluation du fournisseur (entre 0 et 5).

TABLE 3.5 – Description de la dimension fournisseur

Dimension CATEGORY :

Attribut	Description
ID_CATEGORY	Clé de dimension category.
ID_SUPPLIER	Clé de dimension fournisseur.
Name	Nom de la catégorie.
Menu	Menu de la catégorie.

TABLE 3.6 – Description de la dimension category

Dimension PRODUCT_SUPPLIER :

Attribut	Description
ID_PRODUCT	Clé de dimension product_fournisseur.
ID_SUPPLIER	Clé de dimension fournisseur.
Name	Nom du produit.
Price	Prix du produit.
Type	Type du produit.

TABLE 3.7 – Description de la dimension product_fournisseur

Dimension PRODUCT_COMMAND :

Attribut	Description
ID_PRODUCT_COMMAND	Clé de dimension product_command.
ID_SUPPLIER	Clé de dimension fournisseur.
Name	Nom du produit.
Price	Prix du produit.
Quantity	Quantité du produit dans la commande.
Options	Options du produit dans la commande.
Type	Type du produit.

TABLE 3.8 – Description de la dimension product_command

Table de faits :

Attribut	Description
ID_COMMAND	Clé de table command.
Adress	Adresse de la commande.
Bonus	Bonnuus appliqué sur la commande.
ID_client	Id de la dimension client.
Date	Date de la commande.
Cycle	Statut de la commande.
ID_delivery	ID de la dimension delivery.
Mode_pay	Mode de paiement de la commande.
Total_price	Prix total de la commande.
Delivery_price	Prix de livraison la commande.
Type	Type de la commande(delivery or emported).

TABLE 3.9 – Description de la table de fait command

3.3.2 Shémas des données

Modèle flocon de neige :

La figure ci dessous represente le modèle phoysique de notre datamartavec les différents dimensions et la table de fait FACT_Command.

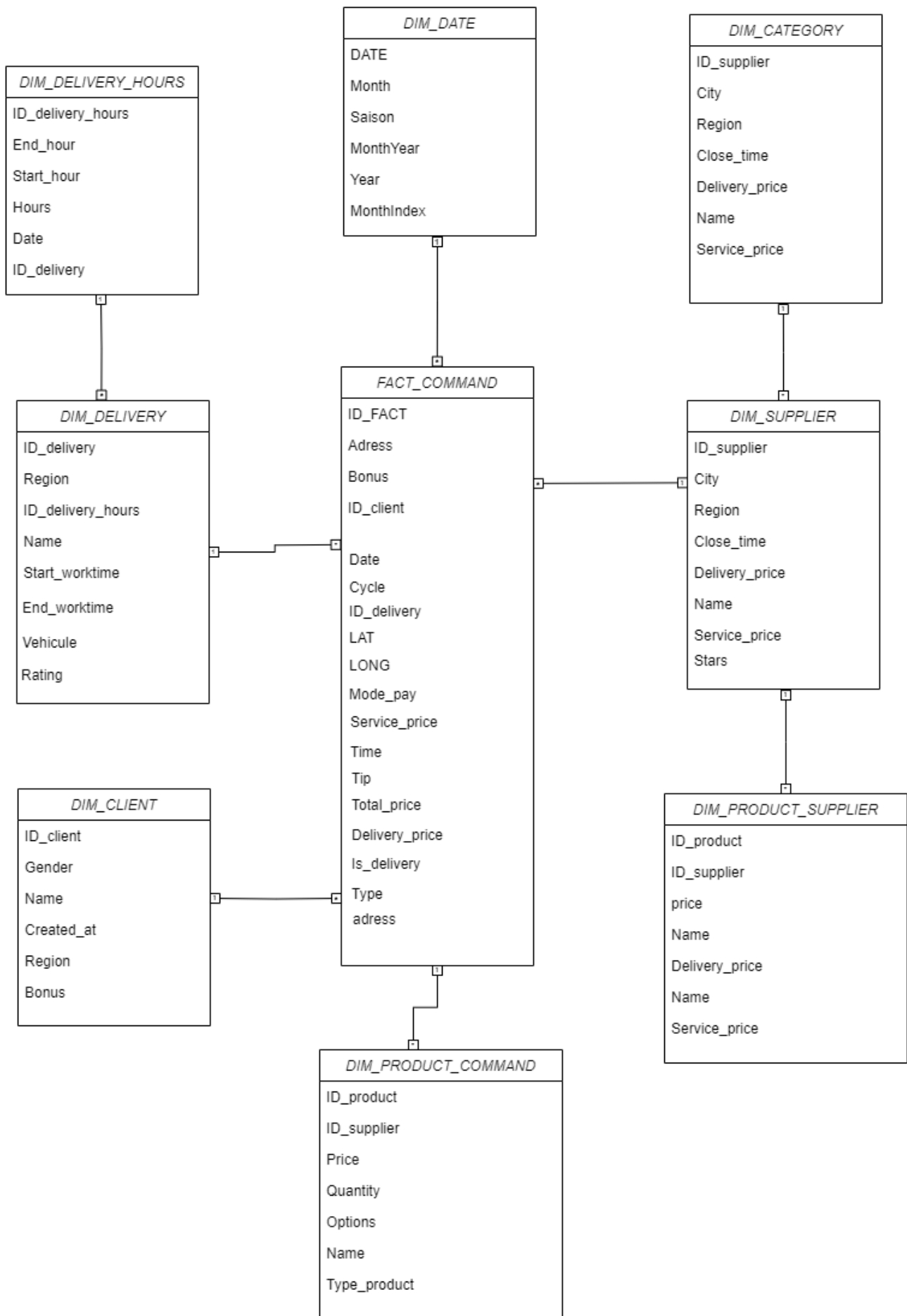


FIGURE 3.9 – Modèle en flocon du datamart Command

3.3.3 Mesures DAX

Dans le cadre de notre processus ETL, nous avons également créé plusieurs mesures DAX (Data Analysis Expressions) pour enrichir nos analyses. Ces mesures ont été conçues pour calculer des indicateurs clés de performance, tels que le total des ventes, le pourcentage de gain et le taux de satisfaction client etc ...

En utilisant DAX, nous avons pu réaliser des calculs dynamiques basés sur les données chargées, permettant ainsi des analyses approfondies et des visualisations interactives. Ces mesures facilitent l'exploration des tendances et des performances des commandes, offrant aux utilisateurs des insights précieux pour prendre des décisions éclairées. Grâce à l'intégration de ces mesures dans notre modèle de données, nous avons optimisé la capacité d'analyse de notre système BI tout en assurant une expérience utilisateur fluide.

Taux de satisfaction client :

```
TauxSatisfaction =
VAR TotalAvis = COUNTROWS(commands)
VAR AvisPositifs = COUNTROWS(FILTER(commands, commands[rating] >= 4))
RETURN
DIVIDE(AvisPositifs, TotalAvis, 0)
```

Pourcentage du gain :

```
Pourcentage gain = DIVIDE(SUM(commands[products_gain]), SUM(commands[total_price]))
```

Nombre de commandes par fournisseur :

```
Nombre Commandes Fournisseur = COUNTROWS(RELATEDTABLE(commands))
Pourcentage Commandes = DIVIDE([Nombre Commandes Fournisseur], [Total commandes], 0)
```

```
Durée Traitement (heures) = AVERAGEX(commands, DATEDIFF(commands[created_at], commands[updated_at], HOUR))
```

Total livraisons :

```
Livraisons par Heure = COUNTROWS(FILTER(commands, HOUR(commands[HourOfDay]) = [Heure Sélectionnée]))
Livraisons par Jour = COUNTROWS(FILTER(commands, commands[DayOfWeek] = [Date Sélectionnée]))
```

Taux de réussite de commandes :

```
Commandes Réussites = COUNTROWS(FILTER(commands, commands[cycle] = "SUCCESS"))
Taux de Réussie = DIVIDE([Commandes Réussites], [Total Commandes], 0)
```

Taux d'annulation de commandes :

```
Commandes Retournées = COUNTROWS(FILTER(commands, commands[cycle] = "REJECTED"))
Taux de Retour = DIVIDE([Commandes Retournées], [Total Commandes], 0)
```

3.4 Conception des applications

3.4.1 Conception des Rapports et des Tableaux de Bord

Définition des besoins en termes de visualisation des données

La première étape de la conception des rapports et des tableaux de bord consiste à identifier et à définir les besoins spécifiques des utilisateurs. Nous avons réalisé des entretiens avec les parties prenantes pour recueillir leurs attentes concernant les indicateurs clés de performance (KPI) et les données nécessaires à une prise de décision éclairée. Les besoins identifiés incluent :

- **Suivi des commandes** : Les utilisateurs souhaitent visualiser les ventes par produit, par fournisseur, par livreur, par région et par période.
- **Satisfaction client** : Les indicateurs liés à la satisfaction des clients, tels que le taux de retour et le délai de livraison, doivent être facilement accessibles.
- **Suivi des performances des livreurs et fournisseurs** : Les utilisateurs souhaitent visualiser les ventes, par fournisseur, par livreur et leurs évaluations, les domaines d'activités les plus demandés en regroupant par plages horaires et par genre.
- **Suivi des produits les mieux vendus** : Les utilisateurs souhaitent visualiser les produits les mieux vendus en indiquant les heures de pic de vente pour chaque produit et fournisseur.
- **Suivi des heures et jours de pic des livraisons** : Les utilisateurs souhaitent savoir une vue globale pour suivre le chiffre d'affaire, la satisfaction globale aux commandes et aux livraisons et le pourcentage global de gain.
- **Suivi de chiffre d'affaires** : Les utilisateurs souhaitent savoir les heures et les jours de pic des livraisons.
- **Comparaison entr volume vs valeurs des commandes** : Les utilisateurs souhaitent savoir une vue globale de comparaison entre la valeur et le volume de commandes avec le pourcentage du gain pour chaque fournisseur et leur moyenne de prix de produits ainsi que leur volume des produits et leur temps de préparation moyenne.
- **Suivi de performances des livreurs** : Les utilisateurs souhaitent savoir les performances de chaque livreur par la comparaison des heures travaillées dans la journée par rapport au nombre de commandes livrées.

Conception des maquettes (wireframes)

Après la définition des besoins, des maquettes (wireframes) des rapports et tableaux de bord ont été créées. Ces maquettes présentent l'interface utilisateur et l'agencement des éléments. Les wireframes incluent :

- **Rapports dynamiques** : Des visualisations graphiques interactives des ventes, avec des filtres permettant de segmenter les données par période, région, produit, et fournisseur.

- **Tableaux de bord interactifs** : Ces tableaux intègrent des éléments de navigation facilitant l'accès aux différentes sections. Les filtres sont synchronisés, permettant une mise à jour automatique des visualisations en fonction des sélections effectuées par l'utilisateur.
- **Segments et filtres** : Les maquettes montrent comment les utilisateurs peuvent appliquer des segments et des filtres pour personnaliser l'affichage des données. Les filtres sont conçus pour être facilement ajustables et peuvent inclure des critères tels que la période de temps, les catégories de produits, ou le fournisseur.
- **Synchronisation des segments** : Pour assurer une analyse cohérente, les filtres appliqués dans un segment du tableau de bord se synchronisent automatiquement avec les autres segments. Par exemple, si un utilisateur filtre les ventes par région dans un graphique, ce filtre sera appliqué à tous les autres graphiques et rapports affichés simultanément.
- **Influenceurs clés** : Des sections spécifiques des maquettes mettent en avant les influenceurs clés des performances, tels que les tendances de chiffre d'affaires et du commadnes. Ces influenceurs permettent une identification rapide des facteurs qui impactent les résultats globaux.
- **Champs d'extraction** : Les wireframes incluent des champs d'extraction définis pour extraire des données spécifiques des bases de données. Ces champs permettent aux utilisateurs de sélectionner les dimensions et les mesures pertinentes pour générer des rapports détaillés.
- **Paramètres de personnalisation** : Les maquettes illustrent les paramètres disponibles pour personnaliser les rapports et les tableaux de bord. Les utilisateurs peuvent ajuster des paramètres tels que les plages de dates, les seuils de performance, ou les seuils d'alerte pour adapter les visualisations à leurs besoins spécifiques.

Ces éléments sont essentiels pour créer des outils de reporting et de visualisation de données qui répondent aux besoins variés des utilisateurs et facilitent une analyse approfondie et personnalisée des données.

3.4.2 Développement de prototypes BI

Une fois les maquettes validées, nous avons procédé au développement de prototypes de Business Intelligence (BI). Ces prototypes intègrent des outils tels que Microsoft Power BI et Tableau pour créer des rapports et des tableaux de bord interactifs. Les principales fonctionnalités développées incluent :

- **Intégration des données** : Connexion aux sources de données existantes (bases de données Thunder) pour assurer une actualisation en temps réel.
- **Visualisations personnalisables** : Options permettant aux utilisateurs de personnaliser les visualisations en fonction de leurs besoins spécifiques..
- **Rapports automatisés** : Génération automatique de rapports périodiques pour faciliter le suivi des performances sans intervention manuelle.

Sur la base des retours des utilisateurs, des ajustements ont été réalisés pour améliorer l'interface et la fonctionnalité des rapports et des tableaux de bord, garantissant ainsi qu'ils répondent aux besoins de l'ensemble des utilisateurs finaux. Cette approche itérative assure un produit final de qualité, adapté aux exigences du projet de notre projet.

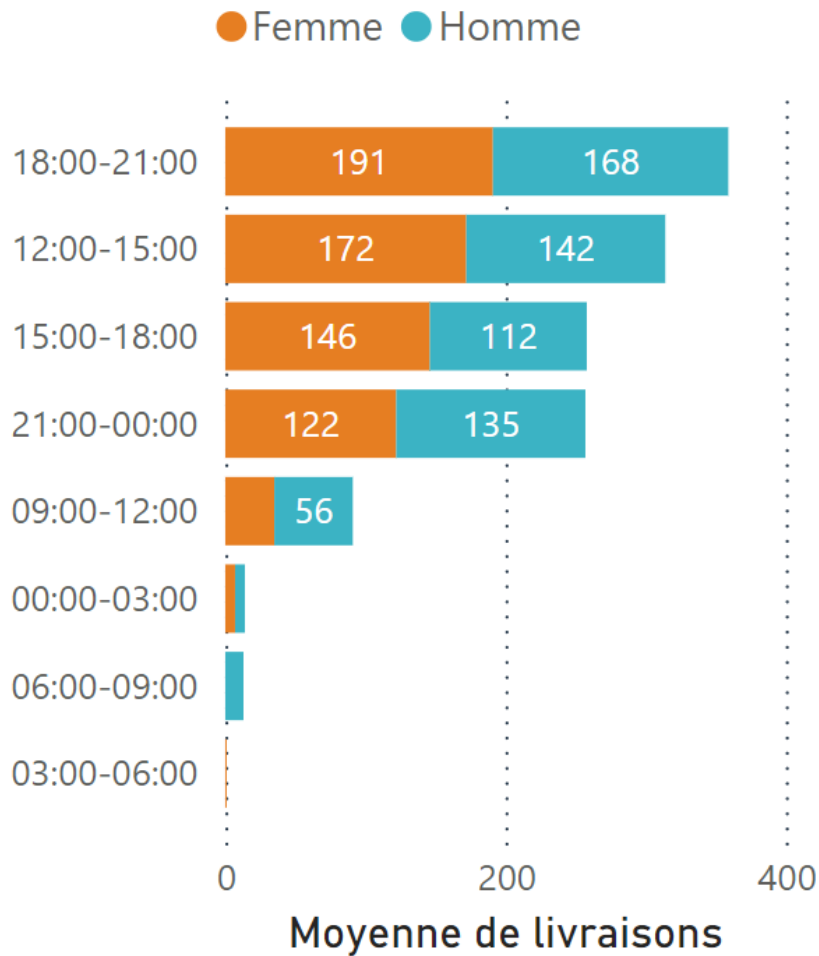


FIGURE 3.10 – Prototype : Plages horaires de livraisons

3.4.3 Tests de prototypes (Ajustements basés sur les retours des utilisateurs)

La phase de test des prototypes est cruciale pour garantir leur efficacité et leur convivialité. Des sessions de tests ont été organisées avec un groupe d'utilisateurs représentatifs, permettant de recueillir des retours sur plusieurs aspects des prototypes.

- **Facilité d'utilisation** : Les utilisateurs ont évalué l'ergonomie du visuel, indiquant que la navigation était intuitive et que les visualisations étaient claires.
- **Satisfaction des utilisateurs** : Les premiers retours ont montré un taux de satisfaction élevé concernant les fonctionnalités offertes par le visuel. Les utilisateurs ont apprécié la personnalisation des visualisations et la pertinence des données présentées. Cette satisfaction témoigne de l'adéquation du visuel avec leurs besoins opérationnels.
- **Pertinence des données** : Les participants ont confirmé que les visualisations fournissaient des informations utiles et adaptées à leurs attentes, renforçant ainsi leur confiance dans l'outil développé.
- **Bugs et améliorations** : Bien que le visuel ait été bien accueilli, des améliorations ont été suggérées pour optimiser certaines fonctionnalités. Les retours ont permis

d'identifier quelques problèmes techniques telque le pourcentage des clients qui remplissent leurs genres et leurs date de naissance est très faibe par rapport au nombre total des clients ce qui ne permet pas d'obtenir des résultats précis pour les kpis qui on relation avec le genre ou l'age du client.

Un autre problème technique à été remarqué dans cette étape est il y a un manque dans les heures de travail des livreurs.

Sur la base de ce problème, on a proposé aux décideurs d'ajouter une interface incitant les clients à compléter leurs informations personnelles en gagnant des points bonus.

Pour le deuxième problème, notre proposition est d'inclure un système de notification pour rappeler l'administrateur de remplir les heures de travail des livreurs à la fin de la journée. Cela permettra d'améliorer la qualité des données et, par conséquent, la pertinence des tableaux de bord .

Les deux propositions ont été très bien accueillie par les décideurs et nous parlerons de leur mise en œuvre dans le chapitre suivant.

3.5 Environnement technique

Nous présentons dans cette section les choix techniques relatifs à l'environnement matériel et logiciel qui ont contribué à la réalisation de notre projet.

3.5.1 Environnement matériel

Au cours des différentes étapes de notre projet, à savoir la documentation, l'implémentation du code et les tests, nous avons disposé de :

Machine	Ordinateur Portable
Marque	Lenovo
Processeur	Intel Core I7
RAM	8 Go
Système d'exploitation	Windows 10

TABLE 3.10 – Matériel de base

3.5.2 Environnement logiciel

Tout au long de la phase de développement, nous avons utilisé les outils logiciels et les langages de programmation suivant :

1. Éditeurs de développement :

- **Visual Studio Code** : est un éditeur de code source autonome qui s'exécute sur Windows, macOS et Linux. Le meilleur choix pour JavaScript et les développeurs web, avec des extensions pour prendre en charge à peu près n'importe quel langage de programmation.[\[9\]](#)

Nous avons utilisé ce logiciel lors du développement de parties web et mobile.



FIGURE 3.11 – Logo Visual Studio Code

2. Langages de programmation :

- **DAX** : (Data Analysis Expressions) est une bibliothèque de fonctions et d'opérateurs qui peuvent être combinés pour générer des formules et des expressions dans Power BI, Analysis Services et Power Pivot dans les modèles de données Excel.[10]



FIGURE 3.12 – Logo DAX

- **PHP** : est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML.[11] Nous l'avons utilisé également avec Laravel pour la partie backend.



FIGURE 3.13 – Logo PHP

- **TypeScript** :est un langage de programmation fortement typé qui s'appuie sur JavaScript et offre de meilleurs outils à n'importe quelle échelle.[12] Nous l'avons également utilisé avec Angular pour la partie frontend.



FIGURE 3.14 – Logo TypeScript

- **Python** : est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété.[13] Nous avons utilisé pour la réalisation du module intelligent.



FIGURE 3.15 – Logo Python

3. Système de gestion de base de données :

- **My SQL** : est un système de gestion de base de données relationnelle (SGBDR) open source avec un modèle client-serveur basé sur le langage de requête structuré SQL.[Mysql]



FIGURE 3.16 – Logo MySQL

4. Structure de développement :

- **Microsoft Power BI** : Power BI est une suite d'outils d'analyse commerciale offrant des insights dans l'ensemble de l'organisation. Power BI est utilisé dans notre projet comme outil de reporting. Le service Power BI a une version gratuite et une version Pro. En effet, Power BI se compose d'une application de bureau Windows appelée **Power BI Desktop**, d'un service SaaS (Software as a Service) en ligne appelé **service Power BI**.



FIGURE 3.17 – Logo Power BI

Power BI Desktop est un outil de création d'applications web hybride de données et de rapports riches en fonctionnalités. La combinaison des données provenant de plusieurs bases de données, fichiers et services web à l'aide d'outils visuels nous permettent de comprendre et de corriger automatiquement la qualité des données et les problèmes de mise en forme. Avec plus de 20 éléments visuels intégrés et un ensemble riche de visualisations personnalisées, nous pouvons créer d'incroyables rapports.

L'utilisation du service Power BI permet de publier des rapports en toute sécurité à l'organisation et configurer une actualisation automatique des données afin que

tout le monde dispose des informations les plus récentes.[14]

Power BI peut unifier toutes les données de l'organisation, qu'elles se trouvent dans le cloud ou qu'elles soient stockées localement. Les passerelles Power BI permettent aux utilisateurs de connecter des bases de données SQL Server, des modèles Analysis Services et de nombreuses sources de données. Bénéficiez d'une scalabilité à l'échelle de l'entreprise, avec une gouvernance et une sécurité intégrées.

Une passerelle est un logiciel qui facilite l'accès aux données résidant sur un réseau local privé en vue de leur utilisation ultérieure dans un service cloud tel que Power BI.[15]

Nous avons choisi d'utiliser ces outils BI car il offre différentes fonctionnalités comme l'indique le tableau suivant :

Rôle	Nom de l'outil	Fonctionnalités
Reporting	Power BI Desktop	<ul style="list-style-type: none"> — Connexion aux différentes sources de données avec une vue des tables et des relations. — Utilisation du langage DAX pour créer des mesures. — Création des rapports interactifs.
Pont entre les données locales et les données en cloud	Gateway on premises	<ul style="list-style-type: none"> — configurer une actualisation automatique des données.
Administration	Service Power BI	<ul style="list-style-type: none"> — Publier des rapports en toute sécurité. — Accès aux différents rapports. — Connexion via un email-professionnel.

TABLE 3.11 – Les fonctionnalités des outils sélectionnés

- **Laravel** : est un framework PHP multi-plateforme permettant de créer des applications web. Laravel est principalement un framework de développement Backend, bien qu'il offre quelques fonctionnalités Frontend. Cependant, de nombreuses fonctionnalités de Laravel sont agnostiques au niveau du Frontend.[16] Nous l'avons utilisé pour la partie backtend.

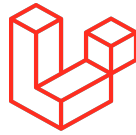


FIGURE 3.18 – Logo Laravel

- **Angular** : est un framework JavaScript développé par Google. Il est utilisé pour développer des applications web et mobile. Avec cette technologie, on réalise des interfaces de type monopage qui fonctionnent sans rechargement de la page web.[17]

Nous avons employé cet outil pour développer l'infrastructure frontend de notre partie web.



FIGURE 3.19 – Logo Angular

5. Outils informatiques :

- **GitLab** : c'est une plateforme permettant d'héberger et de gérer des projets web de A à Z. Présentée comme la plateforme des développeurs modernes, elle offre la possibilité de gérer ses dépôts Git et ainsi de mieux appréhender la gestion des versions de vos codes sources.[18]



FIGURE 3.20 – Logo GitLab

- **Draw.io** : est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre.[19]



FIGURE 3.21 – Logo Draw.io

6. Outil de rédaction du rapport :

- **Overleaf** : est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX sans aucun téléchargement d'application. En outre, elle offre la possibilité de rédiger des documents de manière collaborative.[20]



FIGURE 3.22 – Logo Overleaf

Conclusion