



Université de Tunis El Manar



Faculté des Sciences Economique et de
Gestion de Tunis

Rapport de projet de fin d'étude

Présenté en vue de l'obtention du Diplôme de licence en Business
Intelligence

Mise en place d'une Solution Business Intelligence pour le suivi de l'application **MY BIAT**

Réaliser au sein de la Banque Internationale Arabe de Tunisie



Réaliser par : Samah Ben Ghars

Encadrant pédagogique

Mme Marah Altaher

Encadrant professionnel

Mr Slim Saidi

Année Universitaire : 2023_2024

Remerciements

Mes remerciements vont à tous ceux qui ont bien voulu m'aider afin de réaliser ce projet :

Je tiens d'abord, à remercier tout le corps professoral de la faculté des sciences économiques et de gestion tunis el manar qui durant ces années, nous ont encadré afin d'obtenir ce diplôme ; particulièrement ma directrice de stage madame Marah AL taher .

Nous la remercions pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur. Je tiens à remercier aussi mon encadrant professionnel monsieur Slim Saidi pour l'aide et les conseils concernant les missions évoquées dans ce rapport.

Dédicaces

A mon père & ma mère Qui ont été toujours à côté de moi, qui m'ont éclairé le chemin et m'ont Encouragé. et soutenu tout au long de mes études. C'est à la cause de leurs sacrifices, et leurs conseils prodigues, et leur patience Que j'ai pu atteindre cette étape de ma vie. A tous mes amis qui sont chères Pour leur amour et leur sympathie qui ont été une aide précieuse Je dédie ce travail

Table des matières

Table des figures	vi
Liste des tableaux	viii
Introduction générale	1
1 Cadre général du projet	3
Introduction	4
1.1 Cadre du projet	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Organigramme	5
1.2.2 Présentation de l'équipe de travail	5
1.3 Présentation du projet	5
1.4 Etude de l'existant	6
1.4.1 Problématique	6
1.4.2 Solution proposée	7
1.5 Méthodologie de gestion de projet	7
1.5.1 Méthode Bill Inmmon	7
1.5.2 Méthode ralph kimball	8
1.5.3 bottom_up vs Top_Down	9
1.5.4 Choix de l'approche : Ralph Kimball	9
1.6 Les phases d'un projet BI selon l'approche de Ralph Kimball	10
1.7 Environnement de développement	13
1.7.1 Environnement logiciel	13
1.7.2 les langages de programmation	16
1.7.3 Environnement matériel Requis	16
1.7.4 Quels protocoles de communication allons-nous utiliser ?	18
1.7.5 L'architecture proposée de la solution	18
Conclusion	20

TABLE DES MATIÈRES

2 Étude Préalable Et Spécification Des Besoins	21
Introduction	22
2.1 planification de projet	22
2.2 Concepts de base de la Buisness Intelligence	22
2.2.1 Les intérêts du Système décisionnel (BI) pour le secteur bancaire . .	22
2.2.2 L'informatique décisionnelle (BI)	23
2.2.3 Le processus d'intégration de données dans un entrepôt de données (ETL)	23
2.2.4 Entrepôt de données (DWH)	24
2.2.4.1 Caractéristiques clés d'un entrepôt de données	24
2.2.5 La différences entre un Data Warehouse et un Data Mart	25
2.2.6 Notions des tables de faits et de dimensions	26
2.3 Spécification des besoins	26
2.3.1 Identification des acteurs	27
2.3.2 Identification des besoins fonctionnels	27
2.3.3 Identification des besoins non fonctionnels	27
2.4 Diagramme de cas d'utilisation global	28
Conclusion	28
3 Conception et Mise En Place Du Data Warehouse	29
Introduction	30
3.1 La modélisation dimensionnelle du Data Warehouse	30
3.2 Identification des faits et des dimensions	33
3.2.1 Choix des dimensions	33
3.2.2 Choix des indicateurs	35
3.3 Modélisation du Data Warehouse	39
3.3.1 Modèles en Constellation	39
3.4 Mise En Place Du Data warehouse	41
3.4.1 Data Warehouse my_biat_dwh	41
3.4.2 Diagramme my_biat_dwh	42
Conclusion	42
4 Mise En oeuvre ETL	43
Introduction	44
4.1 Présentation de Apache Airflow	44
4.1.1 C'est quoi une Dag (Directed Acyclic Graph) ?	45
4.1.1.1 C'est quoi une tache (task) ?	46
4.1.1.2 Le cycle de vie d'une tâche	47
4.2 Instalation de l'environnement	48
4.2.1 Création d'une machine virtuelle	48

TABLE DES MATIÈRES

4.2.2	Autorisation des protocoles	48
4.2.3	Instalation de docker compose , Airflow et PostgreSQL sur la machine	49
4.3	Développement de DAG	49
4.3.1	L'interface web d'Airflow	56
4.3.2	La connexion entre Airflow et PostgreSQL	56
4.3.3	Transfert des données de airflow à l'entrepôt de données	57
4.3.4	Remplissage de l'entrepôt de données	58
4.3.5	Les commandes utilisées :	58
	Conclusion	60
5	Déploiement De Tableaux De Bord	61
	Introduction	62
5.1	Objectifs	62
5.2	Connexion et importation de données	63
5.3	Tableaux de bord	64
5.3.1	Page d'accueil	64
5.3.2	Tableaux de bord du nombre de souscriptions	64
5.3.3	Tableaux de bord du nombre de clients	65
5.3.4	Tableaux de bord du conquête et le types d'accées	65
5.3.5	Tableaux de bord segments clients	66
5.3.6	Tableaux de bord des demande de chequier	66
5.3.7	Tableaux de bord des objectifs par rapport aux réalisations	67
	Conclusion	67
	Conclusion générale	68
	Bibliographie	69

Table des figures

1.1	Organigramme BIAT	5
1.2	Approche Top-Down	8
1.3	Approche Bottom-Up	8
1.4	Cycle de vie de la méthodologie Kimball	10
1.5	logo PostgreSQL	13
1.6	logo Dbeaver	13
1.7	logo Apache Airflow	14
1.8	logo : Docker	14
1.9	logo VS Code	15
1.10	logo Power AMC	15
1.11	logo Power BI	15
1.12	logo Overleaf	16
1.13	logo : Microsoft Azure	17
1.14	machine virtuelle	17
1.15	notre machine virtuelle	18
1.16	Architecture de la solution	19
1.17	La chaîne d'information décisionnelle	20
2.1	Gantt Chart	22
2.2	Processus De Business intelligence	23
2.3	Processus ETL	24
2.4	Data Mart vs Data Ware House	26
2.5	Diagramme de cas d'utilisation	28
3.1	Cycle de vie de la méthodologie Kimball	30
3.2	Modèle en étoile	31
3.3	Modèle en flacons	31
3.4	Modèle en constellation	32
3.5	modèles en constellation	40
3.6	my_biat_dwh	41
3.7	Digrame 1 : Modèle en Constellation	42

TABLE DES FIGURES

4.1	Cycle de vie de la méthodologie Kimball	44
4.2	architecture Airflow	44
4.3	Présentation Apache Airflow	45
4.4	Work Flow : flux de travail	45
4.5	DAG : graphiques de cycle dirigés	46
4.6	Cycle de vie d'une tache	47
4.7	Notre machine virtuelle	48
4.8	réseaux autorisé	48
4.9	Default_args	49
4.10	get_conn	50
4.11	delete_trans_tables_content	50
4.12	create_tables	51
4.13	insert_query	51
4.14	process_excel	52
4.15	DAG	53
4.16	postgres_create_query	54
4.17	table_meta	55
4.18	connexion	56
4.19	succès	57
4.20	les tâches	57
4.21	Transfert des données	58
5.1	Diagramme de cas d'utilisation	62
5.2	connexion a l'entrepôt de données	63
5.3	importation de données	63
5.4	HOME	64
5.5	TDB nombre de souscriptions	64
5.6	TDB nombre de client	65
5.7	TDB types d'accès et la conquête client	65
5.8	TDB segment client	66
5.9	TDB demande chéquier	66
5.10	TDB réalisations par rapport aux objectifs	67

Liste des tableaux

1.1	étude comparative de ces deux approches actuelles	9
1.2	Configuration requise	16
2.1	Comparaison entre un Data Warehouse et un Data Mart	25
3.1	dim_agence : dimension agence	33
3.2	dim_zone : dimension zone	33
3.3	dim_region : dimension region	33
3.4	dim_client : dimension client	34
3.5	dim_souscription : dimension souscription	34
3.6	dim_date_authentification : dimension date authentification	34
3.7	dim_date_chequier : dimension date chequier	35
3.8	dim_virement : dimension date virement	35
3.9	fact_nb_souscrit : fact nombre de souscriptions	36
3.10	fact_nb_client : fact nombre de clients	36
3.11	fact_conquête : fact conquête	36
3.12	fact_date_souscription : fact date de souscription	37
3.13	fact_segment : fact segment de client	37
3.14	fact_authentification : fact authentication	37
3.15	fact_virement : fact virement	38
3.16	fact_chéquier : fact chequier	38
3.17	fact_objectifs : fact objectifs	39
3.18	fact_types_acces : fact types d'accés	39

Liste des abréviations

la liste des abréviations est :

BI : business intelligence

FACT : table de fait

DIM : table dimension

SSH : Secure Shell

HTTP : Hypertext Transfer Protocol

FTP : File Transfer Protocol

DWH : DATA WAREHOUSE

ETL : extract transform load

DAG : Directed Acyclic Graph

TRO : taux de réalisations par rapport aux objectifs

Introduction générale

Les entreprises sont de plus en plus connectées à des réseaux mondiaux et ont donc besoin d'une grande quantité d'informations pour prendre des décisions éclairées. Les flux d'informations dans les entreprises ont augmenté de manière exponentielle au cours des dernières décennies. Selon les statistiques, les entreprises ont doublé leur capital informationnel entre 1990 et 2005, et aujourd'hui, elles peuvent le doubler tous les 72 jours. C'est pourquoi les entreprises doivent être en mesure de gérer efficacement ces flux d'informations pour rester compétitives sur le marché. [GR12]

La Business Intelligence (BI), ou l'informatique décisionnelle , est un domaine de l'informatique qui vise à collecter, analyser et présenter les données d'une entreprise de manière à fournir des informations pertinentes pour la prise de décision. Dans le contexte bancaire, la BI est utilisée pour aider les banques à prendre des décisions éclairées en fournissant des informations sur les activités, les clients, les produits et les tendances du marché.

L'industrie bancaire est en constante évolution, avec l'émergence de nouvelles technologies qui permettent de faciliter les opérations financières des clients. Aujourd'hui, l'application mobile est devenue un outil incontournable pour les clients d'une banque. Elle leur permet d'effectuer des opérations bancaires à tout moment et en tout lieu, en utilisant leur smartphone ou leur tablette.

Les systèmes bancaires génèrent des quantités massives de données, allant des transactions financières aux informations sur les clients et aux tendances économiques. L'analyse de ces données peut aider les banques à mieux comprendre les besoins et les comportements de leurs clients, à identifier les risques potentiels, à améliorer l'efficacité opérationnelle et à prendre des décisions éclairées en matière de marketing et de développement de produits.

Dans ce contexte, nous avons entrepris un projet de business intelligence qui de suivre l'application mobile « MY BIAT ». le but de ce projet est d'automatiser et analyser les données générées par l'application mobile pour identifier les tendances , les client souscrit dans différent régions et les comportements des clients .

Ce projet a pour ambition de fournir à notre banque "BIAT" des outils d'analyse de données en temps réel pour mieux comprendre les comportements de ses clients et aider l'équipe marketing digitale et l'équipe informatique à prendre des décisions éclairées pour améliorer l'expérience utilisateur et optimiser les performances de l'application mobile.

Notre rapport de projet se compose de cinq chapitres. Le premier chapitre couvre le cadre général du projet, le deuxième chapitre concerne l'étude Préalabe et spécification Des Besoins , le troisième chapitre contient la conception et la mise en place du data warehouse, le quatrième chapitre porte sur le processus ETL, et le cinquième chapitre traite du déploiement des tableaux de bord.

Chapitre 1

Cadre général du projet

Sommaire

Introduction	4
1.1 Cadre du projet	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Organigramme	5
1.2.2 Présentation de l'équipe de travail	5
1.3 Présentation du projet	5
1.4 Etude de l'existant	6
1.4.1 Problématique	6
1.4.2 Solution proposée	7
1.5 Méthodologie de gestion de projet	7
1.5.1 Méthode Bill Inmon	7
1.5.2 Méthode ralph kimball	8
1.5.3 bottom_up vs Top_Down	9
1.5.4 Choix de l'approche : Ralph Kimball	9
1.6 Les phases d'un projet BI selon l'approche de Ralph Kimball	10
1.7 Environnement de développement	13
1.7.1 Environnement logiciel	13
1.7.2 les langages de programmation	16
1.7.3 Environnement matériel Requis	16
1.7.4 Quels protocoles de communication allons-nous utiliser ?	18
1.7.5 L'architecture proposée de la solution	18
Conclusion	20

Introduction

Avant de procéder à la réalisation de notre projet, il est recommandé de présenter tout d'abord son cadre général. Ce chapitre a pour objectif de présenter l'organisme d'accueil, le cadre du sujet, le contexte du projet ainsi que le choix de la méthodologie de travail et Environnement de développement.

1.1 Cadre du projet

Notre projet s'inscrit dans le cadre de l'obtention du diplôme de licence en Business Intelligence (BI) au sein de la Faculté des sciences Economiques et Gestion de Tunis (FSEGT). Pour cela, nous avons effectué un stage de fin d'études au sein de Banque Internationale Arabe de Tunisie (BIAT). Ce qui vise à compléter notre formation et à nous introduire dans la vie professionnelle grâce à une mise en pratique de nos connaissances acquises durant les trois années d'études.

1.2 Présentation de l'organisme d'accueil

La Banque Internationale Arabe de Tunisie (BIAT) est une banque de secteur privé et comme toute Banque commerciale, son activité générale est de recevoir du public à titre de dépôts ou en vertu d'opérations assimilées, des fonds qui sont ensuite employés à titre de prêts.

Crée au courant des premiers mois de l'année 1976, la Banque Internationale Arabe de Tunis est résultat de l'association entre des initiatives et des capitaux tunisiens principalement du secteur privé et des efforts d'institutions financières arabes et internationales. La BIAT est qualifiée de la plus importante banque privée tunisienne selon les critères de part de marché, de qualité de service et de rentabilité.

En appuyant son développement sur la proximité, l'engagement sociétal, la BIAT a pu mettre son expertise et sa performance au profit de ses clients et de l'économie tunisienne. Cotée à la Bourse de Tunis, la BIAT est une entreprise à capitaux tunisiens. Actionnaire de référence, le groupe MABROUK est entré au capital de la Banque en 2005 et en détient 40 pourcent depuis 2010.

Au 01/01/2023, la banque compte 207 agences sur tout le territoire national contre 121 en 2008. Elle gérée actuellement environ 940 mille client, (particuliers, professionnelles, TPE , PME , ainsi que les grandes).

1.2.1 Organigramme

Le pôle banque de détail est la structure de la banque chargée de la gestion des marchés PP et PME à travers le réseau d'agence

Elle est organisée suivant deux types de structures :

- Structure régionales : 4 directions de régions 14 zones (207 Agences)
- Structure centrales du support : 5 directions

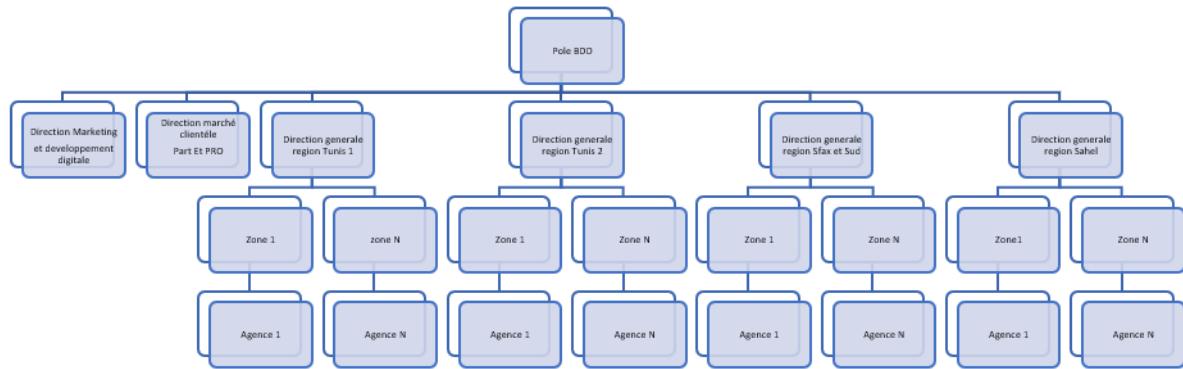


FIGURE 1.1 – Organigramme BIAT

1.2.2 Présentation de l'équipe de travail

Mon stage a été effectué sous la Direction maketing et développement digital dont les principales attributions sont :

- Surveiller l'application MyBIAT.
- Élaboration et mise en place de stratégies marketing.
- Veille et analyse des stratégies concurrentielles.
- Segmentation de la clientèle .
- Superviser les outils de gestion des ventes et la performance des ventes.
- Gérer et mettre à jour la base de données clients.
- Surveiller les représentants du service à la clientèle, les directeurs de points de vente et directeur de zones.

1.3 Présentation du projet

L'industrie bancaire est en constante évolution, avec l'émergence de nouvelles technologies qui permettent de faciliter les opérations financières des clients. Aujourd'hui, l'application mobile est devenue un outil incontournable pour les clients d'une banque. Elle leur permet d'effectuer des opérations bancaires à tout moment et en tout lieu, en utilisant leurs smartphones ou leur ordinateur.

Dans ce contexte, nous avons entrepris un projet de business intelligence qui permet de suivre les clients de l'application mobile « MY BIAT ». Le but de ce projet est d'automatiser par suite d'analyser les données générées par l'application mobile pour identifier les tendances, les clients souscrit dans différent régions et les comportements des clients pour aider l'équipe marketing digitale à prendre les bonnes décisions.

1.4 Etude de l'existant

La direction de Marketing et Développement Digital a constaté la nécessité d'avoir un suivi complet et fiable de l'activité de l'application mobile « My BIAT » et ces réalisations.

La direction se charge de collecter les données de l'application « My BIAT » afin d'effectuer une analyse complète par ces opérations et de proposer un reporting aux responsables marketing digitales.

Certes, les outils de pratiques de gestion actuellement utilisés par la banque ont su nous satisfaire en matière de mesure de la performance de l'activité MY BIAT et de collecte des informations nécessaires.

Cet outil présente plusieurs insuffisances à savoir :

- Absence de rapports complets avec des tableaux de bord clairs.
- Excel ne fournit pas de rapports et de chiffres de formation « instantanés».
- La préparation des reporting se fait d'une façon semi- automatique, nécessitant ainsi une charge de travail supplémentaire pour pouvoir acquérir tous les données, repérer les pertes et les retards, et de faire une analyse détaillée de l'activité de l'application mobile.

1.4.1 Problématique

De nos jours, une application mobile bancaire est devenue un outil incontournable pour les clients. En effet, elle leur permet d'effectuer plusieurs opérations à distance telles que les virements, les demandes de chéquiers et divers autres opérations. De ce fait, l'équipe Marketing et développement digital génère des quantités massives de données afin de suivre toutes les opérations effectuées ainsi que la performance de l'application "MyBiat" chaque période.

ces données peuvent prendre différentes formes mais elles sont généralement sous forme des fichiers Excel mal formés. Donc, afin de pouvoir les traiter, analyser et suivre, il est nécessaire d'avoir un rapport bien déterminer chaque fin de période.

1.4.2 Solution proposée

Après une réunion avec les responsables de l'équipe marketing et développement digital pour bien comprendre leurs besoins spécifiques, nous avons décidé de créer un entrepôt de données pour traiter les données nécessaires au suivi de l'application "MY BIAT". Nous utiliserons l'outil Apache Airflow pour automatiser le remplissage de données et simplifier les tâches liées à la gestion des données. De plus, nous automatiserons la création de rapports avec l'outil Power BI, qui offre plusieurs fonctionnalités pour visualiser, partager et analyser les données de manière efficace. Nous pourrons ainsi créer des tableaux de bord interactifs et faciliter la prise de décision grâce à une visualisation claire et compréhensible des données.

1.5 Méthodologie de gestion de projet

Adopter une méthodologie de gestion de projet est essentiel pour assurer la réussite du projet. Ces méthodologies fournissent une structure organisée pour gérer le projet à travers toutes les étapes, depuis la planification jusqu'à la mise en place, en utilisant des processus clairs et cohérents. Elles permettent aux équipes de travail de coordonner efficacement les activités du projet, d'affecter les ressources de manière optimale et de suivre la progression du projet tout en gérant les risques et les imprévus.

* L'approche Top-Down de Bill Inmon

* L'approche Bottom-Up de Ralph Kimball

1.5.1 Méthode Bill Inmon

Approche TOP-Down .

L'approche Top-Down de Bill Ammon peut également être appliquée dans un projet Business Intelligence (BI). Dans ce contexte, cette approche est utilisée pour concevoir et développer un système de BI qui répond aux besoins de l'entreprise et de l'utilisateur final.

« On ne fait rien tant que tout n'est pas désigné, le Data Warehouse doit être exhaustif!»**Bill Inmon.**

L'approche TOP-Down de Bill Inmon est une méthode de conception de l'entrepôt de données qui commence par la modélisation de l'entreprise et de ses processus, avant de définir les données nécessaires pour soutenir ces processus. Cette approche met l'accent sur l'importance de l'intégration des données à travers l'ensemble de l'entreprise, pour créer un entrepôt de données cohérent et précis.[Inm05]

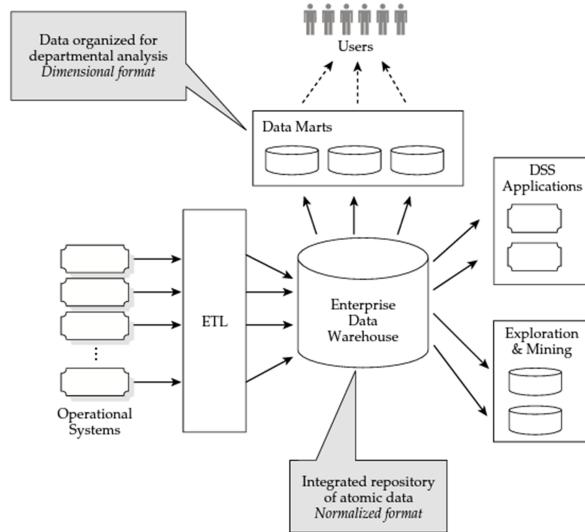


FIGURE 1.2 – Approche Top-Down

1.5.2 Méthode ralph kimball

Approche Bottom-Up.

Dans un projet BI (Business Intelligence), l'approche Bottom-Up est souvent utilisée pour développer des applications BI spécifiques, telles que des rapports ou des tableaux de bord. Cette approche peut être utile pour des projets qui impliquent de petits ensembles de données ou des sources de données peu complexes.

« Que chacun construise ce qu'il veut, on intégrera ce qu'il faudra quand il faudra ! » **Ralph Kimball**

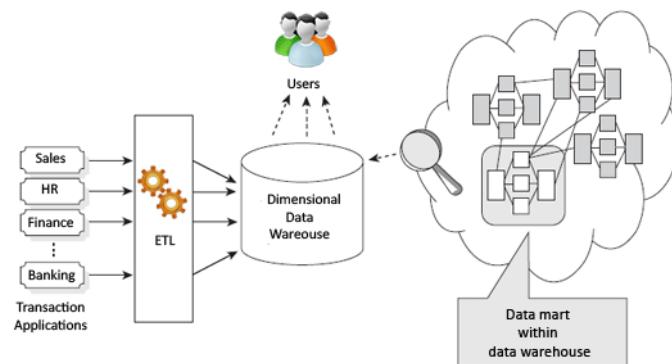


FIGURE 1.3 – Approche Bottom-Up

L'approche Bottom-Up de conception d'un entrepôt de données est une méthode qui commence par l'identification des sources de données, puis la définition de modèles de

données détaillés pour chaque source. Ces modèles sont ensuite combinés pour créer un modèle global de l'entrepôt de données. Cette approche est souvent utilisée pour construire des entrepôts de données de petite ou moyenne taille, avec des objectifs spécifiques. [Kim+13]

1.5.3 bottom_up vs Top_Down

TABLE 1.1 – étude comparative de ces deux approches actuelles

		Bottom-Up	Top-Down
1	Définitions	Une approche où l'on commence par observer des éléments individuels pour ensuite construire une compréhension globale.	Une approche où l'on commence par une compréhension globale pour ensuite la décomposer en éléments individuels.
2	Processus	Les détails sont rassemblés pour former un ensemble plus large.	Les détails sont analysés à partir d'un ensemble plus large.
3	Avantages	Permet de découvrir des tendances émergentes et des relations imprévues.	Offre une vue d'ensemble dès le départ, ce qui peut aider à éviter les erreurs de conception coûteuses.
5	Inconvénient	Peut être difficile à gérer avec de grandes quantités de données.	Peut manquer de détails importants qui ne sont pas inclus dans la conception globale initiale.
6	Objectifs	Livrer une solution technologiquement saine basée sur des méthodes et technologies éprouvées des bases de données.	Livrer une solution permettant aux usagers d'obtenir facilement et rapidement des réponses à leurs requêtes d'analyse.
7	Accessibilité des utilisateurs finaux	Faible	Forte
8	Outils de conception	Traditionnels (diagrammes entité-relation et flot de données)	Modélisation dimensionnelle

1.5.4 Choix de l'approche : Ralph Kimball

En fonction des besoins et contraintes fonctionnelles et opérationnelles dans l'étude comparative entre les deux approches, nous avons choisi d'utiliser la modélisation multidimensionnelle introduite par Ralph Kimball, qui fournit un processus complet répété pour chaque nouveau magasin de données demandé par l'utilisateur.

1.6 Les phases d'un projet BI selon l'approche de Ralph Kimball

Les phases du projet utilisant la méthode Ralph Kimball sont représentées comme suit :

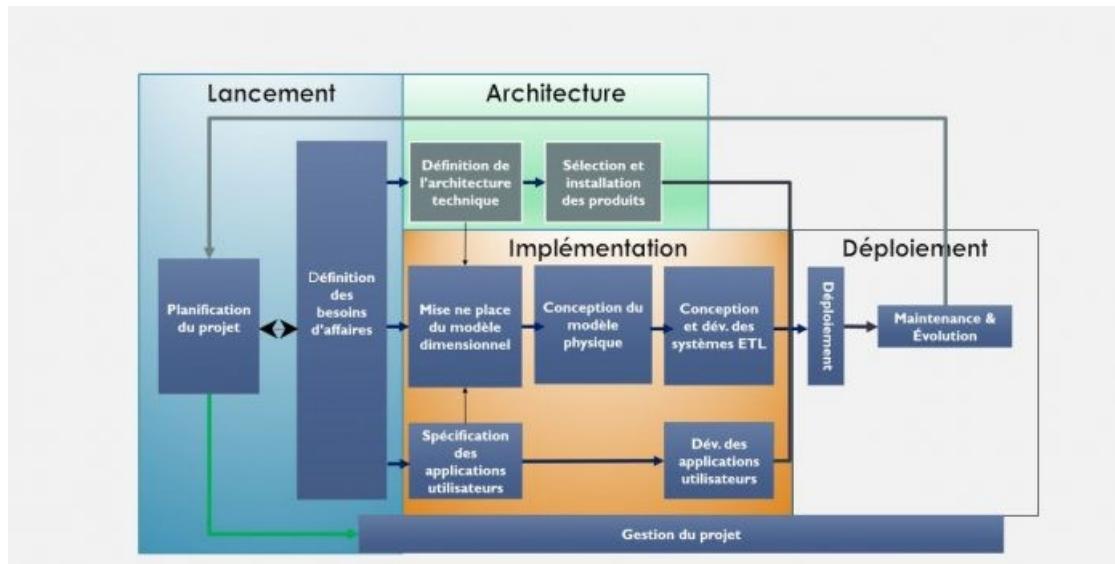


FIGURE 1.4 – Cycle de vie de la méthodologie Kimball

A) Planification

La phase de planification dans l'approche de Ralph Kimball pour la Business Intelligence (BI) est l'une des phases clés du processus de développement du projet. Cette phase vise à établir les exigences du projet, les objectifs, les livrables et les ressources nécessaires pour mettre en œuvre une solution de BI réussie.

B) Définition des besoins

la phase de définition des besoins d'un projet BI selon l'approche de Ralph Kimball est une étape importante pour comprendre les besoins de l'entreprise, les exigences métier et les besoins des utilisateurs finaux. Cette phase est essentielle pour concevoir une solution de BI efficace qui répondra aux besoins de l'entreprise et fournira des informations précieuses pour la prise de décision.

C) Modélisation dimensionnelle

La modélisation dimensionnelle est une étape critique dans la construction d'un entrepôt de données ou d'une solution de BI. Il aide à concevoir des modèles de données

flexibles et efficaces pour un stockage et une analyse optimaux des informations de l'entreprise.

Le modèle de données est conçu avec deux types d'entités : les dimensions et les faits. Les dimensions sont des catégories d'informations utilisées pour décrire les données, tandis que les faits sont des mesures quantitatives de l'activité.

D) Conception du modèle physique

Cette phase consiste à concevoir le schéma physique de l'entrepôt de données pour stocker les données selon le modèle dimensionnel créé lors de la phase de modélisation dimensionnelle.

La conception physique de l'entrepôt de données définit les structures requises pour mettre en œuvre le modèle dimensionnel. La création d'un environnement de base de données, l'indexation de base, les stratégies de partitionnement et les agrégations de base sont également définies.

E) Définition de l'architecture technique

La phase d'architecture technique du projet BI, selon l'approche de Ralph Kimball, consiste à définir l'ensemble de l'infrastructure technique nécessaire au support du système d'information décisionnel. Cette phase est essentielle pour garantir l'évolutivité, les performances et la sécurité du système.

Les environnements d'entrepôt de données nécessitent l'intégration de plusieurs technologies. cela se concentre sur les besoins, pas sur les détails techniques. Elle nécessite la prise en compte de trois facteurs : les besoins ; environnement existant et orientations techniques stratégiques prévues.

F) Choix technologiques et mise en œuvre

Cette phase comprend la sélection des techniques et des outils nécessaires à la création d'un projet BI , ainsi que leur mise en œuvre effective.

Les principales activités de la phase de sélection et de mise en œuvre de la technique sont : évaluation des options technologique, élaboration d'un plan de mise en œuvre , configuration et installation du logiciel, développement de solutions BI et test et validation.

G) La conception et le développement du système ETL

La Phase de conception et de développement du système ETL (Extract, Transform, Load) . L'objectif principal de cette phase est de concevoir et de développer un système ETL efficace pour extraire, transformer et charger des données de différentes sources dans l'entrepôt de données, cette phase représente 70 % des efforts et risque de projet.

H) Conception et développements des applications de BI

La phase de conception et de développement de l'application BI se déroule en parallèle avec la modélisation des données et la conception architecturale. vise à créer des applications qui permettent aux utilisateurs de visualiser des données, de les analyser et de prendre des décisions basées sur des données stockées dans un entrepôt de données.

I) Déploiement

Cette phase Présente la convergence de la technologie et des données ne doit pas être faite avant d'avoir les mécanismes de gestion et de suivi d'erreurs et la validation des données et outils.

J) Maintenance et croissance

La phase de maintenance et de croissance selon l'approche de Ralph Kimball vise à assurer la fiabilité, la sécurité et l'évolutivité du système d'information décisionnel après sa mise en service. Cette phase est essentielle pour s'assurer que le système de BI continue de répondre aux besoins de l'entreprise et reste efficace dans un environnement en constante évolution.

1.7 Environnement de développement

1.7.1 Environnement logiciel

PostgreSQL est un système de gestion de base de données relationnelle open source, développé par l’Université de Californie à Berkeley. Il offre une prise en charge complète du langage SQL et prend en charge de nombreuses fonctionnalités avancées, telles que les clés étrangères, les déclencheurs, les procédures stockées, les vues matérialisées, etc.[Che+05]



FIGURE 1.5 – logo PostgreSQL

Dbeaver est un logiciel libre de gestion de bases de données relationnelles. Il permet de travailler avec différents types de bases de données tels que MySQL, PostgreSQL, Oracle, Microsoft SQL Server et bien d’autres.[RGT19]



FIGURE 1.6 – logo Dbeaver

Apache Airflow est un système de gestion de flux de travail (workflow) open source conçu pour programmer, planifier et surveiller des tâches complexes. Il est particulièrement utile pour les processus ETL (Extract, Transform, Load) en traitement de données, mais peut être utilisé pour tout type de flux de travail automatisé. Airflow est écrit en Python et offre une grande flexibilité grâce à sa capacité à intégrer différents services et technologies.

Airflow permet de créer des workflows en utilisant un langage de programmation Python, dans lequel chaque étape du processus est représentée par une tâche. Les tâches peuvent être organisées en DAGs (Directed Acyclic Graphs), qui sont des graphes orientés acycliques décrivant l’ordre d’exécution des tâches.

Airflow offre également des fonctionnalités de planification avancées, comme la planification en fonction du temps, des déclencheurs externes ou des événements, ainsi qu'une interface utilisateur conviviale pour la surveillance des workflows en cours d'exécution et la gestion des erreurs.[BCM18]



FIGURE 1.7 – logo Apache Airflow

Docker est un logiciel open source qui permet de créer, déployer et exécuter des applications dans des conteneurs logiciels. Les conteneurs Docker fournissent un environnement isolé pour les applications, avec toutes les dépendances et les bibliothèques requises incluses dans le conteneur. Cela permet aux développeurs de créer des applications portables et de les déployer facilement sur différents environnements, tels que des serveurs locaux, des machines virtuelles ou des clusters de cloud computing. Docker est conçu pour être léger, rapide et facile à utiliser, ce qui en fait un choix populaire pour la construction et le déploiement d'applications modernes.[Mer14]

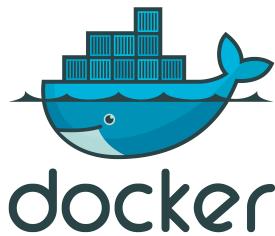


FIGURE 1.8 – logo : Docker

Visual Studio Code est un éditeur de code source gratuit et open source développé par Microsoft. Il est disponible sur plusieurs plateformes et offre une variété de fonctionnalités pour aider les développeurs à écrire du code plus rapidement et plus efficacement, y compris la coloration syntaxique, la suggestion de code, le débogage, la gestion de version, l'intégration de la ligne de commande, la collaboration en temps réel et plus encore. Il est également extensible avec un écosystème de plugins qui permet aux utilisateurs de personnaliser leur environnement de développement en fonction de leurs besoins spécifiques. Avec sa flexibilité, sa facilité d'utilisation et sa riche fonctionnalité, Visual Studio Code est devenu un outil très populaire pour les développeurs de tous niveaux et de nombreux langages de programmation.[ST20]

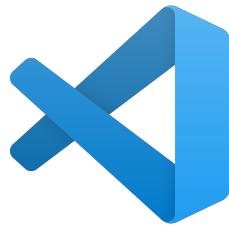


FIGURE 1.9 – logo VS Code

Power AMC est un logiciel de modélisation de données utilisé pour concevoir et modéliser des bases de données relationnelles. Il est développé par la société Sybase, une filiale de SAP, et permet de générer du code SQL pour la création de ces bases de données.[Syb03]



FIGURE 1.10 – logo Power AMC

Power BI Desktop est un outil de Business Intelligence (BI) développé par Microsoft qui permet de collecter, transformer et visualiser des données provenant de différentes sources pour aider les entreprises à prendre des décisions basées sur des données concrètes. Il est particulièrement apprécié pour sa simplicité d'utilisation et son interface graphique intuitive qui permet de créer des tableaux de bord interactifs et des rapports visuels personnalisés.[Sel18]

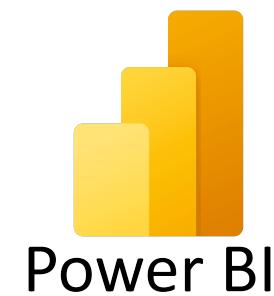


FIGURE 1.11 – logo Power BI

overleaf est un éditeur de texte en ligne qui permet de créer, éditer et collaborer sur des documents LaTeX en temps réel. Il est utilisé principalement pour la rédaction de documents scientifiques, tels que des articles de recherche, des thèses et des rapports techniques. Les utilisateurs peuvent accéder à un large éventail de modèles et de packages

LaTeX prédefinis, ce qui facilite la mise en forme et la création de documents professionnels[HL14]



FIGURE 1.12 – logo Overleaf

1.7.2 les langages de programmation

SQL(Structured Query Language) est un langage de programmation pour gérer, créer et modifier des bases de données relationnelles.

Python est un langage de programmation orienté objet . il est conçu pour être facile à lire, écrire et maintenir, avec une syntaxe simple qui permet aux développeurs de se concentrer sur la résolution de problèmes plutôt que sur les détails de la syntaxe.

LaTeX est un langage de programmation de mise en page, qui permet de créer des documents de qualité professionnelle avec une mise en forme cohérente et précise.

1.7.3 Environnement matériel Requis

TABLE 1.2 – Configuration requise

Composant	Configuration minimale
Système d'exploitation	Windows 10 Professionnel
Processeur	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
RAM	8,00 Go (7,79 Go utilisable)
Disque dur	475 Go
Types du Système	Système d'exploitation 64 bits, processeur x64

Travailler avec des outils comme Airflow et Docker peut parfois nécessiter des ressources système importantes, en particulier en ce qui concerne la RAM. Pour ce projet, nous avons décidé d'utiliser une machine virtuelle Azure pour héberger ces outils. Cette solution nous a permis d'allouer des ressources supplémentaires à notre environnement de développement sans affecter les performances de nos PC.

Avec une machine virtuelle comme . la possibilité d'allouer plus de RAM ou un processeur plus puissant pour faire face à des tâches complexes et simultanées. De plus, comme chaque environnement est isolé des autres, nous pouvons réduire le risque de conflits entre différents outils ou applications.

En utilisant une machine virtuelle azure pour héberger des outils comme Airflow, Docker et PostgreSQL, nous avons pu optimiser nos ressources système tout en améliorant les performances et la stabilité de notre environnement de développement.

Définition de "Microsoft Azure" : est une plateforme cloud de Microsoft qui permet de déployer, gérer et développer des applications dans le cloud. Elle offre une large gamme de services, tels que des machines virtuelles, des services de stockage, des bases de données, des services d'analyse de données et bien plus encore.[Cha12]



FIGURE 1.13 – logo : Microsoft Azure

Définition de "Machine virtuelle" : est un environnement informatique qui reproduit les fonctionnalités d'une machine physique, telles que le système d'exploitation, les applications et les périphériques. Les machines virtuelles sont largement utilisées dans les environnements de développement et de test, car elles permettent aux développeurs de travailler dans un environnement isolé et contrôlé. Elles sont également utilisées dans les centres de données pour optimiser l'utilisation des ressources et fournir un environnement de déploiement plus flexible.[PK13]

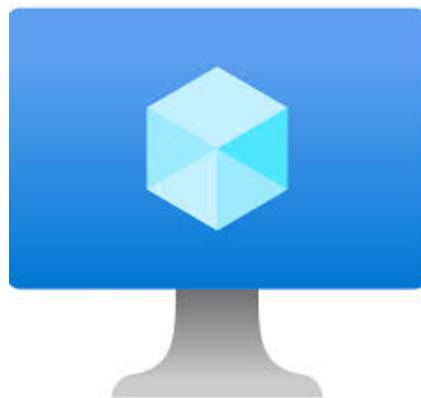


FIGURE 1.14 – machine virtuelle

CHAPITRE 1. CADRE GÉNÉRAL DU PROJET

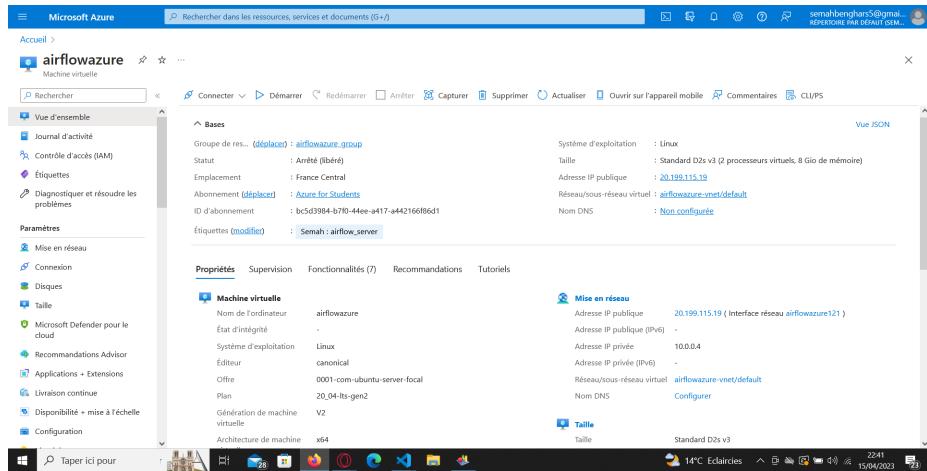


FIGURE 1.15 – notre machine virtuelle

1.7.4 Quels protocoles de communication allons-nous utiliser ?

Pour réussir notre projet, j'ai loué une machine virtuelle pour héberger mes outils. Je prévois d'installer Docker sur cette machine virtuelle et de déployer Airflow et PostgreSQL dans Docker. Pour permettre l'accès à notre machine. nous envisageons d'utiliser différents protocoles de communication tels que :

SSH : pour ssh on va utiliser le port 22 pour établir shell distant à notre machine virtuelle.

HTTP port 8080 : est un port standard utilisé pour le trafic Web non chiffré. Il est utilisé par les serveurs Web pour écouter les demandes de clients HTTP.[Far+19]

le port HTTP 8080 sera utilisé pour afficher l'interface web d'Airflow.

FTP : (File Transfer Protocol) est un protocole de transfert de fichiers largement utilisé pour l'échange de fichiers sur des réseaux informatiques.

remarque : J'ai utilisé mon email institutionnel pour louer cette machine virtuelle.

1.7.5 L'architecture proposée de la solution

La Business Intelligence (BI) regroupe une famille d'outils logiciels spécifiques qui ont pour but de traiter les données et faciliter la prise de décisions pour les décideurs. Les outils de BI sont utilisés pour toutes les phases du processus décisionnel, de la collecte de données à la présentation des résultats pour l'aide à la prise de décision.

Pour mieux comprendre les différents outils de la BI, il est possible de les classer en trois catégories qui correspondent chacune à une fonction spécifique, à une phase du processus décisionnel. La première catégorie correspond à la collecte, et le traitement des données, la deuxième catégorie est destinée au stockage des données, et la troisième catégorie concerne l'analyse et la diffusion des résultats aux décideurs.

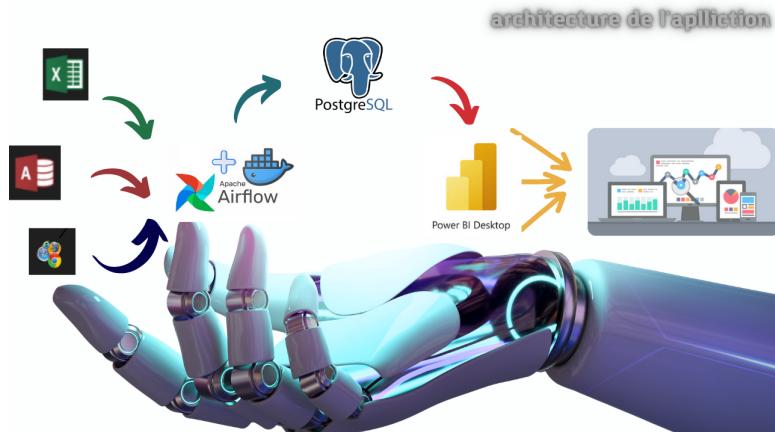


FIGURE 1.16 – Architecture de la solution

La chaîne d'information décisionnelle comprend différentes phases :

- a. La phase d'alimentation** La phase d'alimentation comprend la détection, la sélection, l'extraction, la transformation et le chargement dans un entrepôt (DWH) l'ensemble des données brutes provenant de diverses sources de stockage (bases de données, fichiers plats, applications métier, etc.).
- b. La phase de modélisation** Une fois les données collectées et centralisées, la phase de modélisation consiste à stocker et structurer les données dans un espace unifié appelé Data Warehouse. Cette étape est essentielle pour que les données soient disponibles pour un usage décisionnel.
- c. La phase de restitution** La phase de restitution, également appelée phase de visualisation, est l'étape finale du processus d'analyse de données en Business Intelligence (BI). Cette étape consiste à présenter les données de manière claire, compréhensible et interactive aux utilisateurs finaux afin qu'ils puissent prendre des décisions éclairées.
- d. La phase d'analyse** L'objectif de la phase d'analyse en Business Intelligence est d'aider les utilisateurs à explorer les informations mises à leur disposition de manière efficace et de les guider dans leur prise de décision

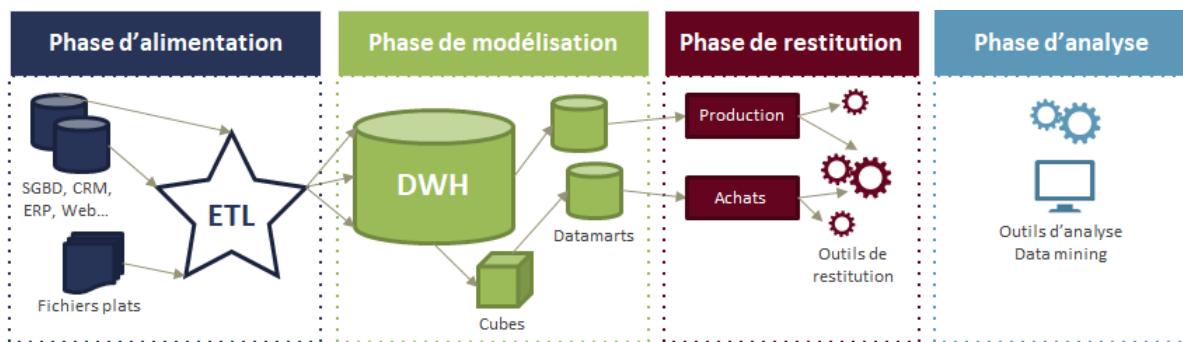


FIGURE 1.17 – La chaîne d'information décisionnelle

Conclusion

Dans ce chapitre, nous avons mis le projet dans son cadre général. Après avoir présenté l'organisme d'accueil, la problématique et la solution proposé et avoir reconnu ses attentes du projet, nous avons défini la méthodologie adopté pour le pilotage du projet et nous avons décrit également l'environnement et l'architecture du développement consacrons le chapitre suivant pour l'étude préalable et la spécification des besoins

Chapitre 2

Étude Préalable Et Spécification Des Besoins

Sommaire

Introduction	22
2.1 planification de projet	22
2.2 Concepts de base de la Buisness Intelligence	22
2.2.1 Les intérêts du Système décisionnel (BI) pour le secteur bancaire	22
2.2.2 L'informatique décisionnelle (BI)	23
2.2.3 Le processus d'intégration de données dans un entrepôt de données (ETL)	23
2.2.4 Entrepôt de données (DWH)	24
2.2.5 La différences entre un Data Warehouse et un Data Mart	25
2.2.6 Notions des tables de faits et de dimensions	26
2.3 Spécification des besoins	26
2.3.1 Identification des acteurs	27
2.3.2 Identification des besoins fonctionnels	27
2.3.3 Identification des besoins non fonctionnels	27
2.4 Diagramme de cas d'utilisation global	28
Conclusion	28

Introduction

Dans ce chapitre, notre objectif est d'identifier et de spécifier les exigences pour assurer une mise en œuvre réussie. L'objectif principal de ce chapitre est de connaître le projet, comprendre les principales caractéristiques du projet, le planifier et l'organiser .

2.1 planification de projet

Nous utiliserons un diagramme de Gantt pour estimer le temps nécessaire pour chaque tâche de notre stage et présenter le planning d'implémentation de notre système de manière optimale. Le diagramme permettra également de visualiser les interactions entre les différentes tâches.



FIGURE 2.1 – Gantt Chart

2.2 Concepts de base de la Business Intelligence

La Business Intelligence (BI) est un ensemble de processus, de technologies et d'outils utilisés pour collecter, analyser et présenter des données pertinentes afin d'aider les entreprises à prendre des décisions éclairées.

2.2.1 Les intérêts du Système décisionnel (BI) pour le secteur bancaire

Les systèmes de Business Intelligence (BI) offrent de nombreux avantages au secteur bancaire. En fait, les banques traitent de grandes quantités de données liées à leurs clients, produits, opérations, etc. Les systèmes de BI permettent de collecter, d'analyser et de transformer ces données en informations utiles à la décision. Fortes de ces informations, les banques peuvent mieux comprendre les besoins des clients, prévoir les tendances du marché, gérer les risques et améliorer la rentabilité. Les systèmes de BI contribuent également à améliorer l'efficacité opérationnelle en automatisant certaines tâches et en

fournissant des tableaux de bord en temps réel pour surveiller les indicateurs de performance clés. En conclusion, les systèmes de BI offrent un avantage concurrentiel aux banques qui cherchent à accroître leur efficacité, leur agilité et leur rentabilité.

2.2.2 L'informatique décisionnelle (BI)

L'informatique décisionnelle (ou La Business Intelligence (BI) est un ensemble de méthodologies, de processus, de technologies et de compétences qui permettent aux entreprises de collecter, stocker et analyser des données provenant de diverses sources pour améliorer la prise de décision. Elle utilise des outils tels que les entrepôts de données, les outils d'extraction, de transformation et de chargement (ETL), les outils de modélisation de données, les outils de visualisation de données et les outils d'analyse de données pour aider les entreprises à identifier les tendances, les modèles et les anomalies dans les données et à prendre des décisions basées sur ces informations.

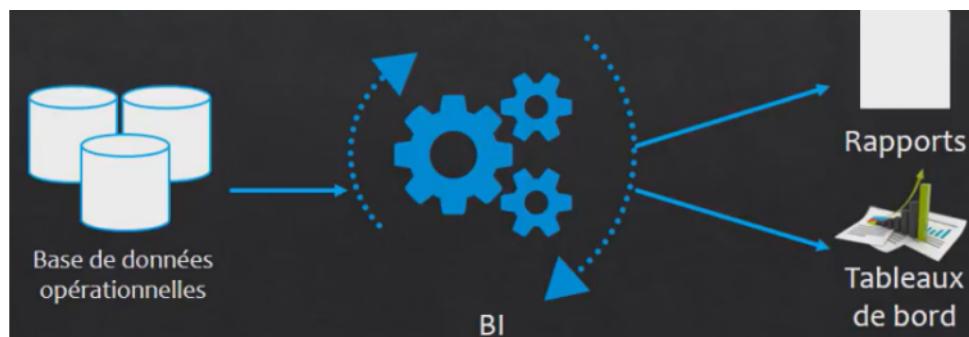


FIGURE 2.2 – Processus De Business intelligence

2.2.3 Le processus d'intégration de données dans un entrepôt de données (ETL)

le processus d'intégration de données dans un entrepôt de données est l'étape cruciale de la construction d'un entrepôt de données. Il s'agit de la consolidation de données hétérogènes provenant de sources diverses en un ensemble de données cohérent pour l'analyse ultérieure. Ce processus comprend la collecte de données, la transformation de données et le chargement de données dans l'entrepôt de données. Le processus d'intégration de données est également connu sous le nom d'ETL (Extract, Transform, Load).[KL07]

- **Extraction :** L'ETL se charge de récupérer toutes les données nécessaires à partir des différentes sources de stockage.
- **Transformation :** Il s'agit d'une étape critique dans un projet de prise de décision,

il est nécessaire de restaurer des données cohérentes et pertinentes pour l'entreprise grâce à la nettoyage des données (correction des erreurs, suppression des doublons, résolution des conflits).

- **Changement :** C'est la dernière étape du projet, qui consiste à charger les données dans un entrepôt de données ou une base de production afin qu'elles soient disponibles pour les différents outils d'analyse.

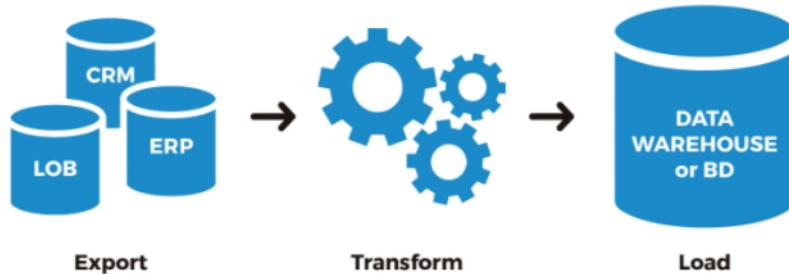


FIGURE 2.3 – Processus ETL

2.2.4 Entrepôt de données (DWH)

Selon Kimball, un entrepôt de données est un système informatique conçu pour stocker, gérer et analyser de grandes quantités de données provenant de différentes sources. Il est conçu pour fournir des données intégrées et cohérentes, qui peuvent être utilisées pour l'analyse, la planification et la gestion des opérations commerciales.[Kim+13]

2.2.4.1 Caractéristiques clés d'un entrepôt de données

Orientation sujet : Les données dans un entrepôt de données sont organisées autour de sujets métier spécifiques.

Intégration : Les données de différentes sources sont collectées, nettoyées, transformées et intégrées dans l'entrepôt de données de manière cohérente et uniforme.

Non-volatilité : Une fois que les données sont chargées dans l'entrepôt de données, elles ne sont généralement pas modifiées. Les données historiques sont conservées et de nouvelles données sont ajoutées régulièrement.

Temps variant : Les données sont organisées dans l'entrepôt de données de manière à pouvoir être analysées sur une période donnée. Les données historiques sont conservées et peuvent être utilisées pour effectuer des analyses comparatives.

Évolutivité : L'entrepôt de données doit être évolutif pour pouvoir traiter des volumes de données en constante augmentation et répondre aux besoins changeants des utilisateurs.

Flexibilité : L'entrepôt de données doit être flexible pour s'adapter aux changements dans les besoins métier et les sources de données. Les modifications apportées aux données et à la structure de l'entrepôt de données doivent être effectuées rapidement et efficacement.

2.2.5 La différences entre un Data Warehouse et un Data Mart

Le Data Warehouse est une solution de stockage centralisée et globale de données, généralement destinée à une utilisation à grande échelle pour l'analyse de données historiques. En revanche, le Data Mart est un sous-ensemble de données ciblé sur un domaine métier spécifique, destiné à une utilisation par des groupes d'utilisateurs limités et pour des analyses spécifiques. Les fonctionnalités, la complexité, les coûts et le temps de mise en place varient considérablement entre les deux solutions, en fonction des besoins et des objectifs de l'entreprise.

TABLE 2.1 – Comparaison entre un Data Warehouse et un Data Mart

Caractéristiques	Data Warehouse	Data Mart
Objectif	Stocker des données provenant de diverses sources pour une analyse globale de l'entreprise	Stocker des données pour une analyse spécifique d'un département ou d'une fonction
Portée	Global	Spécifique
Taille	Grand	Petit à moyen
Fréquence de mise à jour	Faible	Élevée
Niveau de détail	Faible à élevé	Élevé
Niveau d'agrégation	Élevé	Faible à élevé
Conception	Centralisée	Décentralisée
Coût	Élevé	Moyen

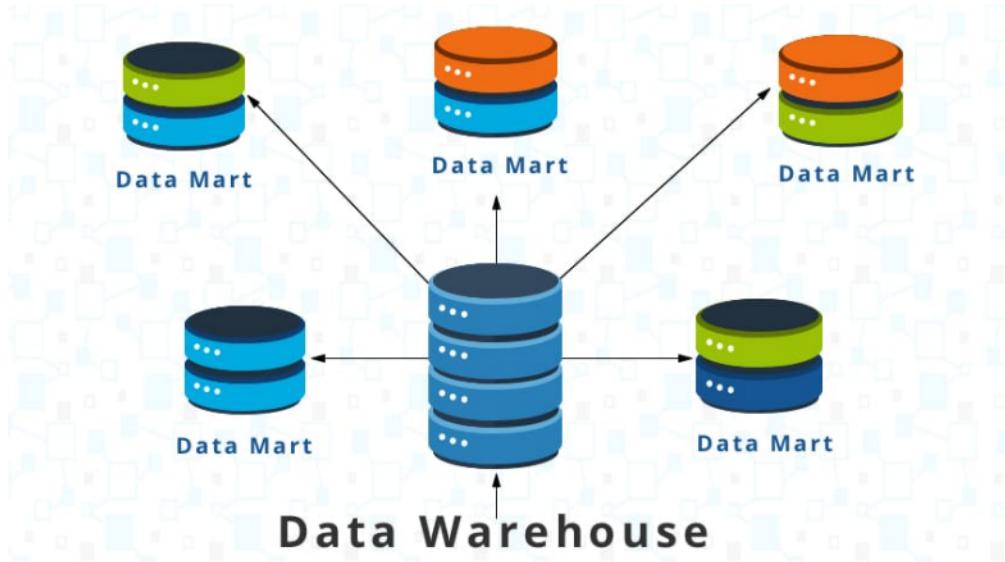


FIGURE 2.4 – Data Mart vs Data Ware House

2.2.6 Notions des tables de faits et de dimensions

Table de fait :

c'est une table dans laquelle sont stockées les mesures numériques ou quantitatives (ex : montant, quantités, etc.) liées à un événement ou à une transaction. Cette table contient généralement des clés étrangères faisant référence aux tables de dimensions pour plus de détails .

Table de dimension :

est une table qui contient des informations textuelles ou descriptives sur les différentes dimensions qui décrivent les événements ou les transactions dans la table de faits. Les tables de dimension sont généralement plus petites que les tables de faits et contiennent souvent des attributs hiérarchiques pour faciliter l'agrégation des données.

2.3 Spécification des besoins

Le présent projet vise à créer une solution pour suivre les clients d'une application mobile d'une banque.

Dans ce contexte, plusieurs besoins fonctionnels ont été identifiés pour aider la banque à mieux comprendre son marché et à prendre des décisions basées sur les données

2.3.1 Identification des acteurs

L'identification des acteurs est une étape importante dans la modélisation des processus et des systèmes, en particulier lors de la création de diagrammes de cas d'utilisation. Il s'agit de comprendre qui ou quoi est impliqué dans un processus ou un système particulier. Les acteurs peuvent être des personnes, des groupes de personnes.

Dans le cadre de notre projet, nous distinguons 1 acteur tel que l'équipe marketing et développement digital. les personnes qui peut visualiser et consulter tous les rapports réalisés pour suivre les opérations et les rendements de l'application MY BIAT.

2.3.2 Identification des besoins fonctionnels

D'après une réunion avec le responsable de l'équipe marketing et développement digital, on a identifié les besoins fonctionnels suivants :

- Création de rapports automatisés pour suivre l'application à travers les souscriptions par période , agence ,zone , région.
- Création de rapports automatisés pour suivre les clients par agence ,zone , région.
- Création de rapports automatisés pour suivre la conquête clients et les types d'accès.
- Création de rapports automatisés pour suivre les segments client par agence, zone.
- Création de rapports automatisés pour suivre les performances de l'application à travers les authentifications.
- Création de rapports automatisés pour suivre l'application à travers les virements effectués par l'application.
- Création de rapports automatisés pour suivre les performances de l'application à travers les demandes de chéquiers.
- Création de rapports automatisés pour suivre l'application à travers les objectifs par rapport aux réalisations.
- Visualisation des rapports sous forme de listes et de tableaux de bord

2.3.3 Identification des besoins non fonctionnels

on a identifié les besoins non fonctionnels suivants :

- **La performance** : Les tableaux de bord doivent répondre à toutes les exigences des décideurs d'une manière optimale.

- **Fiabilité** : garantir la qualité du contenu et l'actualité des informations .
- **Modularité** : La solution doit être modulaire et bien organisée.
- **Ergonomie** : La première chose à laquelle les décideurs prêtent attention est l'ergonomie et la convivialité des tableaux de bord, une attention particulière doit être portée à ce besoin.

2.4 Diagramme de cas d'utilisation global

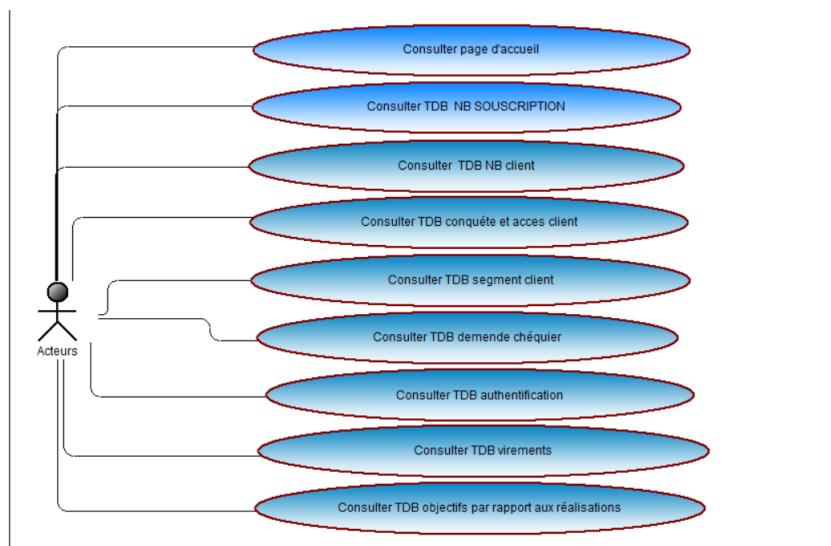


FIGURE 2.5 – Diagramme de cas d'utilisation

Conclusion

Dans ce chapitre, nous avons présenté les principes de fonctionnement du système de prise de décision ainsi que ses différentes parties. Nous avons ensuite identifié ce qu'est un data mart, les caractéristiques d'un entrepôt de données, ainsi que les besoins fonctionnels et non fonctionnels de notre projet

Chapitre 3

Conception et Mise En Place Du Data Warehouse

Sommaire

Introduction	30
3.1 La modélisation dimensionnelle du Data Warehouse	30
3.2 Identification des faits et des dimensions	33
3.2.1 Choix des dimensions	33
3.2.2 Choix des indicateurs	35
3.3 Modélisation du Data Warehouse	39
3.3.1 Modèles en Constellation	39
3.4 Mise En Place Du Data warehouse	41
3.4.1 Data Warehouse my_biat_dwh	41
3.4.2 Diagramme my_biat_dwh	42
Conclusion	42

Introduction

La modélisation et la création d'un data warehouse sont des étapes essentielles dans tout projet de Business Intelligence. Pour garantir la qualité et la fiabilité de vos analyses, il est important de concevoir une structure de données cohérente et optimisée pour l'analyse. Dans ce chapitre nous allons voir comment créer un modèle de données multidimensionnel et la création d'un data warehouse.

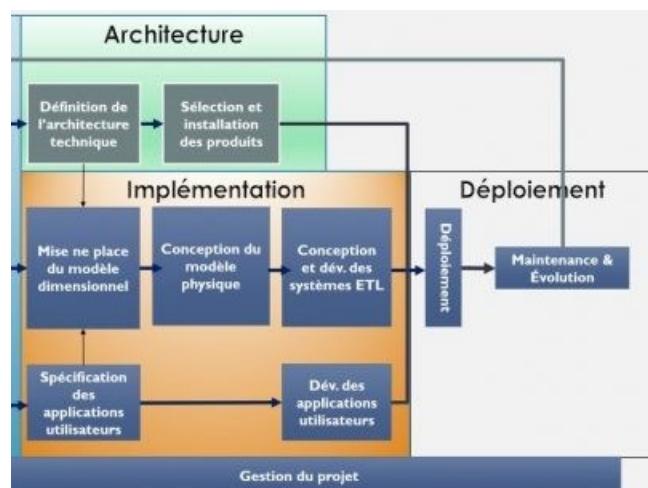


FIGURE 3.1 – Cycle de vie de la méthodologie Kimball

3.1 La modélisation dimensionnelle du Data Warehouse

La modélisation dimensionnelle est une technique de base de données pour l'entreposage de données qui vise à fournir un accès rapide et facile aux données pour l'analyse et la prise de décision. Elle se concentre sur l'organisation des données en fonction de leur pertinence pour les utilisateurs plutôt que de leur structure technique.

La modélisation d'un entrepôt de données comporte 3 modèles :

- Modèle en étoile : C'est un modèle simple et facile à comprendre qui contient une table centrale appelée table de faits qui contient des mesures numériques. Cette table est liée à plusieurs tables de dimension qui contiennent des informations sur les attributs qui décrivent les mesures (par exemple, le temps, les régions, les agences, les clients). Le modèle en étoile est souvent utilisé pour les données transactionnelles avec des mesures numériques.

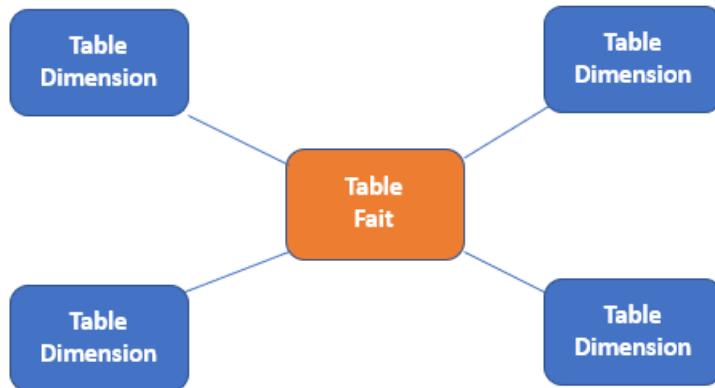


FIGURE 3.2 – Modèle en étoile

2. Modèle en flacons : il s'agit d'un modèle qui ressemble au modèle en étoile, mais qui est plus complexe en raison de la normalisation de certaines tables de dimension. Les tables de dimension normalisées sont divisées en plusieurs tables pour éviter la redondance de données. Ce modèle est souvent utilisé lorsque la taille de l'entrepôt de données est importante et que la normalisation est nécessaire pour éviter les problèmes de performances.

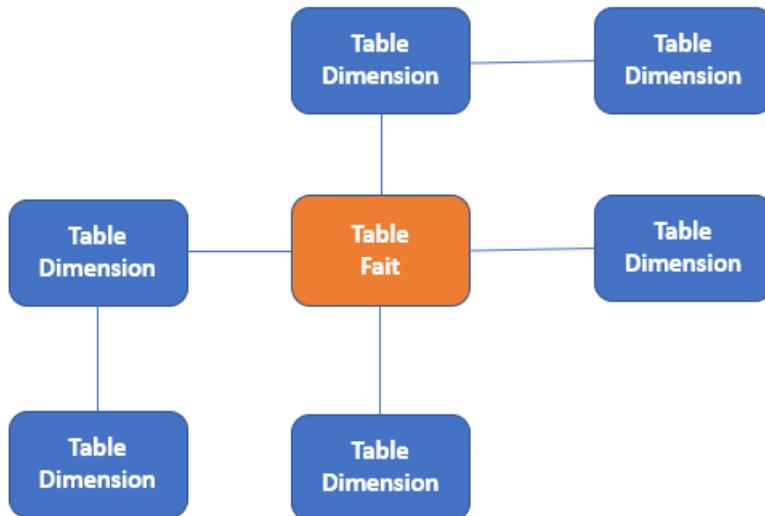


FIGURE 3.3 – Modèle en flacons

3.Modèle en constellation :il s'agit d'un modèle plus complexe que le modèle en flacons , contenant plusieurs tables de faits partageant des tables de dimensions communes. Ce modèle est souvent utilisé pour les organisations ayant des activités diverses et des besoins d'analyse plus complexes.

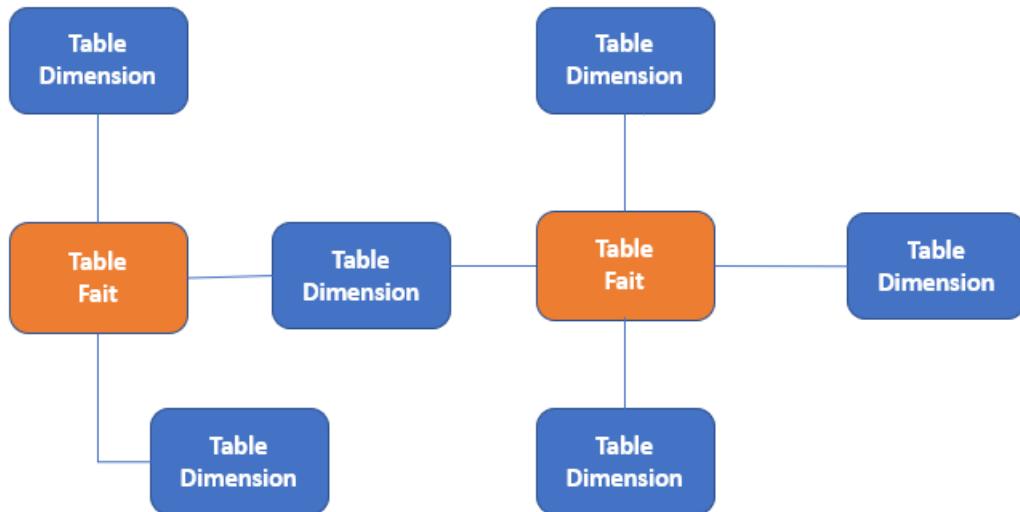


FIGURE 3.4 – Modèle en constellation

En somme, la modélisation dimensionnelle est une approche clé pour concevoir un entrepôt de données efficace et performant, qui facilite l'analyse de données pour la prise de décisions. Les modèles en étoiles, en constellation et en flocons sont les trois modèles de données dimensionnelles les plus courants, chacun offrant des avantages et des inconvénients en fonction des besoins.

3.2 Identification des faits et des dimensions

3.2.1 Choix des dimensions

Dans un entrepôt de données, les dimensions décrivent les données dans les tables de faits, et il est important de choisir les bonnes dimensions pour permettre une analyse pertinente des données. Dans notre cas, nous avons 8 dimensions comme indiqué ci-dessous :

DIM_AGENCE : dimension agence contient toutes les informations sur les agences, telles que leur géolocalisation et leur classement.

TABLE 3.1 – dim_agence : dimension agence

COIONNES	TYPES	LIBLE
ID_AGENCE_	VARCHAR	identifiant Agence(clé primaire)
ID_ZONE	INT	identifiant Zone(clé étrangère)
ID_REGION	INT	identifiant Région(clé étrangère)
LIBELLE_AGENCE	VARCHAR	Nom de L'agence
CLASSE_AGENCE	INT	classement de l'agence

DIM_ZONE : dimension Zone contient toutes les informations sur les zones telles que géolocalisation.

TABLE 3.2 – dim_zone : dimension zone

COIONNES	TYPES	LIBLE
ID_ZONE	INT	identifiant Zone (clé primaire)
ID_REGION	INT	identifiant région (clé étrangère)
LIBELLE_ZONE	VARCHAR	Nom de la ZONE

DIM_REGION : Dimension région contient toutes les informations sur les régions

TABLE 3.3 – dim_region : dimension region

COIONNES	TYPES	LIBLE
ID_REGION	INT	identifiant Région (clé primaire)
LIBELLE_REGION	VARCHAR	Nom de la région

DIM_CLIENT : La dimension client contient les informations spécifiques aux clients,

TABLE 3.4 – dim_client : dimension client

COIONNES	TYPES	LIBELLE
ID_CLIENT	INT	identifiant client (clé primaire)
NOM	VARCHAR	Nom de client
GSM	NUMERIC	numéro de téléphone client
ADRESSE_MAIL	VARCHAR	adresse mail de client

DIM_SOUSCRIPTION : la dimension souscription contient les informations spécifiques aux clients souscrits, tandis que les autres informations sont considérées comme des mesures.

TABLE 3.5 – dim_souscription : dimension souscription

COIONNES	TYPES	LIBELLE
ID_SOUSCRIT	VARCHAR	identifiant souscription(clé primaire)
ID_CLIENT	INT	identifiant client (clé étrangère)
NOM	VARCHAR	Nom De client souscrit
ADRESSE_MAIL	VARCHAR	Adresse mail de client souscrit
GSM	NUMERIC	numéro de téléphone client souscrit
PHONE_VERIFIED	BOOLEEN	verification avec le numero de telephone
UPDATE_PASSWORD	BOOLEEN	mettre à jour le mot de passe
DATE_SOUSCRIT	DATETIME	mettre à jour le mot de passe

DIM_DATE_AUTHENTIFICATION : La dimension de la date d'authentification contient toutes les informations horaires .

TABLE 3.6 – dim_date_authentification : dimension date authentification

COIONNES	TYPES	LIBELLE
ID_DATE_AUTH	VARCHAR	identifiant date authentification (clé primaire)
ID_CLIENT	INT	identifiant client(clé étrangère)
ANNEES	INT	l'années d'authentification
MOIS	INT	le mois d'authentification
JOURS	INT	le jour d'authentification
HEURES	DATETIME	l'heure d'authentification

DIM_DATE_CHEQUIER : la dimension date chéquier contient toute les informations horaires pour chaque demande de chéquier.

TABLE 3.7 – dim_date_chequier : dimension date chequier

COIONNES	TYPES	LIBELLE
ID_DATE_CHEQ	VARCHAR	identifiant date chequier(clé primaire)
ANNEES	INT	L'annees de la demande de chequier
MOIS	INT	le mois de la demande de chequier
JOURS	INT	le jour de la demande de chaequier
HEURES	DATETIME	l'heures de la demande de chéquier

DIM_DATE_VIREMENT : la dimension date virement contient les informations horaire pour chaque virement .

TABLE 3.8 – dim_virement : dimension date virement

COIONNES	TYPES	LIBELLE
ID_DATE_VIRMT	VARCHAR	identifiant date virement (clé primaire)
ANNEES	INT	L'années de virement
MOIS	INT	Le mois de virement
JOURS	INT	Le jour de virement
HEURES	DATE	L'heurs de virement

3.2.2 Choix des indicateurs

Le choix des indicateurs, également appelé choix des tables de fait, est une étape importante dans la conception d'un entrepôt de données. Cette étape consiste à identifier les données clés qui doivent être stockées dans l'entrepôt pour répondre aux besoins des utilisateurs finaux.

Les indicateurs de performance clés (KPI) sont "des mesures quantitatives ou qualitatives utilisées pour évaluer la performance d'une organisation ou d'un processus en fonction des objectifs prédéfinis" (Parmenter, 2015).[Par15]

Nous avons décidé de créer neuf tables de faits pour notre entrepôt de données afin de réaliser une analyse détaillée. Ces tables de faits nous permettront d'obtenir une vue complète et précise de nos données en agrégeant et en organisant les informations importantes en fonction des différentes dimensions. Chaque table de faits sera liée à une ou plusieurs tables de dimensions pour nous aider à comprendre les relations et les interdépendances entre les données. Grâce à ces neuf tables de faits, nous serons en mesure d'analyser efficacement nos données et d'obtenir des informations pertinentes pour prendre des décisions éclairées.

FACT_NB_SOUSCRIT : Notre objectif est d'examiner la croissance des abonnements dans différentes zones géographiques, telles que les régions, les zones et les agences.

TABLE 3.9 – fact_nb_souscrit : fact nombre de souscriptions

COIONNES	TYPES	LIBELLE
ID_NB_SOUSCRIT	VARCHAR	identifiant nembre de souscription (clé primaire)
DI_AGENCCE	VARCHAR	identifiant agence (clé étrangère)
ID_CLIENT	INT	identifiant client (clé étrangère)
NB_SOUSCRIT	NUMERIC	nembre des clients souscrit

FACT_NB_CLIENT : Notre objectif est d'analyser la croissance du nombre de clients dans différentes zones géographiques, telles que les régions, les zones et les agences.

TABLE 3.10 – fact_nb_client : fact nombre de clients

COIONNES	TYPES	LIBELLE
ID_NB_CLIENT	VARCHAR	identifiant nembre de client (clé primaire)
DI_AGENCCE	VARCHAR	identifiant agence (clé étrangère)
NB_CLIENT	NUMERIC	nembre des clients

FACT_CONQUETE :Notre objectif est de mesurer la performance de notre application mobile bancaire en fonction de la conquête client (nouveaux ou anciens clients) pour mieux comprendre les différences entre les groupes d'utilisateurs et prendre des décisions pour améliorer l'expérience utilisateur et la croissance de notre clientèle.

TABLE 3.11 – fact_conquête : fact conquête

COIONNES	TYPES	LIBELLE
ID_CONQUETE	VARCHAR	identifiant conquête et accées client (clé primaire)
ID_CLIENT	INT	identifiant nembre de client (clé étrangère)
ID_SOUSCRIT	VARCHAR	identifiant nembre de souscription (clé étrangère)
DI_AGENCCE	VARCHAR	identifiant agence (clé étrangère)
CONQUETE	BOOLEEN	conquête de client 0 pour les anciens et 1 pour les neauveaux

FACT_DATE_SOUSCRIPTION : Notre objectif est d'analyser la croissance des souscriptions au fil du temps .

TABLE 3.12 – fact_date_souscription : fact date de souscription

COIONNES	TYPES	LIBELLE
ID_DATE_SOUSCRIT	VARCHAR	identifiant Date de souscription (clé primaire)
ID_SOUSCRIT	VARCHAR	identifiant client souscrit (clé étrangère)
ANNES	INT	annes de souscriptions
MOIS	INT	mois de souscriptions
JOURS	INT	jours de souscriptions
HEURES	TIME	jours de souscriptions

FACT_SEGMMENT : Notre objectif est d'analyser la répartition des clients par segments ainsi que la répartition des segments par agence, zone et région.

TABLE 3.13 – fact_segment : fact segment de client

COIONNES	TYPES	LIBELLE
ID_SEGMENT	VARCHAR	identifiant fact segment (clé primaire)
ID_CLIENT	INT	identifiant des clients (clé étrangère)
ID_AGENCE	VARCHAR	identifiant des Agences (clé étrangère)
SEGEMNT	VARCHAR	types de chaques segments

FACT AUTHENTIFICATION ; notre objectif est de Suivre l'utilisation des canaux de connexion (web ou application) permet de comprendre les préférences des utilisateurs et d'adapter la stratégie de développement en conséquence.

TABLE 3.14 – fact_authentification : fact authentification

COIONNES	TYPES	LIBELLE
ID_AUTHEN	BIGINT	identifiant fact authentification (clé primaire)
ID_DATE_AUTH	VARCHAR	identifiant dim date authenfication (clé étrangère)
ID_SOUSCRIT	VARCHAR	identifiant des clients souscrits (clé étrangère)
ID_CLIENT	INT	identifiant des clients(clé étrangère
CANAL	VARCHAR	types de canal utilisé web ou mobile
ERROR	VARCHAR	erreur description
DATA_AUTH	DATETIME	date d'authentification

FACT VIREMENT : notre objectif est d'analyser l'expérience de transfert d'argent du client via l'application mobile, d'identifier les faiblesses et les points à améliorer, et d'analyser la fréquence des transferts de chaque client.

TABLE 3.15 – fact_virement : fact virement

COIONNES	TYPES	LIBELLE
ID_VIREMENT	BIGINT	identifiant fact VIREMENT (clé primaire)
ID_CLIENT	INT	identifiant dim CLIENT (clé étrangère)
ID_DATE_VIRMT	BIGINT	identifiant DIM DATE VIREMENT (clé étrangère)
ID_SOUSCRIT	VARCHAR	identifiant DIM souscription (clé étrangère)
SORT_ETP	VARCHAR	
ERROR_DESC	VARCHAR	erreur description
COMPTE_id	NUMERIC	identifiant du compte
COMPTE_BNF	NUMERIC	identifiant du compte benifçaire
COMPTE_MOTIF	VARCHAR	le motif de compte
NATR_VIRM	VARCHAR	nature de virement
MONT_VRM	MONEY	montant
DEVS	VARCHAR	devis

FACT CHEQUIER :notre objectifs est de Suivre la fréquence des demandes de chéquiers pour comprendre les besoins des clients et pour ajuster la production de chéquiers en conséquence.

TABLE 3.16 – fact_chéquier : fact chequier

COIONNES	TYPES	LIBELLE
ID_CHEQUIER	VARCHAR	identifiant fact authentification (clé primaire)
ID_CLIENT	INT	idantifiant des clients(clé étrangère)
ID_SOUSCRIT	VARCHAR	idantifiant des clients souscrits (clé étrangère)
ID_DATE_CHEQ	BIGINT	identifiant chequier (clé étrangère)
EVENR_CATEGORY	VARCHAR	evenement category
EVENT_DESC	VARCHAR	evenement description

FACT_OBJECTIFS : notre objectifs est de suivre les réalisations par rapport aux objectifs par agence, zone et région

TABLE 3.17 – fact_objectifs : fact objectifs

COIONNES	TYPES	LIBELLE
ID_OBJECTIF	VARCHAR	identifiant fact authentification (clé primaire)
ID_AGENCE	VARCHAR	identifiant AGENCE (clé étrangère)
ID_ZONE	INT	identifiant ZONE (clé étrangère)
ID_REGION	INT	identifiant REGION (clé étrangère)
NB_OBJ_CLIENT	NUMERIC	evenement category
NB_OBJ_SOUSCRIT	NUMERIC	evenement description
DATE_OBJECTIFS	DATETIME	jours de souscriptions

FACT_TYPES_ACRES Notre objectif est de mesurer la performance de notre application mobile bancaire en fonction du type d'accès (complet ou réduit) pour mieux comprendre les différences de performance entre les groupes d'utilisateurs et prendre des décisions pour améliorer l'expérience utilisateur et la croissance de notre clientèle

TABLE 3.18 – fact_types_acces : fact types d'accés

COIONNES	TYPES	LIBELLE
ID_TYPES_ACRES	VARCHAR	identifiant types acces (clé primaire)
ID_CLIENT	INT	identifiant client (clé étrangère)
ID_SOUSCRIT	VARCHAR	identifiant souscrit(clé étrangère)
TYPES_ACRES	VARCHAR	les types d'accès complet ou reduit

3.3 Modélisation du Data Warehouse

Pour la modélisation de notre data warehouse, nous avons décidé d'utiliser le modèle en constellation pour répondre de manière optimale à nos besoins d'analyse de données.

3.3.1 Modèles en Constellation

Selon Ralph Kimball, le modèle en constellation est une variante du modèle en étoile, qui permet de relier plusieurs tables de faits à des tables de dimensions différentes. Les tables de dimension dans le modèle en constellation peuvent être partagées entre plusieurs tables de faits, mais chaque table de fait contient au moins une dimension unique. Le modèle en constellation est utile dans les environnements de Data Warehouse où plusieurs tables de faits ont des structures de granularité différentes[KR13]

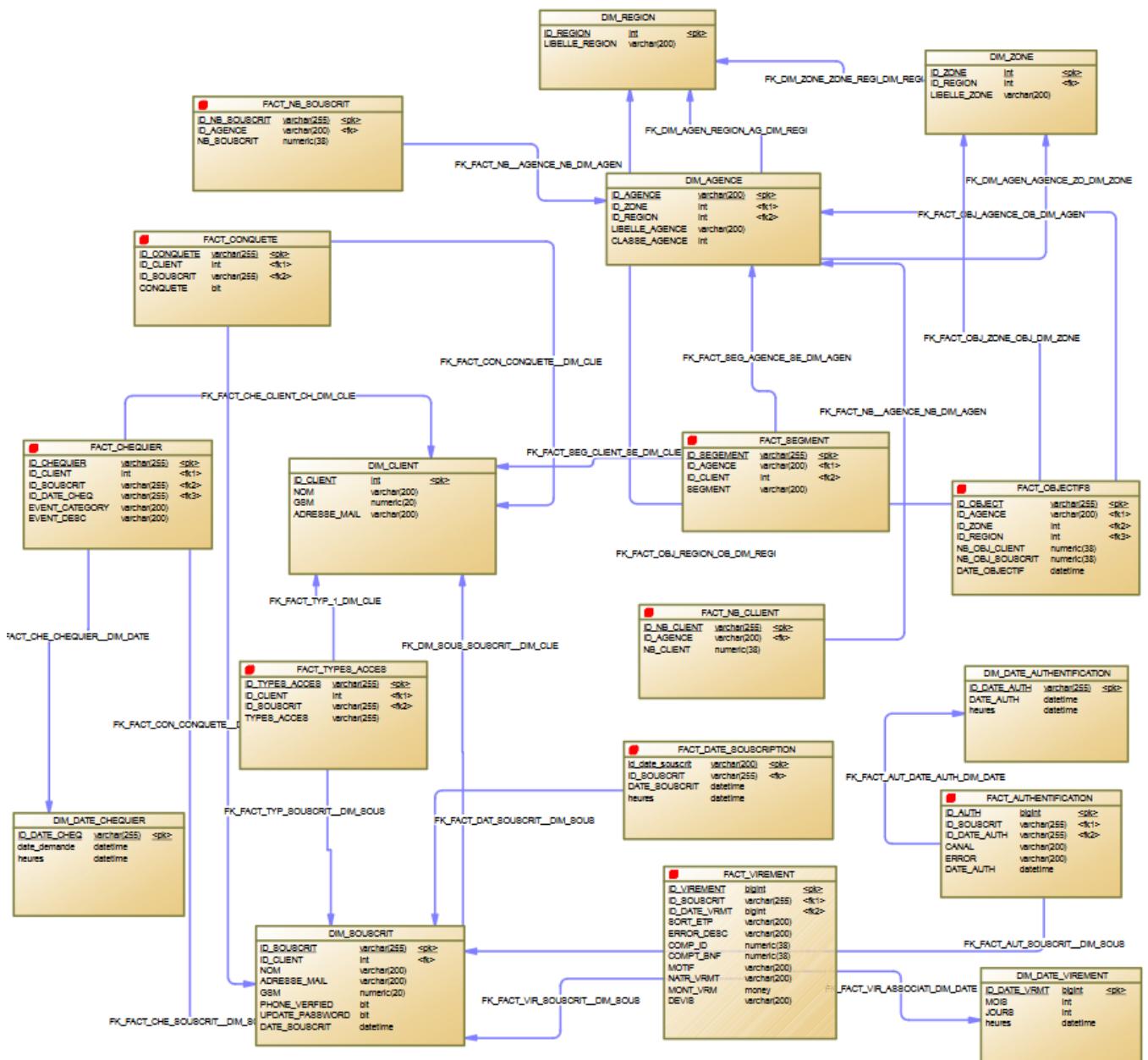


FIGURE 3.5 – modèles en constellation

3.4 Mise En Place Du Data warehouse

3.4.1 Data Warehouse my_biat_dwh

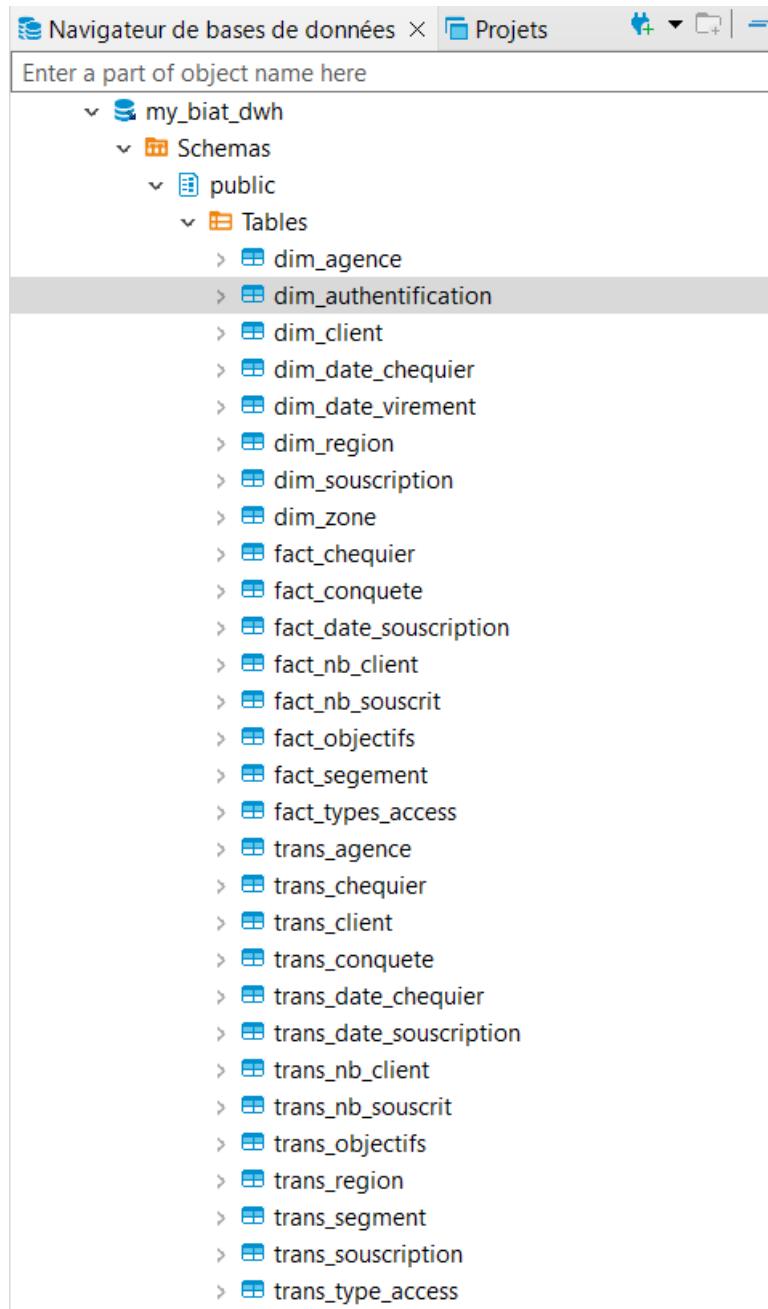


FIGURE 3.6 – my_biat_dwh

Remarque : Les tables trans sont des tables temporaires qui ne sont pas reliées entre elles. Elles servent uniquement de stockage temporaire des données avant leur insertion ou mise à jour dans les tables cibles de faits et de dimensions grâce à l'opération upsert. Une fois que les données ont été chargées dans les tables cibles, le contenu des tables de transaction est supprimé et l'opération est répétée chaque fois qu'il y a de nouvelles données

3.4.2 Diagramme my_biat_dwh

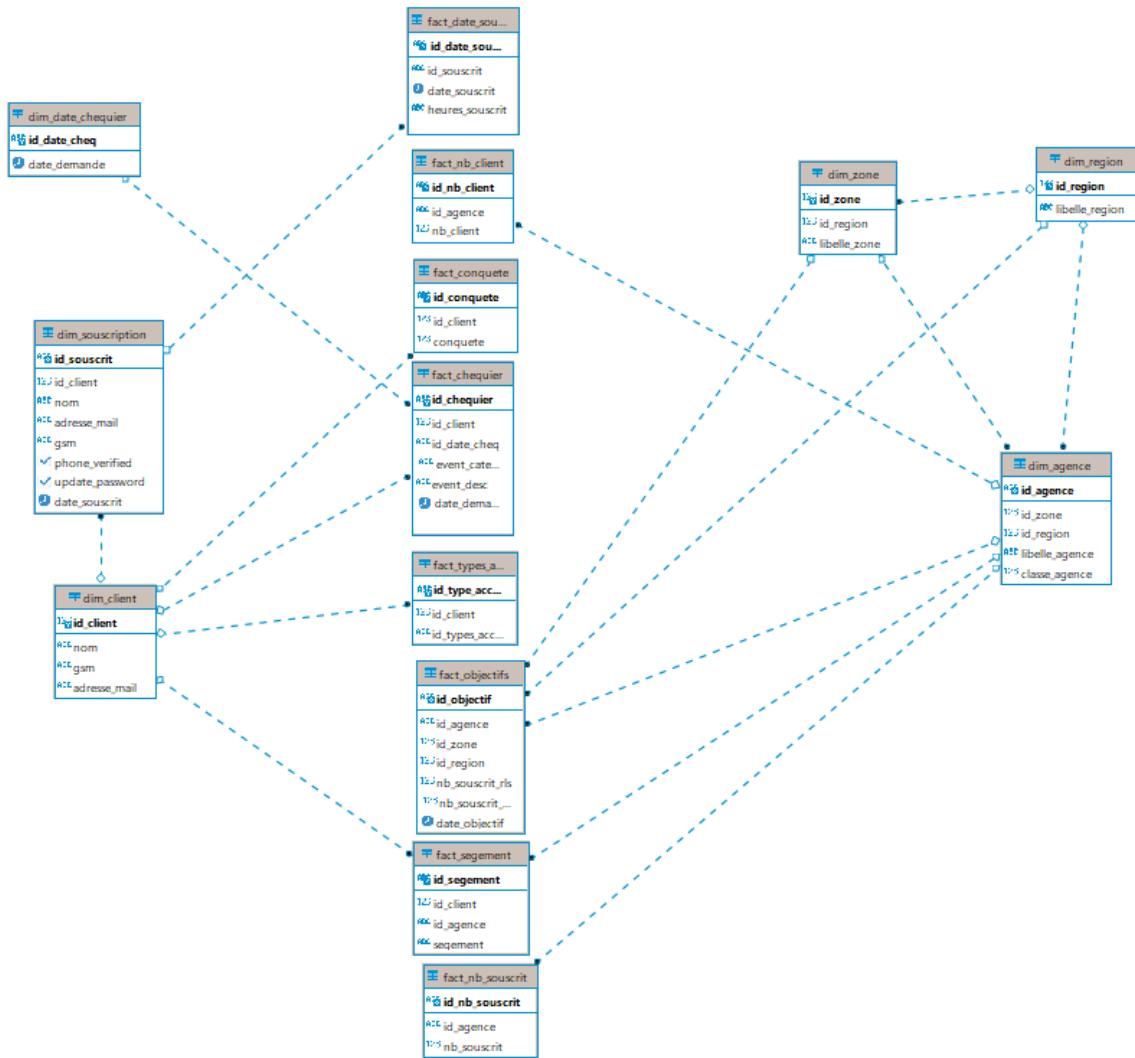


FIGURE 3.7 – Digrame 1 : Modèle en Constellation

Conclusions

Ce chapitre offre une vue d'ensemble du processus de modélisation des dimensions utilisés pour réaliser l'entrepôt de données et explique les différentes méthodes pour effectuer cette tâche. Il est important de souligner que dans le chapitre suivant, Nous traiterons de l'étape ETL, (Extract, Transform, Load)

Chapitre 4

Mise En oeuvre ETL

Sommaire

Introduction	44
4.1 Présentation de Apache Airflow	44
4.1.1 C'est quoi une Dag (Directed Acyclic Graph) ?	45
4.2 Instalation de l'environement	48
4.2.1 Création d'une machine virtuelle	48
4.2.2 Autorisation des protocoles	48
4.2.3 Instalation de docker compose , Airflow et PostgreSQL sur la machine	49
4.3 Développement de DAG	49
4.3.1 L'interface web d'Airflow	56
4.3.2 La connexion entre Airflow et PostgreSQL	56
4.3.3 Transfert des données de airflow à l'entrepôt de données	57
4.3.4 Remplissage de l'entrepôt de données	58
4.3.5 Les commandes utilisées :	58
Conclusion	60

Introduction

Pour réussir le Processus de mise en œuvre ETL, nous avons décidé d'utiliser l'outil Airflow pour automatiser le flux de données dans notre entrepôt de données en fonction des besoins de l'entreprise. Cependant, pour que cet outil fonctionne correctement, il est nécessaire de disposer d'au moins 4Go de ram par instance. Malheureusement, notre machine physique ne dispose que de 8Go de ram et 7Go utilisables. Pour résoudre ce problème nous avons décidé d'utiliser une machine virtuelle Azure

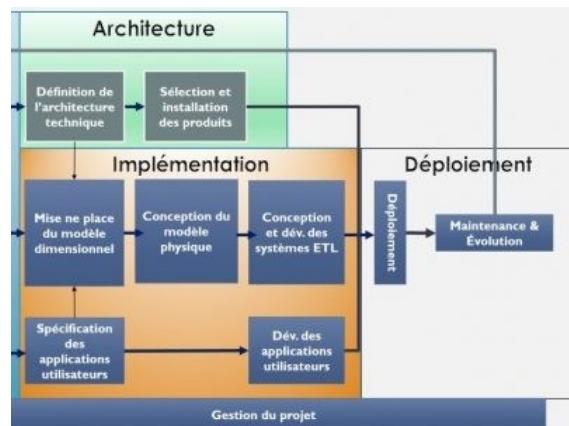


FIGURE 4.1 – Cycle de vie de la méthodologie Kimball

4.1 Présentation de Apache Airflow

L'architecture fondamentale d'Airflow avec les exécuteurs Local et séquentiel est principalement destinée à une utilisation en développement, et offre un point de départ intéressant pour comprendre l'architecture globale d'Apache Airflow.

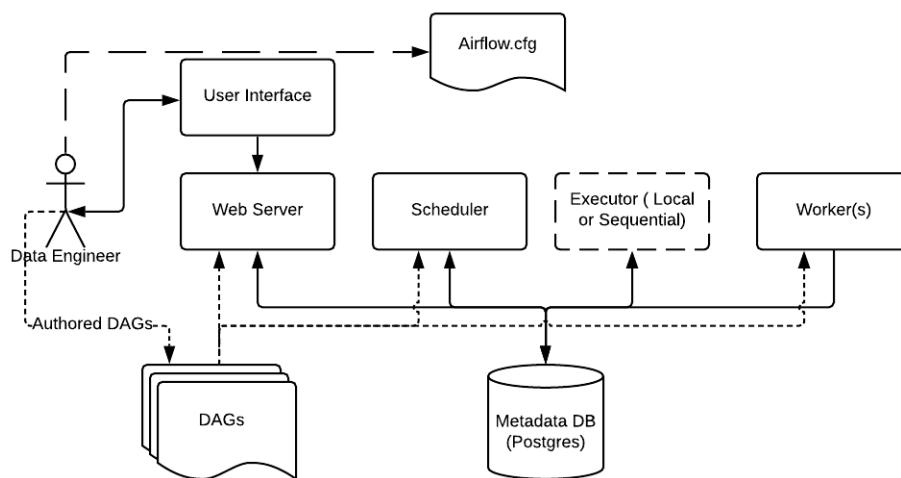


FIGURE 4.2 – architecture Airflow

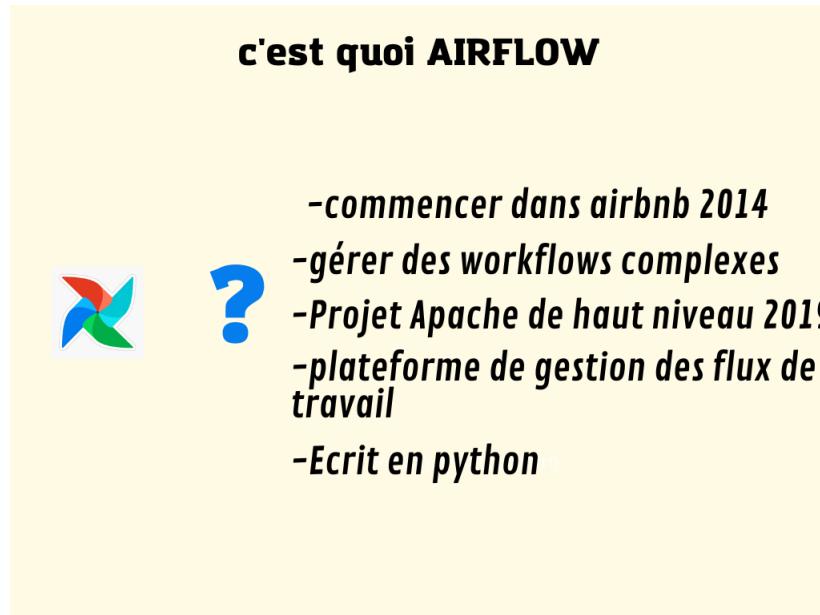


FIGURE 4.3 – Présentation Apache Airflow

Apache Airflow est un framework open-source pour la gestion des workflows et l'orchestration. Il permet de planifier, de surveiller et d'exécuter automatiquement des tâches complexes. Airflow est conçu pour gérer les workflows de traitement de données tels que l'extraction-transformation-chargement (ETL). Airflow utilise un méthode de définition de flux de travail en Python.

4.1.1 C'est quoi une Dag (Directed Acyclic Graph) ?

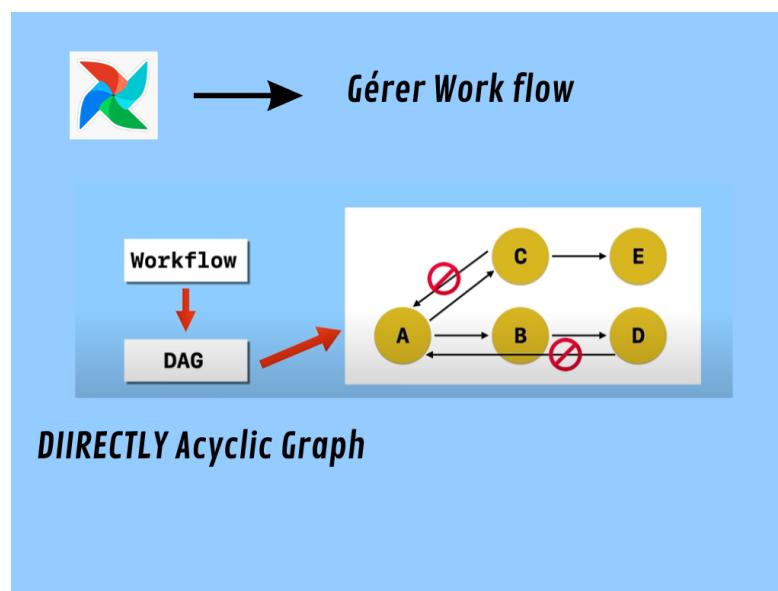


FIGURE 4.4 – Work Flow : flux de travail

Apache Airflow est une framework de gestion de flux de travail composée de graphiques de cycle dirigés (DAG).

une DAG est une collection de tâches où chaque tâche représente une unité de travail à exécuter sous la forme d'un opérateur. La DAG définit les dépendances entre les tâches, ce qui signifie qu'une tâche ne peut être exécutée que lorsque toutes ses tâches amont ont été terminées.[HR19]

4.1.1.1 C'est quoi une tache (task) ?

Dans Airflow, une tâche est la plus petite unité de travail. Une instance de tâche est l'unité d'exécution pour une tâche dans une DAG. Les tâches sont généralement définies comme des opérateurs tel que BashOperator(exécute un script Bash), PythonOperator(exécute une fonction Python), EmailOperator(envoie un e-mail), SQLOperator(exécute une requête SQL) qui implémentent le travail à effectuer. Chaque tâche dans une DAG représente une unité de travail qui peut être exécutée indépendamment. Les tâches peuvent être enchaînées pour former un workflow, et les dépendances peuvent être définies entre les tâches pour spécifier l'ordre dans lequel les tâches doivent être exécutées. Une tâche peut également avoir plusieurs tâches descendantes, qui ne commenceront à s'exécuter qu'après que la tâche a été exécutée avec succès. Les tâches dans une DAG peuvent être exécutées en parallèle tant que leurs dépendances sont satisfaites.[BCJ17]

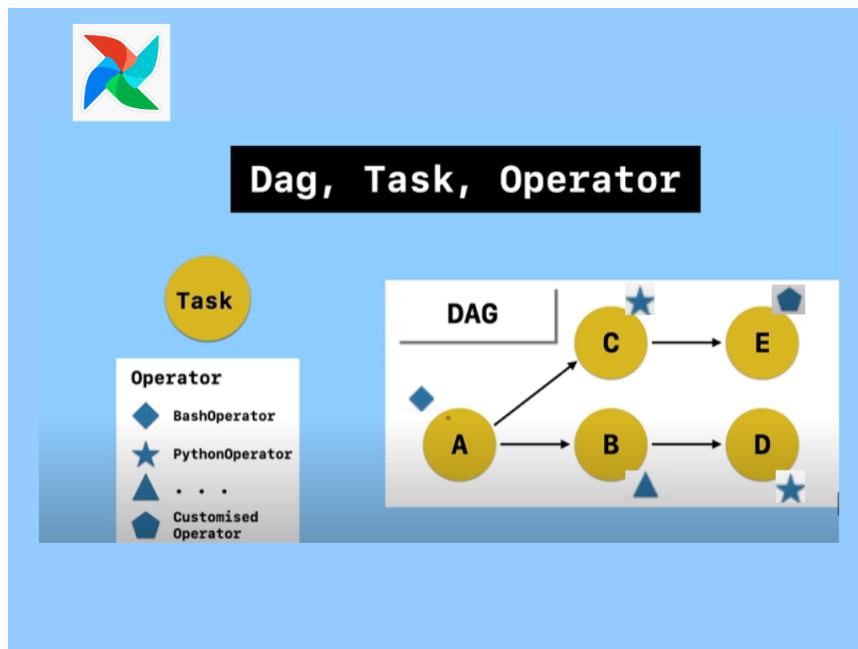


FIGURE 4.5 – DAG : graphiques de cycle dirigés

4.1.1.2 Le cycle de vie d'une tâche

Le cycle de vie d'une tâche dans la DAG dans Airflow commence par sa création, où elle est ajoutée au DAG et configurée avec les paramètres nécessaires à son exécution. La DAG est ensuite planifiée pour s'exécuter à une date et une heure spécifiques dans le futur en fonction des dépendances et des planifications définies dans le DAG.

Une fois la tâche exécutée, elle passera par différents états, tels que la préparation, l'exécution, la mise en pause , l'annulation ou l'échec. Par exemple, lorsque nous attendons les ressources nécessaires, la tâche peut également être suspendue pendant un certain temps.

Après avoir terminé avec succès la tâche, il sera transféré à l'état "complet". Cependant, si la tâche échoue, elle entrera dans l'état de la «rétention» et la configurera pour récupérer automatiquement si nécessaire.

Enfin, une fois que toutes les tâches de la DAG sont terminées, la DAG elle-même est considérée comme terminée.

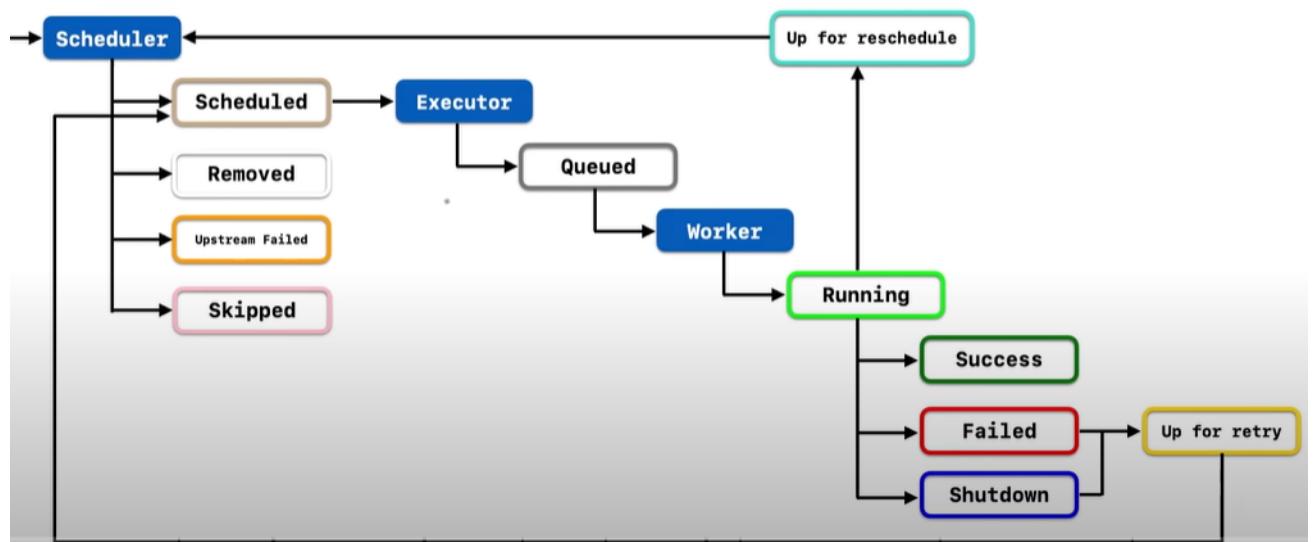


FIGURE 4.6 – Cycle de vie d'une tache

4.2 Instalation de l'environement

4.2.1 Cr eation d'une machine virtuelle

The screenshot shows the Azure portal interface for managing a virtual machine. The main pane displays the 'airflowazure' VM details, including its configuration (Linux, Standard D2s v3), network settings (public IP 20.199.115.19, private IP 10.0.0.4, subnet airflowazure-vnet/default), and current status (Stopped). The left sidebar provides navigation links for the VM's activity, monitoring, and configuration. The bottom of the screen shows the Windows taskbar with various pinned icons.

FIGURE 4.7 – Notre machine virtuelle

4.2.2 Autorisation des protocoles

This screenshot shows the 'Mise en réseau' (Network) section for the 'airflowazure121' interface. It lists security rules for the interface. The table below shows the rules:

Priorit�	Nom	Port	Protocole	Source	Destination	Action
300	SSH	22	TCP	N'importe lequel	N'importe lequel	<input checked="" type="checkbox"/> Autoriser
310	airflow	8080	N'importe lequel	N'importe lequel	N'importe lequel	<input checked="" type="checkbox"/> Autoriser
330	AllowAnyPostgreSQLInbound	5432	TCP	N'importe lequel	N'importe lequel	<input checked="" type="checkbox"/> Autoriser
65000	AllowVnetInbound	N'importe lequel	N'importe lequel	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Autoriser
65001	AllowAzureLoadBalancerInbound	N'importe lequel	N'importe lequel	AzureLoadBalancer	N'importe lequel	<input checked="" type="checkbox"/> Autoriser
65500	DenyAllInbound	N'importe lequel	N'importe lequel	N'importe lequel	N'importe lequel	<input checked="" type="checkbox"/> Refuser

FIGURE 4.8 – r seaux autoris 

SSH : pour ssh on a utilisé le port 22 pour établir shell distant à notre machine virtuelle

HTTP : on a utilisé le port HTTP 8080 pour afficher l'interface web d'Airflow.

FTP : le protocole FTP utilisé pour le transfert des fichiers entre les deux machines

4.2.3 Instalation de docker compose , Airflow et PostgreSQL sur la machine

- Docker est un outil de gestion d'applications multi-conteneurs. Il aide à définir, configurer et déployer ces applications, les réseaux et les volumes.
- Pour installer Docker sur notre machine, nous allons suivre les instructions disponibles sur le site officiel de Docker en fonction de notre système d'exploitation.
- Pour installer Airflow nous allons suivre les instructions disponibles sur le site officiel.
- PostgreSQL est une base de données relationnelles Pour installer PostgreSQL sur notre machine , nous allons suivre les instructions sur le site officiel.

4.3 Développement de DAG

Notre DAG se compose de trois tâches conçues pour automatiser le traitement et le flux de données à partir des fichiers Excel vers notre entrepôt de données . la DAG est conçue pour garantir que chaque tâche est exécutée dans un ordre spécifique, permettant une gestion efficace et optimale du flux de travail

```
default_args = {
    'owner': 'airflow',
    'retries': 5,
    'retry_delay': timedelta(minutes=2)
}
```

FIGURE 4.9 – Default_args

Ce code définit un dictionnaire nommé default_args qui contient trois clés : 'owner', 'retries' et 'retry_delay'. La clé 'owner' a pour valeur la chaîne de caractères 'airflow', ce qui peut être utilisé pour indiquer le propriétaire du flux de travail dans lequel ces

paramètres seront utilisés.

La clé 'retries' a pour valeur 5, ce qui signifie que si une tâche échoue, elle sera réexécutée jusqu'à 5 fois avant d'être marquée comme échouée.

La clé 'retry_delay' a pour valeur timedelta(minutes=2), ce qui signifie qu'il y aura un délai de 2 minutes entre chaque tentative de relance d'une tâche échouée.

```
def get_conn():
    pg_hook = PostgresHook(postgres_conn_id='postgres_conn_obj')
    engine = pg_hook.get_sqlalchemy_engine()
    return engine
```

FIGURE 4.10 – get_conn

Ce code définit une fonction appelée get_conn(). Cette fonction utilise le module PostgresHook de la bibliothèque Apache Airflow pour se connecter à une entrepôt de données PostgreSQL cette fonction va être utilisée dans les fonctions de traitement de données.

La première ligne de la fonction crée une instance de l'objet PostgresHook, en utilisant le paramètre postgres_conn_id qui est défini à la valeur 'postgres_conn_obj'.

La deuxième ligne de la fonction utilise la méthode get_sqlalchemy_engine() en utilisons l'objet PostgresHook pour obtenir une connexion de base de données PostgreSQL. SQLAlchemy est une bibliothèque Python pour la gestion des bases de données relationnelles.

La fonction retourne finalement l'instance du moteur de base de données SQLAlchemy, qui peut être utilisée pour exécuter des requêtes SQL.

```
def delete_trans_tables_content():
    conn = get_conn()
    for table, meta in table_meta.items():
        conn.execute(f"DELETE FROM {meta['trans_table']}")
```

FIGURE 4.11 – delete_trans_tables_content

Ce code définit une fonction appelée delete_trans_tables_content() qui peut être utilisée pour supprimer le contenu des tables transactionnelle de notre entrepôt de données.

La deuxième ligne de la fonction démarre une boucle for qui itère sur chaque élément du l'objet table_meta. Ce dictionnaire contenir les métadonnées des tables, telles que les noms et les tables de transaction.

La troisième ligne de la fonction utilise la méthode execute() de l'objet de connexion conn pour exécuter une requête SQL DELETE sur chaque table de transaction spécifiée dans l'objet table_meta.

```
def create_tables():
    conn = get_conn()
    conn.execute(postgres_create_query)
    print('created tables !!')
```

FIGURE 4.12 – create_tables

Ce code définit une fonction appelée create_tables() qui est utilisée pour créer des tables dans l'entrepôt de données si elles n'existent pas.

La deuxième ligne de la fonction utilise la méthode execute() de l'objet de connexion conn pour exécuter une requête SQL qui crée les tables. Cette requête SQL doit être stockée dans une variable nommée postgres_create_query. La requête SQL doit être une chaîne de caractères valide, qui peut contenir plusieurs instructions SQL pour créer des tables, des colonnes, des contraintes, etc.

```
def insert_query(upsert_query, trans_table, df):
    conn = get_conn()
    df.to_sql(
        trans_table,
        conn,
        index=False,
        if_exists='append',
        schema='public'
    )
    conn.execute(upsert_query)
```

FIGURE 4.13 – insert_query

Ce code définit une fonction appelée insert_query() qui est utilisée pour insérer des données dans les tables

- upsert_query : une requête SQL qui sera exécutée pour insérer ou mettre à jour des données dans la table cible.
- trans_table : le nom de la table de transaction qui contient les données à insérer dans la table cible.
- df : un objet DataFrame de pandas qui contient les données extrait des fichiers excels.

```
def process_excel():
    for table_name, meta in table_meta.items():
        file_path = os.path.abspath(meta['file_path'])
        df = pd.read_excel(file_path)
        df_length = len(df)
        if df_length >= 100000:
            for i in range(0, df_length, 50000):
                df_chunk = df.iloc[i:i+50000]
                insert_query(meta['upsert_query'],
                             meta['trans_table'],
                             df_chunk)
        else:
            insert_query(meta['upsert_query'], meta['trans_table'], df)
```

FIGURE 4.14 – process_excel

Ce code définit une fonction appelée `process_excel()` qui est utilisée pour lire les données à partir de fichiers Excel et les insérer dans une base de données PostgreSQL en utilisant la fonction `insert_query()` définie précédemment.

```
with DAG(
    dag_id='excel_db_etl',
    default_args=default_args,
    description='test test',
    schedule_interval='@daily',
    start_date=datetime(2023, 05, 24, 2),
    max_active_runs=1
) as dag:
    create_tables_if_not_exists = PythonOperator(
        task_id='create_db_tables_if_not_exists',
        python_callable=create_tables,
        dag=dag,
    )
    etl_process = PythonOperator(
        task_id='process_excel_to_db',
        python_callable=process_excel,
        dag=dag,
    )
    delete_trans_table_content = PythonOperator(
        task_id='delete_transaction_tables_content',
        python_callable=delete_trans_tables_content,
        dag=dag,
    )

    create_tables_if_not_exists >> etl_process >> delete_trans_table_content
```

FIGURE 4.15 – DAG

Dans ce code j'ai crée trois tâches dans une DAG nommée "excel_db_etl". La première tâche utilise la fonction "create_tables" pour créer des tables dans un entrepôt de données PostgreSQL. La deuxième tâche utilise la fonction "process_excel" pour lire des fichiers Excels et insérer les données dans l'entrepôt de données créée précédemment. La troisième tâche utilise la fonction "delete_trans_tables_content" pour supprimer le contenu des tables de transaction.

```
'postgres_create_query='''  
  
        CREATE TABLE IF NOT EXISTS dim_region (  
            id_region INT PRIMARY KEY NOT NULL,  
            libelle_region VARCHAR(255),  
            CONSTRAINT id_region_unique UNIQUE (id_region)  
        );  
  
        CREATE TABLE IF NOT EXISTS dim_zone (  
            id_zone INT PRIMARY KEY NOT NULL,  
            id_region INT REFERENCES dim_region(id_region) NULL ,  
            libelle_zone VARCHAR(255),  
            CONSTRAINT zone_unique_id_fk UNIQUE (id_zone)  
        );  
  
        CREATE TABLE IF NOT EXISTS dim_agence (  
            id_agence VARCHAR(4) PRIMARY KEY NOT NULL,  
            id_zone INT REFERENCES dim_zone(id_zone) NULL ,  
            id_region INT REFERENCES dim_region(id_region) NULL,  
            libelle_agence VARCHAR(255) NULL,  
            classe_agence INTEGER,  
            CONSTRAINT agence_unique_id UNIQUE (id_agence)  
        );
```

FIGURE 4.16 – postgres_create_query

Ce code crée les tables "dim_region" et "dim_zone", ainsi que toutes les autres tables de notre entrepôt de données PostgreSQL, si elles n'existent pas déjà. Il est possible de modifier notre entrepôt de données en modifiant les requêtes sql .

```
152 table_meta = [
153
154     'dim_region': {
155         'file_path': 'dags/sheets/DIM_REGION.xlsx',
156         'trans_table': 'trans_region',
157         'upsert_query':
158             '''
159             INSERT INTO dim_region (id_region, libelle_region)
160             SELECT "ID_REGION", "LIBELLE_REGION" FROM trans_region
161             ON CONFLICT (id_region) DO
162                 UPDATE SET libelle_region = EXCLUDED.libelle_region
163             '''
164     },
165
166     'dim_zone': {
167         'file_path': 'dags/sheets/DIM_ZONE.xlsx',
168         'trans_table': 'trans_zone',
169         'upsert_query':
170             '''
171             INSERT INTO public.dim_zone (id_zone, libelle_zone, id_region)
172             SELECT "ID_ZONE","LIBELLE_ZONE","ID_REGION " FROM public.trans_zone
173             ON CONFLICT (id_zone) DO
174                 UPDATE SET
175                     libelle_zone = EXCLUDED.libelle_zone,
176                     id_region = EXCLUDED.id_region
177             '''
```

FIGURE 4.17 – table_meta

Ce code définit un objet appelé ”table_meta” qui contient les meta donnée : ”dim_region” et ”dim_zone” et d’autres tables de notre entrepôt. Chacune de ces entrées possède elle-même trois clés :

”file_path” : contient le chemin d'accès au fichier Excel contenant les données tabulaires.

”trans_table” : contient le nom de la table de transaction.

”upsert_query” : contient la requête SQL d'insertion et de mise à jour des données dans la table finale.

4.3.1 L'interface web d'Airflow

4.3.2 La connexion entre Airflow et PostgreSQL

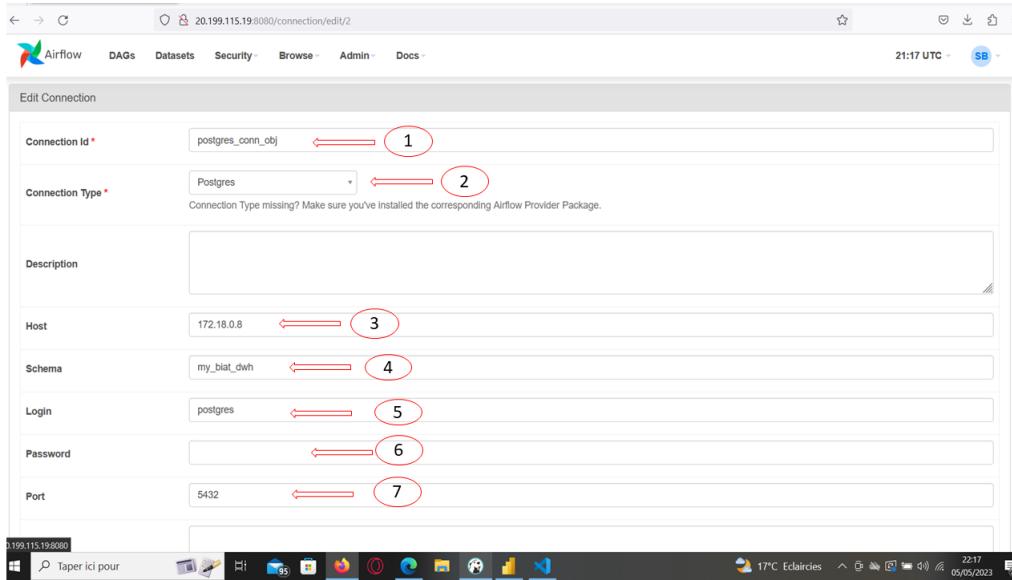


FIGURE 4.18 – connexion

c'est pour la définition des paramètres de connexion a l'entrepôt de donnée :

- 1) Définissez le paramètre de connexion a une entrepôt de donnée
- 2) Sélectionnez le type de base de données
- 3) Entrez notre adresse IP
- 4) Entrez le nom de notre entrepôt de données
- 5) login le nom utilisateur
- 6) password *****
- 7) définir le port logique utilisé par l'entrepôt de données

4.3.3 Transfert des données de airflow à l'entrepôt de données

Le vert dans un cercle rouge signifie que notre dag a réussi

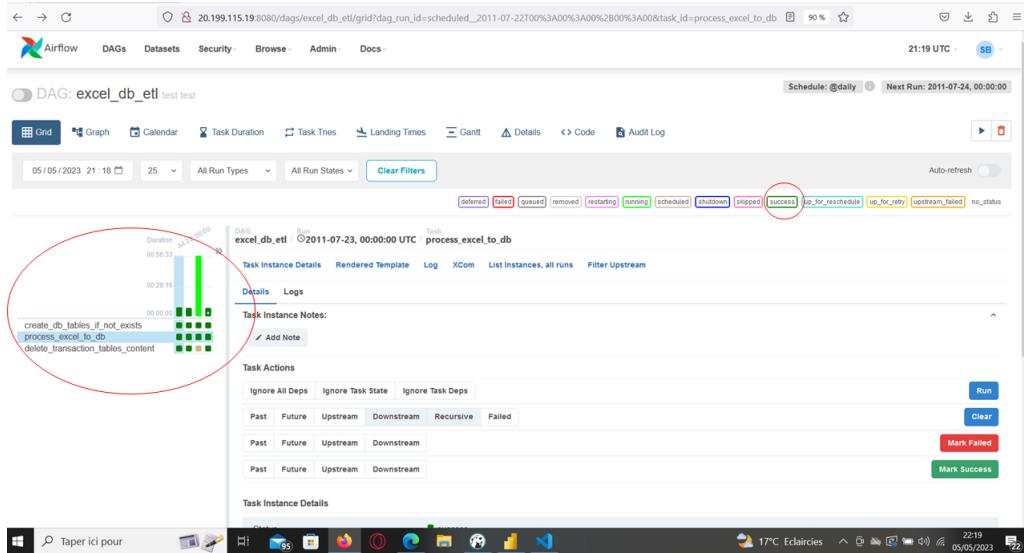


FIGURE 4.19 – succès

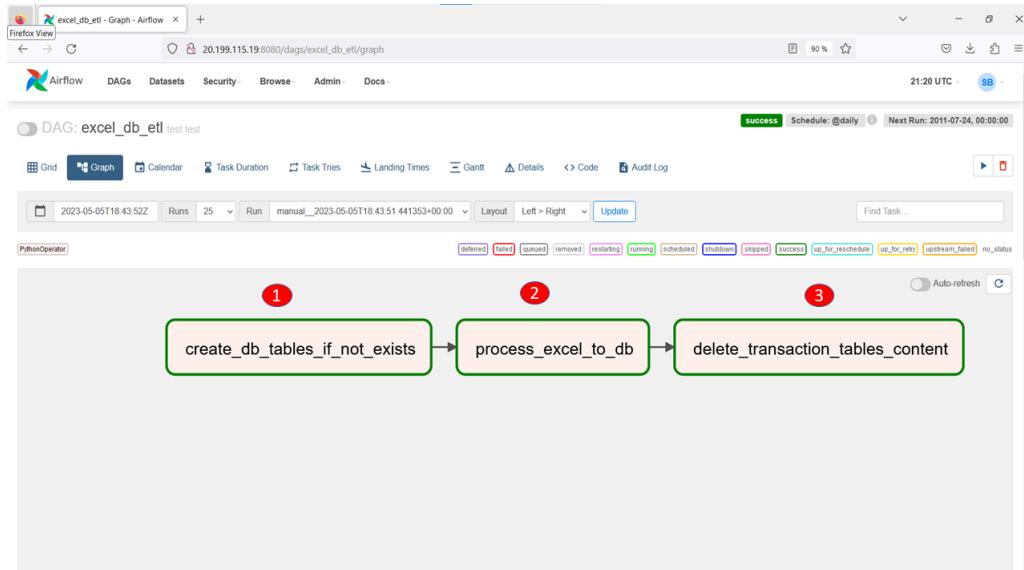


FIGURE 4.20 – les tâches

4.3.4 Remplissage de l'entrepôt de données

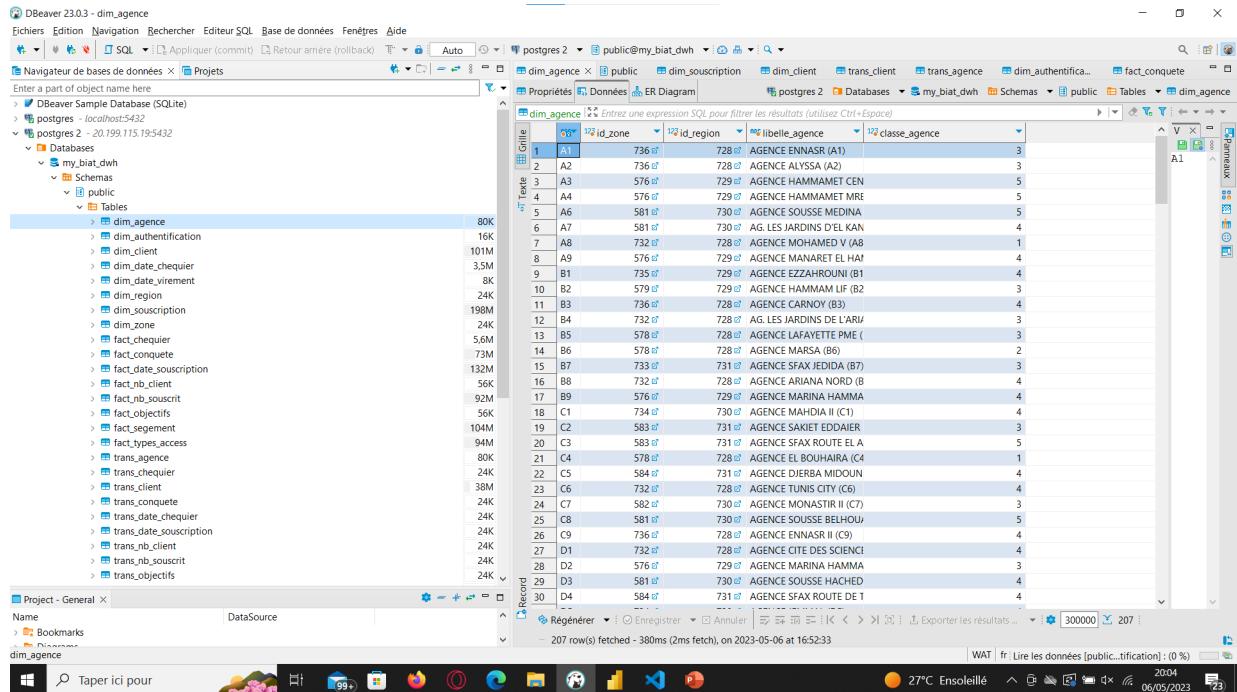


FIGURE 4.21 – Transfert des données

4.3.5 Les commandes utilisées :

Utilisée dans les systèmes d'exploitation de type Unix :

- ssh-keygen -t rsa

est utilisée pour générer une paire de clés SSH avec l'algorithme de chiffrement RSA

- ssh-copy-id -i /Dossier_cles/Nom_cles.pub Nom_Utilisateur@address_de_serveur;

La commande ssh-copy-id est utilisée pour copier une clé publique sur un serveur distant pour permettre une authentification lors de la connexion SSH.

-i est une option de la commande qui spécifie le chemin de la clés

- ssh Nom_utilisateur@address_ip ou le Nom de domaine

ssh est la commande qui permet d'établir une connexion SSH à un serveur distant.

- exit

Utiliser pour quitter la shell

- scp Nom_fichier Nom_utilisateur@address_ip :emplacement_de_fichier

est utilisée pour copier des fichiers entre un hôte local et un hôte distant via une connexion SSH

- sudo (commande)

permet à un utilisateur de Linux d'exécuter une commande en tant que utilisateur ayant des priviléges élevés

- ls

lister les fichiers et répertoires dans le répertoire de travail courant

- pwd

utilisée dans les systèmes d'exploitation de type Unix pour afficher le répertoire de travail actuel de l'utilisateur (abréviation de "print working directory")

- cd

est utilisée pour changer le répertoire de travail. Supposons que nous commençons dans le répertoire de travail /home/name_user

exemples du commande cd :

- cd mywork

Cette commande changera le répertoire de travail en /home/name_user/mywork.

- cd /

Cette commande changera le répertoire de travail en la racine du système de fichiers.

- cd ../

Cette commande changera le répertoire de travail en le répertoire de niveau supérieur du répertoire actuel. Si le répertoire de travail actuel /home/name_user/mywork cette commande le changerait en /home/name_user

exemples des commandes docker et dockercompose :

- docker images ls

Elle affichera une liste de toutes les images Docker disponibles sur la machine hôte, triées par leur nom, leur tag et leur ID de référence.

- docker ps

Cette commande affichera une liste de tous les conteneurs Docker actifs sur la machine hôte, triés par leur nom ou leur ID, leur image de base, leur état d'exécution et leur date de création.

- docker ps -a
listera tous les conteneurs, actifs ou arrêtés

- docker exec Nom_container commande_a_executé
utilisée pour exécuter une commande à l'intérieur d'un conteneur Docker en cours d'exécution.

- docker network ls
permet de lister tous les réseaux Docker existants sur le système.

- docker cp chemin_vers_fichier_local nom_ou_id_conteneur :chemin_destination_dans_conteneur
permet de copier des fichiers de l'hôte Docker vers un conteneur Docker

- docker-compose up
cette commande permet d'installer des services dans un fichier de configuration de format yaml

- docker compose down
permet de supprimer tous les services dans un fichier de configuration yaml

- mv
pour déplacer ou renommer des fichiers

- rm
pour supprimer des fichiers

- rm -rf
pour supprimer des dossiers

Conclusion

Pendant ce processus, nous avons automatisé l'importation et l'exportation des données depuis des fichiers Excel vers notre entrepôt de données. Dans la prochaine tâche, nous nous concentrerons sur la création des tableaux de bord, qui seront automatisés en utilisant notre entrepôt de données.

Chapitre 5

Déploiement De Tableaux De Bord

Sommaire

Introduction	62
5.1 Objectifs	62
5.2 Connexion et importation de données	63
5.3 Tableaux de bord	64
5.3.1 Page d'accueil	64
5.3.2 Tableaux de bord du nombre de souscriptions	64
5.3.3 Tableaux de bord du nombre de clients	65
5.3.4 Tableaux de bord du conquête et le types d'accées	65
5.3.5 Tableaux de bord segments clients	66
5.3.6 Tableaux de bord des demande de chequier	66
5.3.7 Tableaux de bord des objectifs par rapport aux réalisations	67
Conclusion	67

Introduction

Ce chapitre présentera les objectifs et les réalisations des tableaux de bord. Nous allons discuter des buts que nous cherchons à atteindre , ainsi que des résultats concrets que nous avons obtenus jusqu'à présent.

5.1 Objectifs

Notre objectif est de créer des tableaux de bord qui répondent efficacement aux besoins de l'équipe marketing et développement digital, tels qu'illustrés dans la figure ci-dessous. Nous cherchons à concevoir des tableaux de bord pratiques et faciles à utiliser, en nous concentrant sur les indicateurs clés de performance (KPI) pertinents . Nous voulons également nous assurer que les tableaux de bord sont régulièrement mis à jour et qu'ils fournissent une visualisation claire et concise des données importantes pour aider l'équipes à prendre des décisions éclairées.

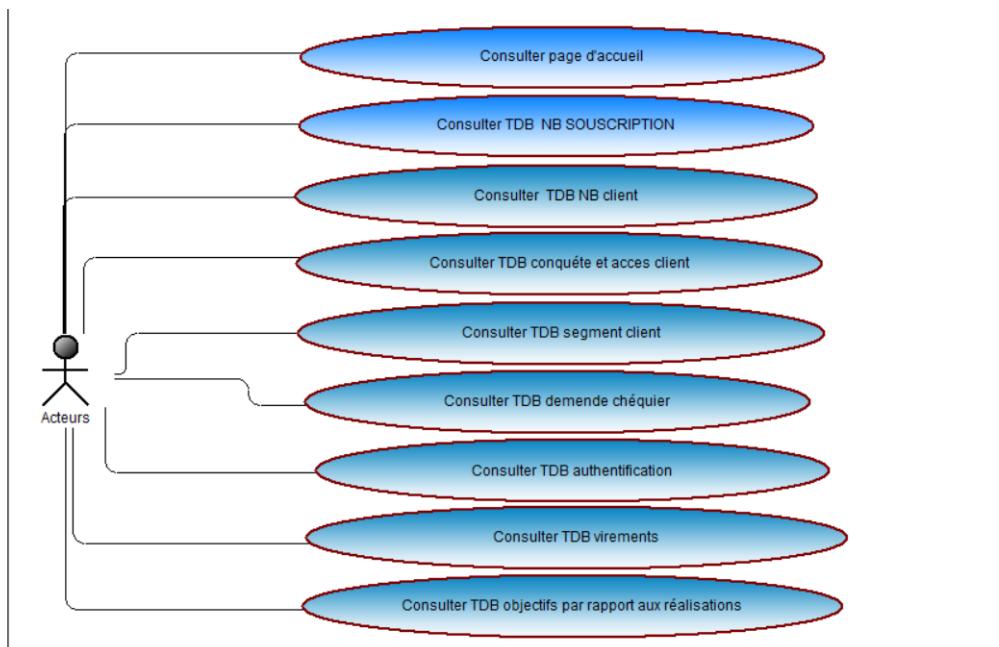


FIGURE 5.1 – Diagramme de cas d'utilisation

5.2 Connexion et importation de données

Pour effectuer la phase de reporting, power bi doit être connecté à l'entrepôt de données postgresql Pour importer les données comme indiqué sur la figure suivante :

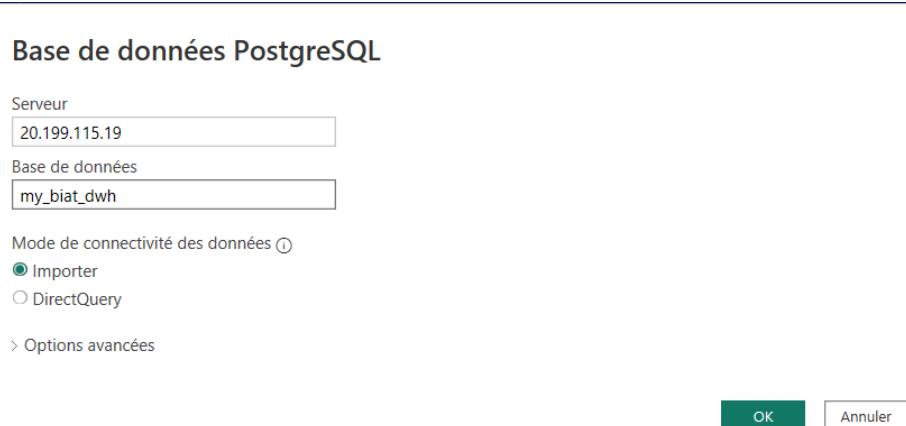


FIGURE 5.2 – connexion a l'entrepôt de données

Une fois la connexion à l'entrepôt de données PostgresSQL effectuée, nous avons choisi les dimensions et les table des faits sur les dépenses du magasin de données, comme illustré à la figure suivante :

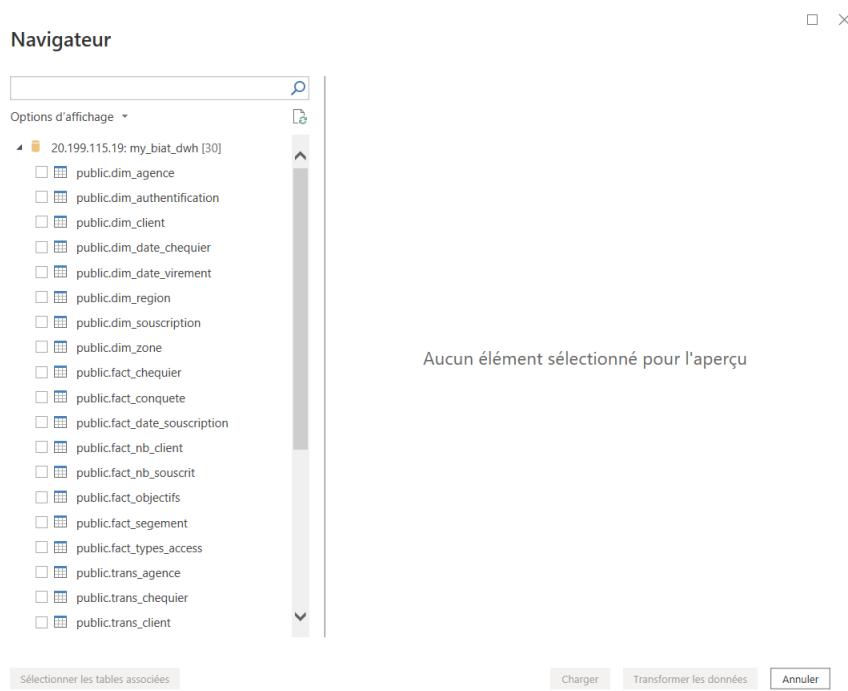


FIGURE 5.3 – importation de données

5.3 Tableaux de bord

5.3.1 Page d'accueil

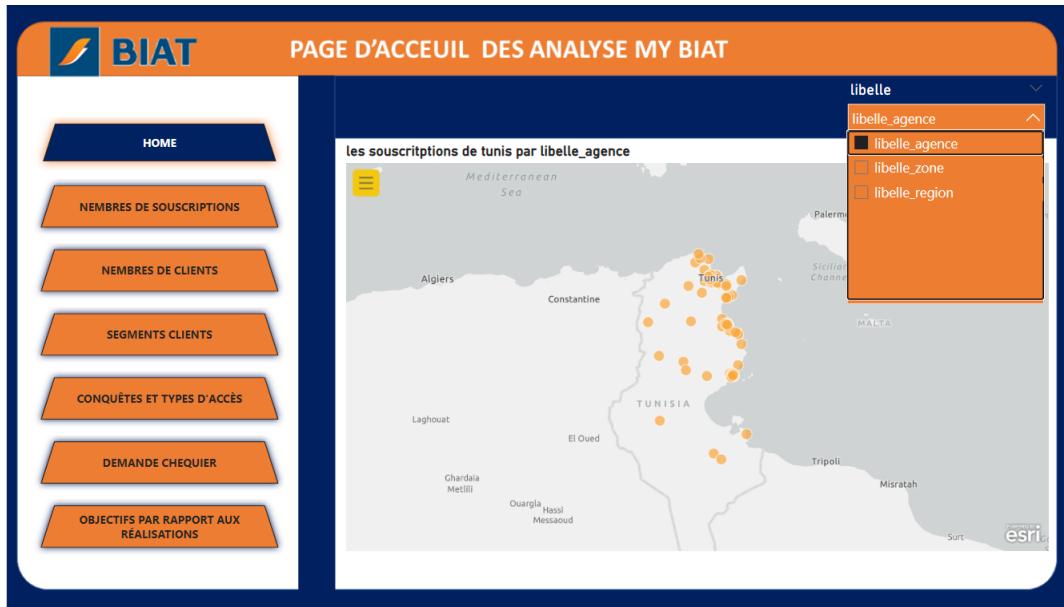


FIGURE 5.4 – HOME

5.3.2 Tableaux de bord du nombre de souscriptions

Ce tableau de bord permet de suivre l'application à travers les souscriptions par période, agence, zone et région

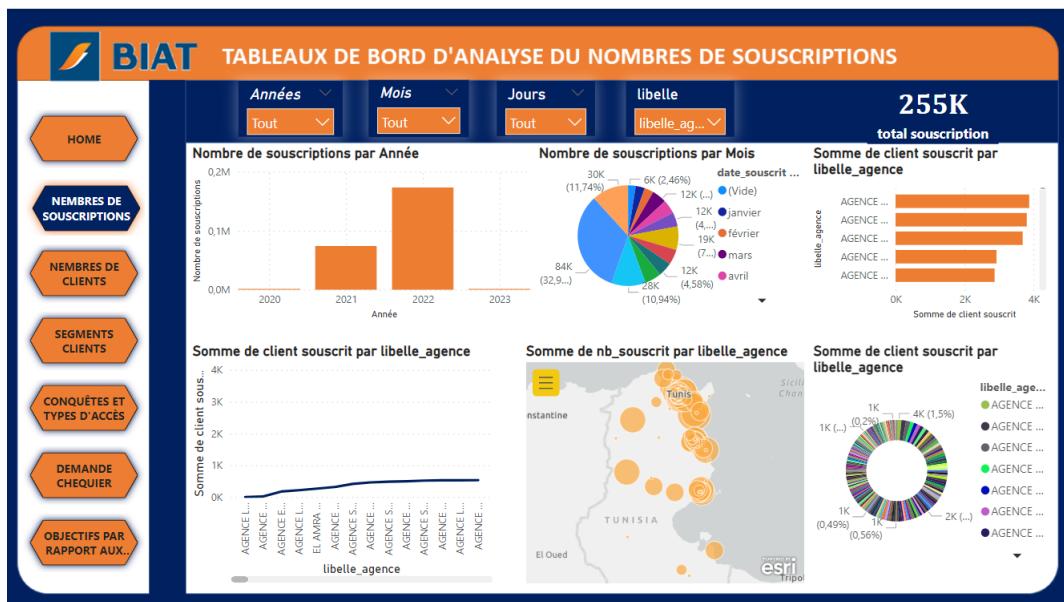


FIGURE 5.5 – TDB nombre de souscriptions

5.3.3 Tableaux de bord du nombre de clients

Ce tableau de bord permet de suivre le nombre de client par agence ,zone et région

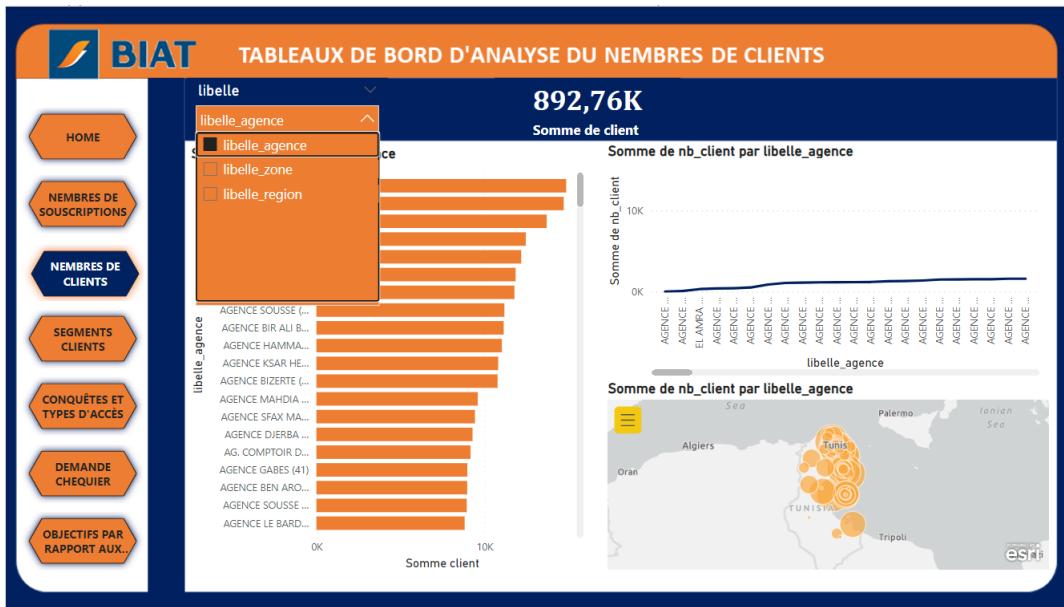


FIGURE 5.6 – TDB nombre de client

5.3.4 Tableaux de bord du conquête et le types d'accès

Ce tableau de bord permet de suivre la conquête clients et les types d'accès.

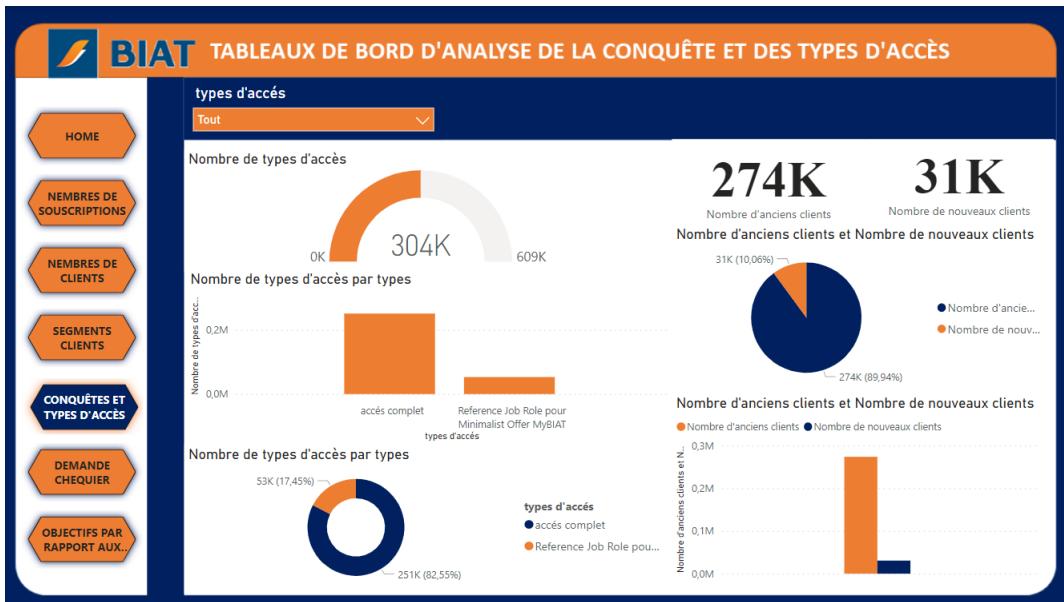


FIGURE 5.7 – TDB types d'accès et la conquête client

5.3.5 Tableaux de bord segments clients

Ce tableau de bord permet de suivre les segments client par agence, zone, région et par types de segment

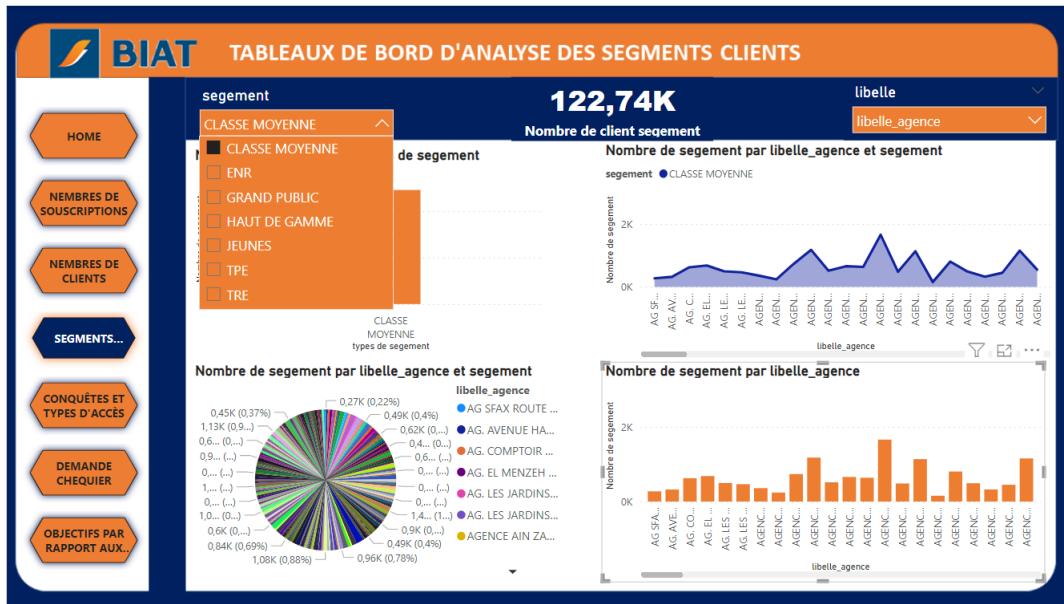


FIGURE 5.8 – TDB segment client

5.3.6 Tableaux de bord des demande de chequier

Ce tableau de bord permet de suivre l'application à travers les demandes de chéquier

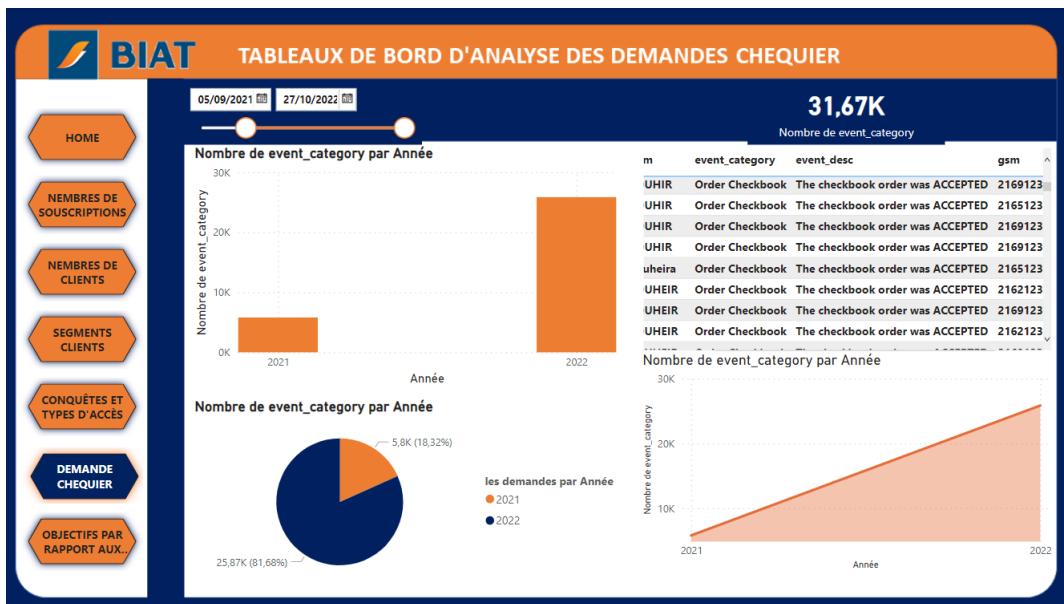


FIGURE 5.9 – TDB demande chéquier

5.3.7 Tableaux de bord des objectifs par rapport aux réalisations

Ce tableau de bord permet de suivre l'application à travers Le TRO

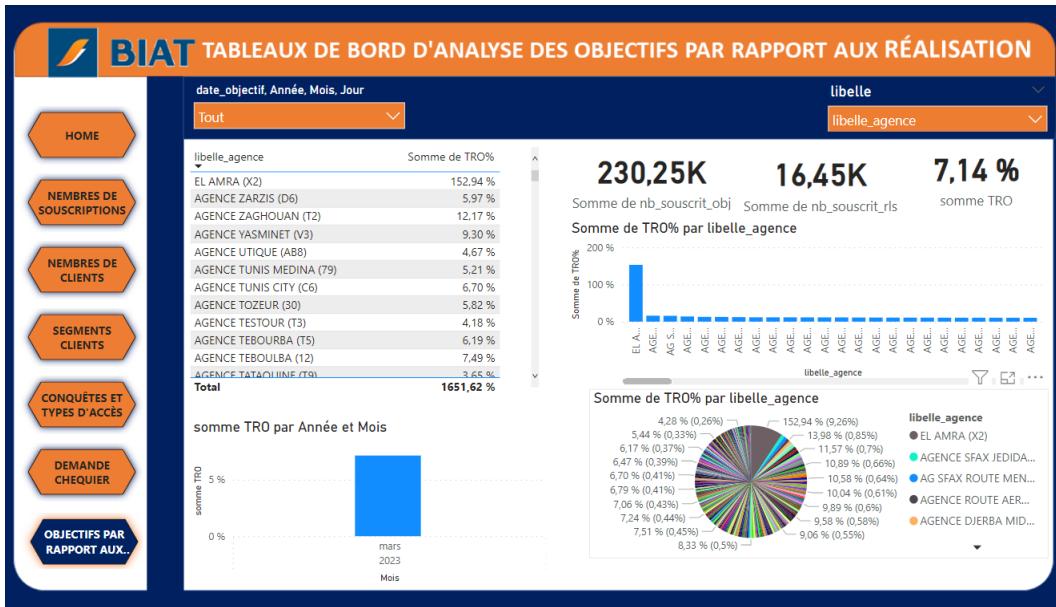


FIGURE 5.10 – TDB réalisations par rapport aux objectifs

Conclusion

La réalisation des tableaux de bord est une étape importante dans le développement d'un système d'information. Dans ce chapitre, nous avons présenté les objectifs de ce chapitre et les différentes étapes de réalisation des tableaux de bord. Nous avons notamment abordé la connexion et l'importation des données, ainsi que la création de différents tableaux de bord pour la visualisation des données.

Conclusion générale

Ce rapport est le fruit d'un stage que j'ai effectué au sein de la Banque Internationale Arabe de Tunisie (BIAT) et qui a duré trois mois et 24 jours. Ce travail a été accompli avec passion et persévérance, et j'y ai consacré beaucoup d'efforts. Ce rapport me tient vraiment à cœur.

Nous avons été affectées au département de marketing digital où notre encadrant nous a exprimé son besoin pour une solution BI qui permettra de suivre l'application My BIAT, notamment en ce qui concerne le nombre de souscriptions, le segment de clients souscrits et les réalisations par rapport aux objectifs fixés, etc. Cette solution permettra de gagner du temps dans la prise de décision grâce à l'automatisation du remplissage de l'entrepôt de données. La flexibilité de la solution permettra également d'ajouter des autres workflow. De plus, la création de tableaux de bord automatisés permettra une visualisation rapide et claire des données pour une meilleure analyse et une prise de décision plus efficace.

D'abord, pour mieux organiser et réussir un projet, il faut comprendre la problématique, décider d'une méthodologie de travail, effectuer une recherche approfondie pour déterminer l'environnement logiciel et les ressources nécessaires, identifier les besoins, choisir les méthodes de travail appropriées, et passer à la phase de développement.

Cependant, il convient de souligner que nous avons rencontré des événements imprévus pour diverses raisons. Malgré cela, nous n'avons jamais abandonné et nous avons consacré le temps nécessaire pour résoudre chaque problème, car nous croyons fermement qu'il existe une solution pour chaque difficulté rencontrée.

Les faits ont prouvé que ce stage est une mine d'informations et une source de connaissances précieuses. Il nous a offert l'opportunité de développer nos compétences techniques, notamment en acquérant une solide maîtrise des outils les plus couramment utilisés dans le domaine de l'informatique décisionnelle. Cette expérience nous sera très utile dans notre vie professionnelle.

Cette précieuse opportunité a mis en évidence notre esprit d'équipe ainsi que nos compétences en matière de communication et de relations interpersonnelles.

Bibliographie

- [BCJ17] Marco BOLKESTEIN, Andreas van CRANENBURGH et Bas de JONGH. « Apache Airflow: a platform to programmatically author, schedule, and monitor workflows ». In : *The Journal of Open Source Software* 2.18 (2017), p. 430.
- [BCM18] Maxime BOLIN, Alban CRITE et Guillaume MATHY. « Apache Airflow: A platform to programmatically author, schedule, and monitor workflows ». In : *Big Data Spain*. 2018.
- [Cha12] David CHAPPELL. « Introducing Windows Azure ». In : *Microsoft Corporation* (2012).
- [Che+05] P. CHEN et al. « PostgreSQL: A High-Performance Open-Source Object-Relational Database System ». In : *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. 2005, p. 1206-1208.
- [Far+19] Ahmad FAROOQ et al. « Review of HTTP Protocol ». In : *2019 International Conference on Innovative Computing (ICIC)*. IEEE. 2019, p. 1-5.
- [GR12] John GANTZ et David REINSEL. « The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east ». In : *IDC iView* 2012.1 (2012), p. 1-16.
- [HL14] John HAMMERSLEY et John LEES-MILLER. « Overleaf: A Collaborative La-TeX Editor ». In : *Journal of Open Source Software* 1.1 (2014), p. 42.
- [HR19] Bas P HARENSLAK et Julian Rutger de RUITER. *Data Pipelines with Apache Airflow*. Manning Publications, 2019.
- [Inm05] W. H. INMON. *Building the data warehouse*. New Jersey : John Wiley & Sons, 2005.
- [Kim+13] Ralph KIMBALL et al. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. 3rd edition. New Jersey : John Wiley & Sons, 2013.
- [KL07] Jongwook KIM et Yookun LEE. « Data warehouse design for e-commerce environment ». In : *International Journal of Software Engineering and Its Applications* 1.4 (2007), p. 3-10.

- [KR13] Ralph KIMBALL et Margy ROSS. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons, 2013.
- [Mer14] Dirk MERKEL. « Docker: lightweight linux containers for consistent development and deployment ». In : *Linux journal* 2014.239 (2014), p. 2.
- [Par15] David PARMENTER. *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*. New Jersey : John Wiley & Sons, 2015.
- [PK13] Kaveh PAHLAVAN et Prashant KRISHNAMURTHY. « Virtual machines: concepts, technology, and applications ». In : *Morgan Kaufmann* (2013).
- [RGT19] Serge RIELAU, Martin GRÖSSING et Martin TOMITSCH. « DBeaver: A Universal Database Tool ». In : *Proceedings of the 20th International Conference on Extending Database Technology (EDBT)*. ACM. 2019, p. 631-634.
- [Sel18] S SELVAKUMAR. « Power BI Desktop: The Self-Service BI Tool ». In : *International Journal of Scientific and Research Publications* 8.4 (2018), p. 302-305.
- [ST20] Madhur SHARMA et Shashank TRIPATHI. « Visual Studio Code: A Review ». In : *International Journal of Computer Science and Mobile Computing* 9 (8 2020), p. 41-45.
- [Syb03] SYBEX. *Power AMC Designer*. Wiley Publishing, 2003.

BIBLIOGRAPHIE
