

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITÉ MOHAMMED SEDDIK BENYAHIA

JIJEL

FACULTÉ DE SCIENCES EXACTES ET D'INFORMATIQUE



Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de :

MASTER

En **INFORMATIQUE**

Option : Système d'information et aide à la décision

Thème

**Développement d'une approche
d'automatisation de la conception
des entrepôts de données actifs**

Réalisé par :

Djaoui Faouzi
Merada Noureddine

Encadré par :

Mme, BOUAINAH
MADIHA

Promotion : 2022

Remerciements

*Nous tenons tout d'abord à remercier le **Dieu** tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce travail.*

*En second lieu, nous tenons à remercier notre encadreur Mme : **Bouinah Madiha** , son précieux conseil et son aide durant toute la période du travail.*

Nos vifs remerciements également aux membres du jury pour leurs attentions et intérêts portés envers notre travail. Merci de nous avoir honorés de votre présence.

Nous tenons à exprimer nos remerciements à tous les professeurs qui nous ont enseigné.

Sans oublier de présenter nos sincères remerciements à nos parents qui ont été toujours de nous encourager durant notre parcours de mes études, ainsi que pour leurs aides, leurs compréhensions et leurs soutiens.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce mémoire...

Merci à tous et à toutes

Merci 

Dédicaces

Je dédie ce travail :

*★A Mes chers **parents**, **Ms Boualem** , **Mme Louiza** , pour leurs soutient et leurs patience ;*

*★A Mes chères sœurs **Nessrine** et **Ikram** ;*

*★A Mes chères frères **Hossam** et **ramzi** ;*

*★A mon binôme **Noureddine** ;*

*★A Mes chers Amies, **Zaki**, **Sid Ahmed**, **Fares**, **Zako** ;*

*★A Mme **Bouinah Madiha** ;*

★A Tous mes enseignants depuis mes premières années d'études ;

★A Tous les membres de ma promotion ;

★A Tous ceux qui j'aime et ceux que m'aiment.

*★ **Faouzi***

Dédicaces

Je dédie ce travail :

★**A** *Mes chers **parents**, **Ms Djamel** , **Mme Nacira** , pour leurs soutient et leurs patience ;*

★**A** *Ma chère femme **Khouloud** ,pour leur soutient et leur patience, et mes chères enfants **Anes** et **Nazim** ;*

★**A** *mon chère frère **Abderaouf** ;*

★**A** *mes chères **khaled**, **Mohcine** ,**Hichem** et **Imen** ;*

★**A** *mon binôme **Faouzi** ;*

★**A** *Mme **Bouinah Madiha** ;*

★**A** *Tous mes enseignants depuis mes premières années d'études ;*

★**A** *Tous les membres de ma promotion ;*

★**A** *Tous ceux qui j'aime et ceux que m'aiment.*

★ ***Noureddine***

TABLE DES MATIÈRES

Table des matières	i
Liste des figures	v
Liste des figures	vii
Introduction générale	1
1 Les entrepôts de données passifs/actifs	3
1.1 Introduction	3
1.2 Définition	3
1.3 Histoire d'un système décisionnel	4
1.4 Fonctionnalités de système décisionnel	5
1.4.1 Phase d'alimentation (collecter)	5
1.4.2 Phase de stockage (dataWarehouse)	5
1.4.3 Phase de distribution	6
1.4.4 Phase de présentation (exploiter)	6
1.4.5 Administration	6
1.5 Objectifs d'un système décisionnel	6
1.6 Composants de base d'un système décisionnel	7
1.6.1 Sources de données	7
1.6.2 Entrepôt de données (data Warehouse)	8
1.6.2.1 Définition	8
1.6.2.2 Objectifs d'un entrepôt de données	8
1.6.2.3 Caractéristiques des entrepôts de données	9
1.6.2.3.1 Orientées sujet	9
1.6.2.3.2 Intégrées	9
1.6.2.3.3 Non volatiles	9
1.6.2.3.4 Historisées	9

1.6.2.3.5	Détaillées	10
1.6.2.4	Architecture d'un entrepôt de données	10
1.6.2.4.1	Sources de données :	11
1.6.2.4.2	Zone de préparation (ETL) :	11
1.6.2.4.3	Zone de stockage :	11
1.6.2.4.4	Zone de présentation (restitution) :	12
1.6.2.5	Différences entre l'entrepôt de données et magasin de données	12
1.6.2.6	Modélisation des données de l'entrepôt	13
1.6.2.7	Différents modèles de la modélisation dimensionnelle	13
1.6.2.7.1	Le modèle en étoile	13
1.6.2.7.2	Le modèle en flocon	14
1.6.3	Outils d'analyse	15
1.6.3.1	Serveur OLAP (On-line Analytical Processing)	15
1.6.3.1.1	Définition	15
1.6.3.1.2	Les 12 règles d'OLAP	15
1.6.3.1.3	Architecture d'OLAP	16
1.6.3.2	Data mining	16
1.6.3.3	Reporting	17
1.6.3.3.1	Définition	17
1.6.3.3.2	Le rôle du Reporting	17
1.7	Entrepôt de données actif (ADW : active data Warehouse)	17
1.7.1	Définition	17
1.7.2	Architecture d'un EDA	18
1.7.3	Règles d'analyses	19
1.7.3.1	Approche orientée objet	20
1.7.3.2	Approche multidimensionnelle	20
1.7.4	Analyse multidimensionnelle	21
1.7.5	Graphe d'analyse	21
1.7.6	Taxonomie d'événements dans un entrepôt de données actif	23
1.7.6.1	Évènement OLTP	23
1.7.6.2	Évènement OLAP	23
1.7.7	Comparaison entre l'entrepôt de données actif et passif	23
1.8	Conclusion	24
2	Automatisation de la conception et d'alimentation des entrepôts de données	25
2.1	Introduction	25
2.2	Définition	25
2.3	Objectifs de l'automatisation des DW	26
2.4	Processus d'automatisation	26

2.5	Les différentes approches d'automatisation	27
2.6	Outils d'automatisation disponibles	27
2.7	L'automatisation des entrepôts de données actifs	34
2.8	Conclusion	35
3	Analyse et Conception	36
3.1	Introduction	36
3.2	Architecture du système	37
3.3	Partie 01 : L'entrepôt de données passif (la partie passive de DW)	37
3.3.1	Le passage du modèle relationnel vers le modèle multidimensionnel	37
3.3.2	Règles pour la conception d'entrepôts de données à partir de données relationnelles	38
3.3.2.1	Règles pour les faits :	38
3.3.2.2	Règles pour les dimensions et les attributs	38
3.3.3	Algorithme de génération du modèle multidimensionnel à partir du modèle relationnel	39
3.3.4	<u>Algorithme</u> :	41
3.3.5	Exemple sur le déroulement de l'algorithme	41
3.4	partie 2 : les règles ECA : le module de génération des règles ECA	44
3.4.1	Introduction	44
3.4.2	L'idée	44
3.5	La conception de système ADADW :	45
3.5.1	Langage de modélisation UML	45
3.5.2	Processus de développement d'application :	46
3.5.3	Etude préliminaire	48
3.5.3.1	Identification des acteurs	48
3.5.3.2	Identification des messages	48
3.5.3.3	Modélisation en contexte de notre système ADADW « automatic design of active data warehouse »	49
3.5.3.4	Capture des besoins fonctionnels	49
3.5.3.4.1	Identification et description des cas d'utilisations	50
3.5.3.5	Capture des besoins non fonctionnels	52
3.5.3.6	Diagramme de cas d'utilisation globale du système	52
3.5.3.7	L'analyse	53
3.5.3.7.1	Diagramme de classe	53
3.5.3.7.2	Diagrammes de séquence des cas d'utilisation	55
3.6	Conclusion	58

4	Implémentation et réalisation	59
4.1	Introduction	59
4.2	Choix Technique	59
4.2.1	Langages de développement utilisés	59
4.2.1.1	Language JAVA	59
4.2.2	JavaFX	59
4.2.3	Environnement de développement	60
4.2.3.1	IntelliJ IDEA	60
4.2.3.2	PHPmyadmin	60
4.2.4	Outils logiciel	61
4.2.4.1	L'éditeur Draw.io	61
4.3	Présentation de l'application	61
4.3.1	Page d'authentification	61
4.3.2	Page d'accueil	62
4.3.2.1	Cube	62
4.3.2.2	Tables	63
4.3.2.3	attributes	64
4.3.2.4	Associations	64
4.3.2.5	Diagram	65
4.3.2.6	Add rules	66
4.4	Conclusion	67
	Conclusion Générale	68
	Bibliographie	69

TABLE DES FIGURES

1.1	Les 4 phases de la fonction du système décisionnel[1]	5
1.2	les composants de base des systèmes décisionnels[2]	7
1.3	Données Orientés sujet d'un entrepôt de données [3]	9
1.4	Non volatilité des données [3]	10
1.5	Données historiées	10
1.6	L'architecture d'un entrepôt de donnée [4].	11
1.7	Schéma en étoile [5].	14
1.8	Schéma en flocon de neige [5].	15
1.9	Architecture conceptuelle d'un entrepôt de données actif [6].	19
1.10	Graphe d'analyse [6].	22
2.1	ingrédients de l'automatisation de l'entrepôt de données[7].	28
2.2	l'outil ActiveBatch[8]	29
2.3	l'outil Zapbi ETL [8].	30
2.4	l'outil astra ETL [9].	31
2.5	l'outil WhereScape[8]	32
2.6	l'outil Bitwise QualiDI [9]	33
2.7	l'outil Bitwise QualiDI [8]	34
3.1	Architecture générale du système	37
3.2	Exemple d'un modèle d'une base de données des films.	42
3.3	le modèle de la base de données des films après appliqué l'algorithme	43
3.4	Le schéma flocon de neige pour l'exemple BD film	44
3.5	Le processus de développement en Y	47
3.6	le diagramme de contexte dynamique	49
3.7	Diagramme de cas d'utilisation de notre système [ADADW]	53
3.8	Diagramme de classe de système	54
3.9	Diagramme de séquence de cas d'utilisation authentifier	55
3.10	Diagramme de séquence de cas d'utilisation s'inscrire	55

3.11	Diagramme de séquence de cas d'utilisation consulter les règles	56
3.12	Diagramme de séquence de cas d'utilisation supprimer une règle	56
3.13	Diagramme de séquence de cas d'utilisation créer le cube	57
3.14	Diagramme de séquence de cas d'utilisation générer une règle ECA	58
4.1	Capture d'écran de l'interface authentification.	61
4.2	Capture d'écran de l'interface d'accueil.	62
4.3	Capture d'écran de l'interface crée cube.	63
4.4	Capture d'écran de l'interface ajoute classe.	63
4.5	Capture d'écran de l'interface ajoute attribut.	64
4.6	Capture d'écran de l'interface ajoute association.	65
4.7	Capture d'écran de l'interface schéma multidimensionnel.	66
4.8	Capture d'écran de l'interface ajoute règle.	66

LISTE DES TABLEAUX

1.1	Magasin de données et ED [5]	12
1.2	comparaison entre DW et ADW	24

INTRODUCTION GÉNÉRALE

la prise de décision est un problème essentiel qui préoccupe les gestionnaires des entreprises, L'actif principal d'une entreprise est l'information qu'elle détient. Défi d'aujourd'hui Il ne s'agit plus de rechercher des informations, il s'agit de savoir comment utiliser des quantités massives de données qui peuvent être utilisées pour extraire des informations utiles, permettant aux entreprises de créer valeur supplémentaire, et renforcer sa stratégie pour atteindre ses objectifs et s'adapter aux attentes ses clients.

Les nouveaux besoins de l'entreprise, la grande quantité de données générées par les systèmes opérationnels, et les technologies adaptées à leur mise en œuvre, ont conduit à l'émergence du concept de « data Warehouse » comme support des systèmes décisionnels.

Les entrepôts de données sont une solution aux besoins des entreprises. Ils ont été formalisés dans les années 90 par Inmon et Richard Hackathorn (1994). Leur finalité est de constituer une base de données orientée sujet, intégrée et contenant des informations historisées, non volatiles et destinées aux processus d'aide à la décision[10].

Les entrepôts de données actifs est une nouvelle catégorie de systèmes d'aide à la décision, qui offre la possibilité d'automatiser la prise de décision, Les entrepôts de données actifs étendent les entrepôts de données conventionnels avec des règles d'analyse qui imitent le travail d'un analyste pendant la prise de décision, Un entrepôt de données actif peut exporter les décisions vers les systèmes OLTP directement ou après consultation de l'utilisateur Ce faisant.

Problématique et objectifs du travail :

La conception et la construction d'un entrepôt de données est une tâche complexe et délicate. La méthodologie pour la conception d'un entrepôt se compose de plusieurs décisions majeures qui sont à la base d'une conception complète parmi elles nous pouvons citer : L'identification des tables de faits, le niveau de granularité de chaque table de faits et les dimensions appartenant à la table de faits.

L'objectif de notre travail est le développement d'un système d'aide à la conception des entrepôts de données actifs qui regroupe deux parties :

- Mise en œuvre d'une méthode de conception d'entrepôt de données passif fondée sur un algorithme de comparaison de clés étrangères : dans cette partie nous détaillons une approche pour transformer les modèles relationnels depuis les bases de données sources vers une organisation multidimensionnelle dans un cube de données.
- le développement d'un système de génération des règles ECA dynamique : dans cette deuxième partie nous développons un modèle de représentation conceptuel pour les règles ECA, ce modèle est associé à une base de données pour l'enregistrement et la génération des règles ECA.

Organisation du mémoire :

pour présenter notre travail , nous avons retenu pour ce mémoire une organisation en quatre chapitres.

- **Chapitre 1** : Dans le premier chapitre, nous avons abordé une étude sur les systèmes décisionnelles et les entrepôts de données actifs. Nous présentons les principes de la modélisation multidimensionnelle et nous distinguons les concepts d'entrepôt et de magasin de données . Nous détaillons aussi l'architecture des entrepôts de données actifs avec les règles ECA.
- **Chapitre 2** : Le deuxième chapitre concerne La conception automatisée des entrepôts de données. nous montrons que la conception des entrepôts de données et l'automatisation de la conception des DWs sont peu étudiées. Nous présentons également les solutions clés disponibles à l'heure actuelle dans les outils de conception pour créer un entrepôt de données.
- **Chapitre 3** : le troisième chapitre concerne la conception en détail de notre travail.
- **Chapitre 4** : le quatrième chapitre est consacré à l'implémentation et l'évaluation de notre application.

CHAPITRE 1

LES ENTREPÔTS DE DONNÉES PASSIFS/ACTIFS

1.1 Introduction

UN système décisionnelle désigne les moyens, les outils et les méthodes qui permettent d'offrir une aide à la décision et de permettre aux responsables de la stratégie d'entreprise d'avoir une vue d'ensemble de l'activité traitée, Parmi les outils décisionnels les systèmes d'entrepôts de données qui apparut pour la première fois en 1980, à fin des années 90 le système décisionnel est devenu basé sur l'utilisation des entrepôts de données.

Nous allons présenter dans ce chapitre la définition et l'histoire des systèmes décisionnels, ainsi que l'objectif d'un système décisionnel et leur composant de base, puis nous allons voir les Entrepôts de données actifs, d'une manière plus détaillée, Dans un premier temps, Nous allons présenter la définition d'un Entrepôt de données et leur architectur, en plus nous allons présenter les règles d'analyses, l'analyse multidimensionnelle et le graphe d'analyse. Nous finissons par une comparaison entre ED (Entrepôt de données) et EDA (Entrepôt de données actif).

1.2 Définition

Les systèmes décisionnels font partie d'une famille de système d'information qui souvent appelée « SID » (système d'information décisionnelle) en anglais Décision Support System ou «BI » pour Business Intelligence.

Parmi les nombreuses définitions du système décisionnel que nous avons étudiées, a été défini par l'un des pionniers dans ce domaine comme « une copie des données transactionnelles organisées spécialement pour l'interrogation et l'analyse »,Le SID

se définit progressivement comme un système composé de plusieurs modules[11].

Devisy a dit « Le SID est un système permettant aux décideurs de l'entreprise de disposer des informations pertinentes et des outils d'analyse puissants pour aider à prendre les bonnes décisions aux bons moments » [12].

Une autre définition donnée dans [13] « Un système d'information décisionnel, SID, est un système qui réalise la collecte, la transformation des données brutes issues de sources de données et le stockage dans d'autres espaces ainsi que la caractérisation des données résumées en vue de faciliter le processus de prise de décision ».

Un système de décision correspond donc à tous les moyens, outils et des méthodes permettant la collecte, l'intégration, la modélisation et la récupération des données internes, et Externes à l'entreprise, matériels ou non, pour apporter une aide à la décision Et permettre aux responsables de la stratégie d'avoir une vue d'ensemble de l'activité traitée.

1.3 Histoire d'un système décisionnel

Il est utile de faire un bref historique qui retrace les étapes qui ont marqués la marche du concept du système décisionnel [14].

En 1962, Ken Iverson publie le premier langage, multidimensionnel (APL : : A Programming langage) ce langage a été ciblé par les premiers outils multidimensionnels Demande de commercialisation.

Dix ans plus tard, un moteur OLAP conçu pour les applications financières voit le jour (Banques, assurances ...).

En 1993, E. Codd précise les règles générales de description des moteurs OLAP (modèles relation). Puis vinrent deux grands théoriciens, Ralph Kimball et Bill Inmon, qui ont bouleversé entièrement le monde de l'information en transformant les données en une information fiable et cohérente en entreprise.

Ce qui motive les grandes entreprises de traitement de données comme MICROSOFT, IBM à crée des outils d'analyse puissants pour les applications de données orientées métier. De plus, la concurrence les oblige à produire plus d'outils, de logiciels pour mettre en œuvre les indicateurs stratégiques de l'entreprise et la manière de les interpréter pour définir les prévisions et détecter les cibles ou les tendances.

En 2000, Microsoft a fait un véritable lifting de l'informatique décisionnelle et qui a voulu s'imposer sur le marché des SGBD face à Oracle et IBM, leader mondial de la Business Intelligence, en créant la version SQL server 2000. Ce qui a conduit à une grande concurrence entre les firmes et notamment entre MICROSOFT et IBM qui a fait progresser d'une façon spectaculaire le SD et ses applications [15].

1.4 Fonctionnalités de système décisionnel

Un système décisionnel remplit quatre fonctions fondamentales, à savoir la collecte, Intégration, diffusion et présentation des données. Ajouter à ces quatre fonctions La fonction de contrôle du système décisionnel lui-même, l'administration[16] :

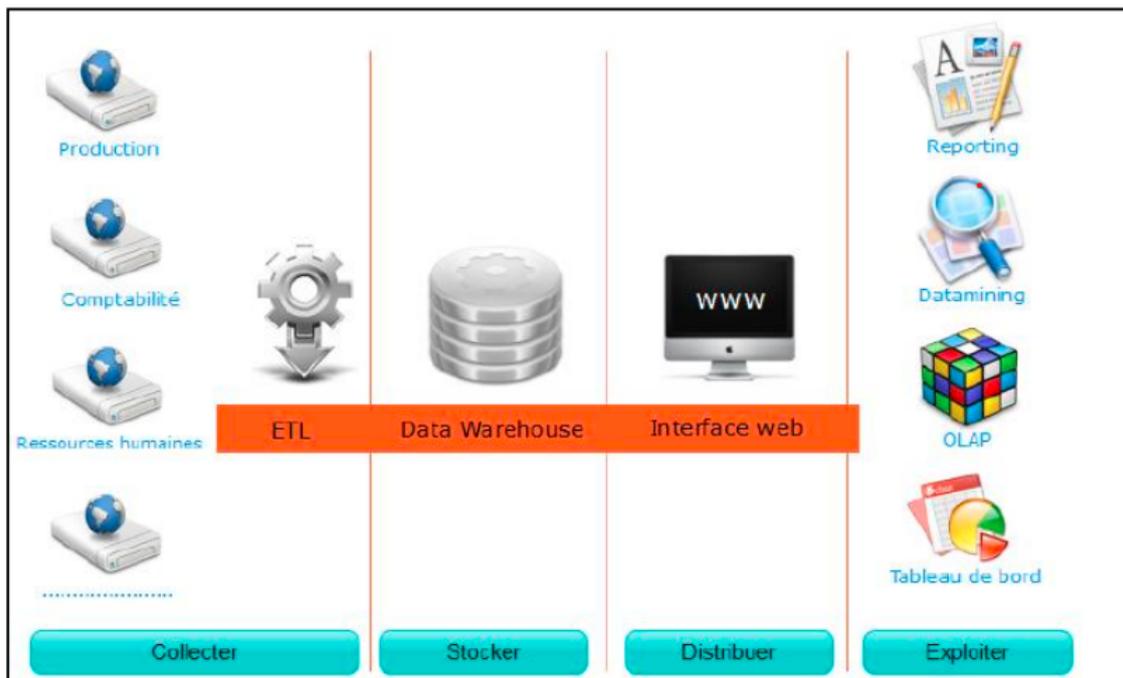


FIGURE 1.1 – Les 4 phases de la fonction du système décisionnel[1]

1.4.1 Phase d'alimentation (collecter)

La collecte est l'ensemble des tâches qui consistent à détecter, à sélectionner, à extraire et à filtrer les données issues de différentes sources de stockage, de les formater, nettoyer et consolider.

Les sources de données internes ou externes sont souvent hétérogènes Techniquement et sémantiquement, cette fonctionnalité est la plus difficile à mettre en œuvre dans des systèmes décisionnels complexes. Il est particulièrement dépendant des outils ETL[1].

1.4.2 Phase de stockage (dataWarehouse)

Cette étape permet de stocker les données sous une forme appropriée. Par ce que Les demandes de décision sont très coûteuses en ressources machine, les données doivent être Stockés dans des bases de données spécialisées, ? l'entrepôt de données ou datawarehouse. Le datawarehouse se charge de stocker et de centraliser les données en vue de la constitution du système d'information décisionnel[16].

1.4.3 Phase de distribution

Cette étape implique des outils de récupération de données afin de distribuer et Faciliter l'accessibilité de l'information en fonction de la fonction et du type d'utilisation.

Le portail décisionnel EIP (Entreprise Information Portal) remplit la fonction de distribution de l'information auprès de l'ensemble des partenaires internes de l'entreprise[16].

1.4.4 Phase de présentation (exploiter)

Cette quatrième fonction est la plus évidente pour l'utilisateur, régit les conditions d'accès de l'utilisateur aux informations. Elle assure le fonctionnement du poste de travail, le contrôle d'accès, la prise en charge des requêtes, afficher les résultats sous une forme ou autre. Il utilise également toutes les techniques de communication possibles comme les outils bureautiques, raquetteurs et générateurs d'états spécialisés, infrastructure web, télécommunications mobiles... etc[1].

1.4.5 Administration

C'est une fonction transversale qui supervise la bonne exécution de toutes les autres fonctions. Il guide le processus de mise à jour des données, la documentation sur les données et sur les métadonnées, la sécurité, les sauvegardes et finalement la gestion des incidents[1].

1.5 Objectifs d'un système décisionnel

Le système de prise de décision comporte comme objectifs [1] :

- **Anticiper les changements** : En étudiant les informations historiques et en faisant des prédictions, le SD aide à expliquer les performances actuelles et à prédire l'avenir, en utilisant des prédictions et des modèles vraisemblables dans l'univers possible.
- **L'accroissement de la force de la réactivité de l'entreprise** :Les systèmes décisionnels exigent la flexibilité d'une entreprise pour répondre avec souplesse aux exigences de l'évolution, dans un monde en perpétuel mouvement où les variables de succès sont beaucoup et ne cessent pas d'évoluer.
- **Offrir aux décideurs une vision transversale de l'entreprise, en intégrant toutes les dimensions.**
- **Assurer la protection et la sécurisation des informations** :Le système doit permettre le contrôle d'accès à ces informations confidentielles.

- **Conversion de la masse des données en une valeur métier** : Les systèmes via des outils d'analyse peuvent libérer de la valeur qui contribue aux opérations de prise de décision.

1.6 Composants de base d'un système décisionnel

Se compose d'outils informatiques et de progiciels pour assurer Le fonctionnement de la chaîne de traitement de l'information. Il a suggéré d'utiliser les données Grâce à des systèmes d'information, généralement des données de production, dans Informations pouvant être utilisées à des fins de prise de décision[1].

les composants d'un système décisionnel, représenté dans la figure 2, comprend trois composantes principales :

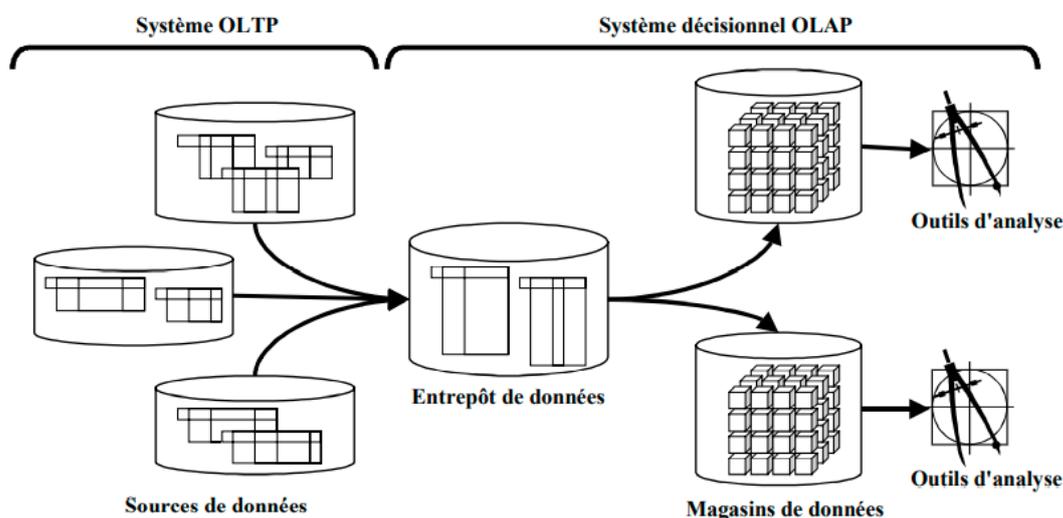


FIGURE 1.2 – les composants de base des systèmes décisionnels[2]

1.6.1 Sources de données

Il s'agit principalement de données internes à l'entreprise, mais dispersées au fur et à mesure qu'elles sont stockées dans les bases de données de production de différents services (legacy system).

Il peut également s'agir d'une source externe, récupérées via des services distants, des web services, par exemple :

Ce sont des données complexes : plusieurs technologies (types de fichiers, encodages, liens d'accès aux systèmes de gestion de bases de données SGBD), environnements (systèmes d'exploitation, matériels) et principes de sécurité pour les atteindre (mécanismes réseaux, authentifications) entrent en jeu pour les acquérir[16].

1.6.2 Entrepôt de données (data Warehouse)

1.6.2.1 Définition

Bill Inmon définit un entrepôt de données Dans son livre « Building the Data-warehouse » comme suit : « Une collection de données orientées sujet, intégrées, non volatiles et historique dont le but est d'aider à la prise de décisions managériales » [17].

1.6.2.2 Objectifs d'un entrepôt de données

- **Rendre l'information d'une organisation facilement accessible :** Le contenu de l'entrepôt doit être facile à comprendre et les utilisateurs finaux doivent pouvoir naviguer facilement et rapidement. En fait, les données sont considérées comme compréhensibles lorsqu'elles sont pertinentes et bien définies. Ils doivent également renvoyer les résultats des requêtes des utilisateurs dans un délai minimal[15].
- **Rendre les informations d'une entreprise cohérentes :** Les données de l'entrepôt doivent être crédibles, soigneusement assemblées à partir d'une variété de sources, nettoyées et libérées pour les urgences consommer. Les informations provenant d'une branche de l'entreprise peuvent être mises en corrélation avec celles d'une autre branche. Si deux unités de mesure portent le même nom, elles doivent alors signifier la même chose. Au lieu de cela, deux unités Cela ne signifie pas que la même chose doit être définie différemment. Une information cohérente est supposée être une information de haute qualité. [1].
- **Constituer une source d'information souple et adaptable :** Nous essayons simplement de ne pas éviter les changements dus à l'évolution des besoins des utilisateurs, à l'évolution des conditions commerciales et à l'évolution de la technologie (la conception de l'entrepôt de données a la perspective d'une modification permanente). L'arrivée de nouveaux problèmes ne doit pas perturber les données et applications ou technologies existantes [1].
- **présenter un bastion sécurisé qui protège la capitale information :**Un entrepôt de données contrôle non seulement l'accès aux données, mais aussi Offrir à ses responsables une bonne visibilité sur les usages et les données, même après avoir quitté l'entrepôt. Les informations Les plus précieuses d'une organisation sont conservées dans l'entrepôt de données, il doit donc efficacement contrôler l'accès à ces données confidentielles[1].
- **Constituer la base décisionnelle de l'entreprise :** Un entrepôt de données doit avoir les bonnes données pour appuyer les décisions, et il n'y a qu'un seul résultat attendu vraiment concret, et ce sont les décisions prises à la suite des données obtenues[1].

1.6.2.3 Caractéristiques des entrepôts de données

Les données d'un entrepôt de données possèdent les caractéristiques suivantes :

1.6.2.3.1 Orientées sujet Les données de l'entrepôt sont organisées par sujet et non par application. Par exemple, une chaîne de magasins d'alimentation organise les données de son entrepôt par rapport aux ventes qui ont été réalisées par produit et par magasin, au cours d'un certain temps[3].

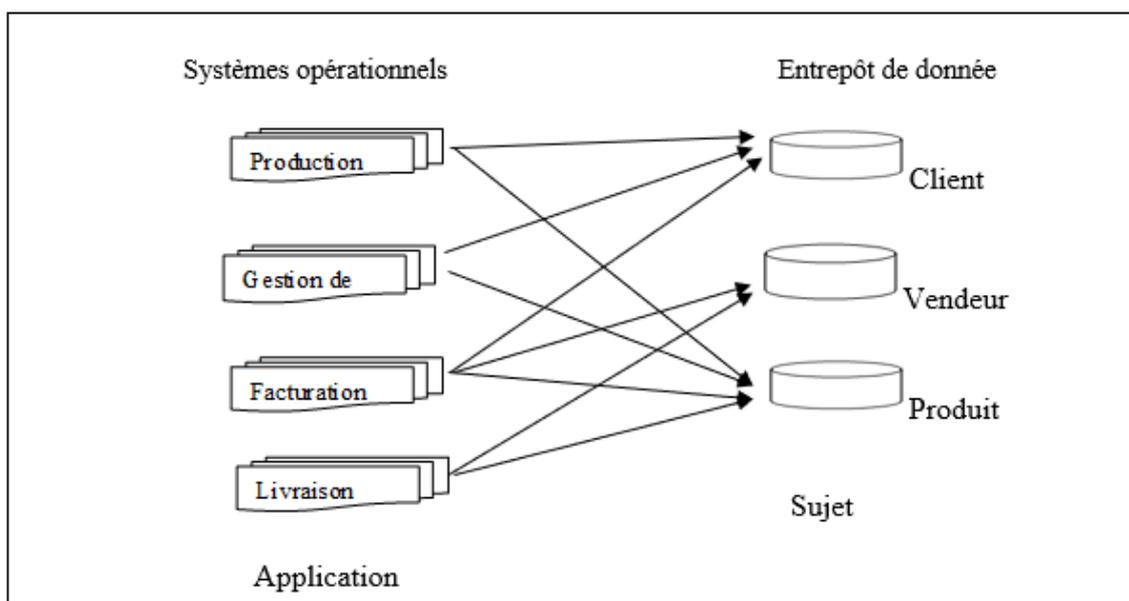


FIGURE 1.3 – Données Orientés sujet d'un entrepôt de données [3]

1.6.2.3.2 Intégrées Les données provenant de différentes sources doivent être consolidées avant d'être stockées dans un entrepôt de données. Intégration (par exemple, correspondance des formats), permettant la cohérence des informations[18].

1.6.2.3.3 Non volatiles Les données d'un entrepôt sont généralement utilisées en mode consultation. Ils peuvent être interrogés, mais ni modifiés ni supprimés., ni supprimés (sauf dans les cas de rafraichissement de l'entrepôt). Cela permet de garder Traçabilité des informations pour permettre une analyse à long terme période[3].

1.6.2.3.4 Historisées Les données de l'entrepôt doivent être associées à une référence temporelle afin que Reflète les activités de l'entreprise sur une période de temps. L'historique des données peut s'étendre sur plusieurs années. Ceci permet de suivre et d'analyser les variations des données et d'effectuer des analyses comparatives dans le temps[3].

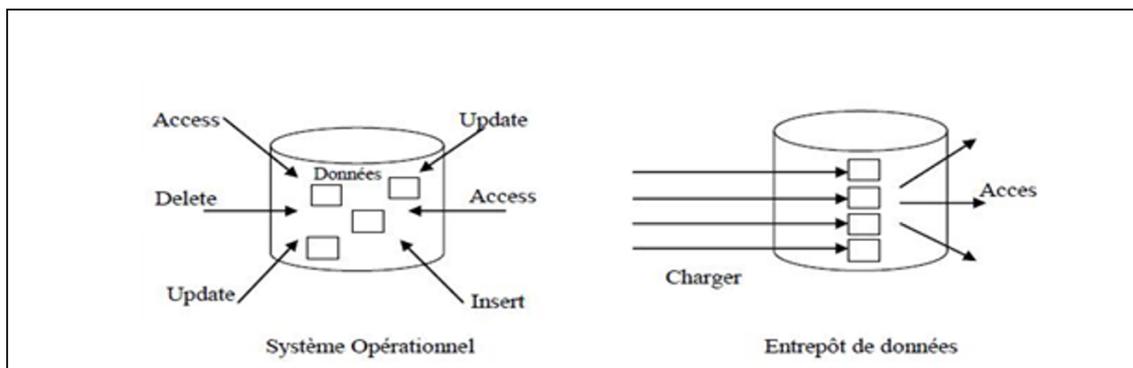


FIGURE 1.4 – Non volatilité des données [3]

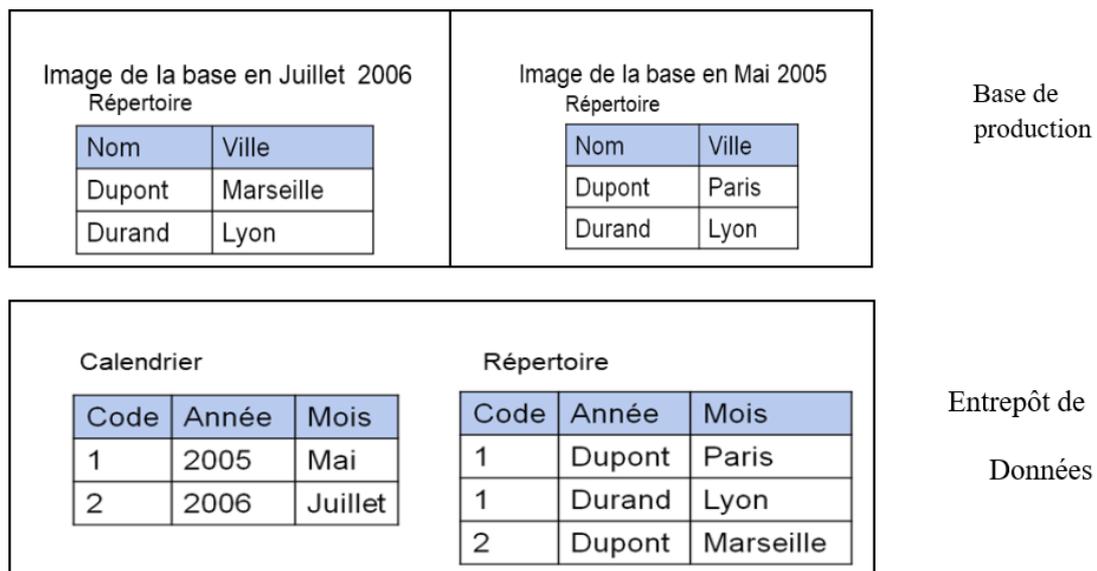


FIGURE 1.5 – Données historiées

1.6.2.3.5 Détaillées Issues des bases de données de production. Ils reflètent les événements les plus reçus. L'intégration régulière des données du système de production a lieu à ce niveau[19].

1.6.2.4 Architecture d'un entrepôt de données

Le processus de construction d'un ED peut être structuré en quatre axes, comme illustré dans la figure 6 :

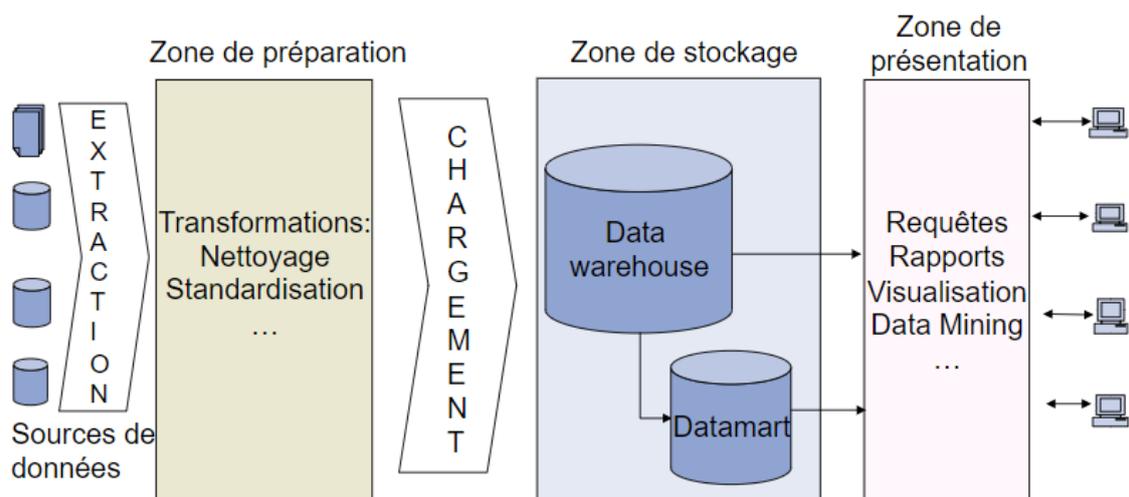


FIGURE 1.6 – L'architecture d'un entrepôt de donnée [4].

1.6.2.4.1 Sources de données : Un entrepôt de données stocke des données provenant de différentes sources d'informations hétérogènes et distribuées. Ces sources peuvent être des bases de données (oracle, MySQL . . .) des fichiers de données (xml, texte, Excel . . .), des sources externes à l'entreprise[3].

1.6.2.4.2 Zone de préparation (ETL) : Les données doivent passer par le processus d'Extraction, Transformation et chargement (ETL) pour qu'elles soient exploitables. Ce processus peut être résumé comme suit [20] :

Extraction (E) : Pendant la phase d'extraction, seules les données destinées à l'exploitation pour l'analyse qui sont gardées en connectant aux différentes applications ou bases en production.

Transformation (T) : Dans la phase de transformation, la mise au format des données, la fusion ou l'éclatement des informations et l'agrégation des données peuvent être effectuées.

Load (L) : Enfin, dans la phase de chargement, les informations sont stockées dans l'entrepôt de données.

1.6.2.4.3 Zone de stockage : Le lieu de stockage de données est l'entrepôt de données, Un ED peut comporter plusieurs magasins de données (data marts). Les magasins sont des extraits de l'entrepôt consacrés à un type d'utilisateurs et répondant à un besoin spécifique. Ils sont dédiés aux analyses décisionnelles de type OLAP[3].

- **Magasins de données (data marts) :** Est un sous-ensemble de l'entrepôt de données concernant un secteur particulier de l'entreprise. Il s'agit de données

extraites à une classe de décideurs ou à un usage particulier (recherche de corrélation, logiciel de statistiques, ...). L'organisation des données suit un modèle spécifique qui facilite les traitements décisionnels[2],[21].

1.6.2.4.4 Zone de présentation (restitution) : C'est tout ce que l'utilisateur peut manipuler par le biais des outils d'accès. Elle est constituée l'entrepôt de données ou d'une série de Datamarts[3].

1.6.2.5 Différences entre l'entrepôt de données et magasin de données

Le tableau suivant montre les différences entre l'entrepôt de données et les magasins de données à plusieurs égards importants.

Caractéristique	Magasin de données	Entrepôt de données(EDW)
Portée	Un domaine d'analyse	Plusieurs domaines d'analyse
Temps de développement	Mois	Années
Coûts de développement	\$ 10,000 à \$ 100, +000	\$ 1, 000,000 +
Complexité de développement	Faible à moyenne	Grande
Taille des données	Mb à plusieurs Gb	Gb jusqu'à plusieurs Pb
Horizon des données	Courantes et historiques	La plupart du temps historiques
Transformation des données	Faible à moyenne	Importante
Fréquence des mises à jour	Horaire, journaliser ou hebdomadaire	Peut aller jusqu'à mensuel
Nombre d'utilisateurs simultanés	Dizaines	Centaines à milliers
Types d'utilisateur	Analystes dans le domaine spécifique et gestionnaires	Analystes d'entreprise et cadres seniors
Objectifs d'affaires	Optimisation des activités dans inter-fonctionnelle le domaine spécifique	Optimisation et support à la décision

TABLE 1.1 – Magasin de données et ED [5]

1.6.2.6 Modélisation des données de l'entrepôt

Modélisation multidimensionnelle La représentation et le stockage des données dans un entrepôt de données doivent être conçus de manière à ce qu'il puisse non seulement gérer des requêtes multidimensionnelles complexes, mais aussi l'améliorer. La modélisation dimensionnelle est un système basé sur la recherche. Il prend en charge l'accès à un grand nombre de requête, cette requête qui caractérisé de divise les données en fait et dimension[5].

Fait Les faits représentent le sujet ou les thèmes analysés. Il représente le centre d'intérêt de l'entreprise et est considéré comme un concept clé sur lequel repose le processus décisionnel. Les faits consistent en des "mesures" ou attributs du fait (atomiques ou dérivés) qui correspondent aux informations liées au thème analysé. Par exemple pour la gestion des commandes, les mesures peuvent être la quantité du produit commandé et le montant de la commande[3].

Dimension Une dimension représente un contexte d'analyse d'un fait, tel que la gestion Les commandes peuvent être analysées en fonction des dimensions Client, Magasin, et Temps. Les dimensions sont affichées sous la forme d'une liste d'éléments organisés d'une certaine manière Structuré. Par exemple, pour la dimension temps, nous pouvons avoir la hiérarchie suivante : année, semestre, trimestre, mois, semaine et jour. Les dimensions sont caractérisées par des attributs de dimensions. Une hiérarchie est dite complète si tous les objets d'un niveau de la hiérarchie appartiennent à une seule classe d'objets d'un niveau supérieur et forment cette classe.[3].

1.6.2.7 Différents modèles de la modélisation dimensionnelle

1.6.2.7.1 Le modèle en étoile C'est un modèle sous forme d'étoile, avec une table de faits centrale et des dimensions comme branches, la table de fait identifie de dimension et il y'a absence totale de la notion d'hiérarchie[22].

Avantage

- Facilité de navigation.
- Requetes simples.

Inconvénients

- Alimentation complexe.
- Redondance dans les dimensions.
- Hiérarchies implicites.

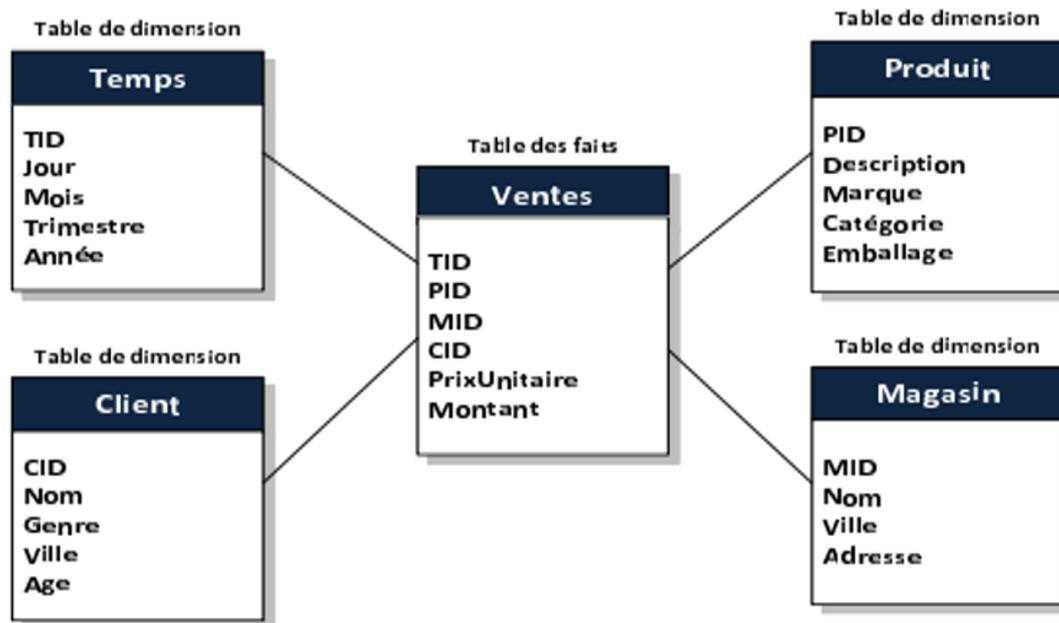


FIGURE 1.7 – Schéma en étoile [5].

1.6.2.7.2 Le model en flocon Le schéma en flocon de neige est une variante du schéma en étoile avec une normalisation des dimensions[23].

Avantage

- Hiérarchies explicites.
- Facilité d'alimentation.

Inconvénients

- Requêtes difficiles.
- Requêtes complexes.

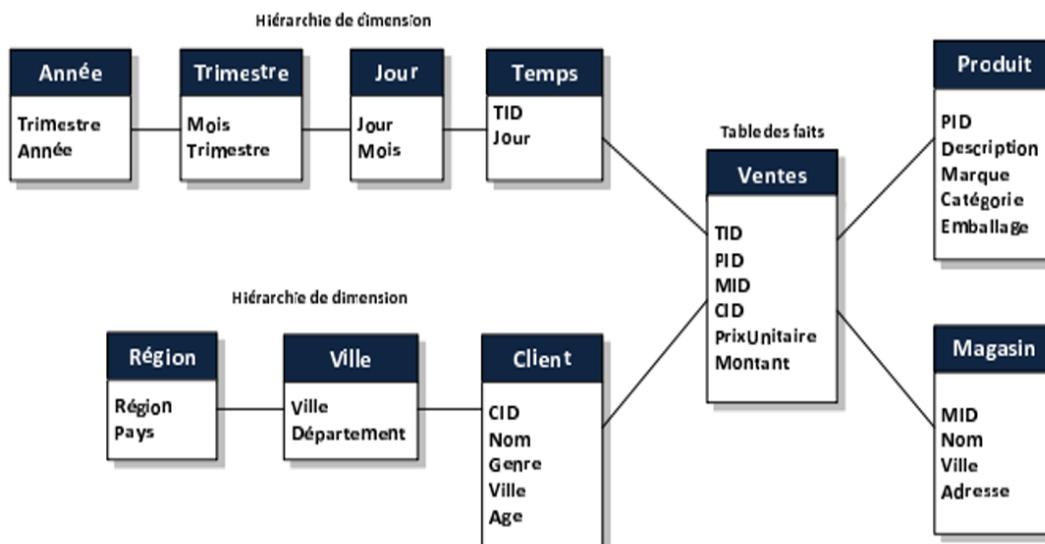


FIGURE 1.8 – Schéma en flocon de neige [5].

1.6.3 Outils d'analyse

Il existe divers outils d'aide à la décision sur le marché, comme les outils de fouille de données (data mining), outils d'analyse en ligne (pour la synthèse et l'analyse des données multidimensionnelles), outils d'interrogation (pour faciliter l'accès aux données en fournissant une interface conviviale au langage de requêtes)[23].

1.6.3.1 Serveur OLAP (On-line Analytical Processing)

1.6.3.1.1 Définition Un serveur OLAP est défini comme « une catégorie de logiciels axés sur l'exploration et l'analyse rapides des données selon une approche multidimensionnelle à plusieurs niveaux d'agrégation » [24].

Et selon (Kimball and M. Ross) « . . . une catégorie d'outils qui permet exploration interactive suivant une approche multidimensionnelle à plusieurs niveaux d'agrégation . . . » [25].

Le but d'OLAP est de distribuer les données aux utilisateurs sans les obliger à apprendre des complexes formules de programmation, d'interrogation ou même à ce qu'ils aient à programmer leurs tableaux[16].

1.6.3.1.2 Les 12 règles d'OLAP F. Codd (1993) définit 12 règles que tout système de pilotage multidimensionnel devrait respecter [26] :

1. **La structure multidimensionnelle :** le modèle OLAP est multidimensionnel.
2. **La transparence :** le serveur OLAP est transparent pour l'utilisateur.
3. **L'accessibilité :** l'utilisateur OLAP dispose de l'accessibilité à toutes les données nécessaires à ses analyses.

4. **La stabilité** : le système reste stable quel que soit le nombre de dimensions.
5. **Architecture client-serveur** : le serveur OLAP s'intègre dans une architecture client-serveur.
6. **Le dimensionnement** : le dimensionnement est générique pour assurer les analyses.
7. **La gestion complète** : le serveur OLAP assure la gestion des données claires.
8. **Les multiutilisateurs** : le serveur OLAP offre un support multiutilisateur (gestion des mises à jour, intégrité, sécurité).
9. **L'inter-dimension** : le serveur OLAP permet la réalisation d'opérations inter-dimensions sans restriction.
10. **L'aspect intuitif** : le serveur OLAP permet une manipulation intuitive des données.
11. **La flexibilité** : la souplesse de l'édition des rapports est intrinsèque au modèle.
12. **L'analyse sans limites** : le nombre de dimensions et de niveaux d'agrégation possibles est suffisant pour autoriser les analyses les plus poussées.

1.6.3.1.3 Architecture d'OLAP L'architecture OLAP se compose en trois services :

1. **Base de données** Il doit prendre en charge les données agrégées ou résumer de l'entrepôt de données. Données à structure multidimensionnelle (SGDB)[16].
2. **Serveur OLAP** Le serveur OLAP permet l'analyse des données par paradigme Multidimensionnel avec un temps de réponse optimisé. Le serveur OLAP fournit une représentation multidimensionnelle aux utilisateurs données sous la forme d'un ensemble d'hypercubes et implémenter un ensemble d'opérateurs OLAP (roll up, drill down, etc.) permet d'explorer ces hypercubes[16].
3. **Client OLAP** Cette couche définit une série d'interfaces utilisateur intuitives pour explorer Données interactives et multidimensionnelles. Ces interfaces peuvent déclencher Les opérateurs OLAP et présentent des informations à l'aide de différents types d'affichages interactifs : Tableaux croisés dynamiques, histogrammes et diagrammes statistiques[16].

1.6.3.2 Data mining

Le terme data mining couvre toutes les méthodes d'analyse non dirigées et partiellement automatisées qui visent à identifier les modèles, tendances et relations pertinentes dans la base de données. Les outils d'exploration de données reposent sur des méthodes statistiques et mathématiques ainsi que sur des techniques d'intelligence artificielle et d'apprentissage automatique de la machine.

La quantité de données que les entreprises génèrent, traitent et intègrent dans DWH à des fins d'analyse a augmenté de façon exponentielle. Le volume moyen des données dans le monde double tous les deux ans. Dans ce contexte, les méthodes d'exploration des données prennent une importance croissante dans le cadre de l'entreposage de données [22].

1.6.3.3 Reporting

1.6.3.3.1 Définition L'outil de rapport peut générer des rapports selon un format prédéterminé. Les bases de données sont interrogées selon les requêtes SQL préparées lors de l'élaboration du modèle. Les rapports peuvent ensuite être automatiquement distribués périodiquement sur l'intranet, ou ponctuellement à la demande.

Bien sûr, les outils de création de modèles de rapport offrent des fonctionnalités spécifiques Calculs et présentations (graphiques) pour des rapports spécialement conçus deviennent pertinents [16].

1.6.3.3.2 Le rôle du Reporting Le Reporting est probablement l'application la plus utilisée encore aujourd'hui de l'informatique décisionnelle, il permet aux : [16]

- Améliorer la prise de décision.
- Améliore l'efficacité de la gestion.
- Améliore la réactivité aux problèmes.
- Améliorer l'efficacité des ressources dans la prestation des services organisationnels.
- Gérer les processus de génération de rapports à l'aide de définitions de calendrier .
- Ajouter des rapports personnalisés au traitement du système de gestion des rapports.

1.7 Entrepôt de donnée actif (ADW : active data Warehouse)

1.7.1 Définition

Les entrepôts de données actifs appartiennent à une nouvelle catégorie de système d'aide à la décision, qui offre la possibilité d'automatiser la prise de décision pour les tâches [27] :

1. Tâches de décisions courantes.

2. Tâches de décisions semi-courantes.
3. Tâches de décisions non courantes.

Les tâches courantes et la partie courante des tâches semi-courantes peuvent être faites automatiquement dans l'entrepôt de données, et pour les tâches de décision non courantes devraient être faites par l'analyste.

L'idée d'un entrepôt de données actif a été inspirée par l'introduction des systèmes de bases de données actives, qui utilisent le paradigme ECA (événement-condition-action) pour automatiser les tâches récurrentes des processus d'affaires ("règles d'affaires") dans les bases de données OLTP (online transaction processing). Dans un entrepôt de données actif, ces règles sont appelées "règles d'analyse", par ce qu'elles imitent le travail d'un analyste pendant la prise de décision. Un entrepôt de données actif peut exporter des décisions aux systèmes OLTP directement ou après confirmation des utilisateurs [6].

1.7.2 Architecture d'un EDA

Pour automatiser les tâches décisionnelles qui se produisent fréquemment, l'architecture d'entrepôt de données traditionnelle peut être étendue en ajoutant les règles d'analyse.

L'idée de base derrière les règles d'analyse est d'étendre les règles générales de l'ECA pour les systèmes de base de données actifs avec des dispositifs des Entrepôts de données multidimensionnels (cubes, opérateurs OLAP) et méthodes pour la prise de décision.

Les règles d'analyse seront spécifiées une fois par l'analyste et seront ensuite exécutées automatiquement par l'entrepôt de données. En spécifiant une règle d'analyse, les connaissances tacites sur les procédures de prise de décision sont transformées en règles explicites, qui sont ensuite disponibles pour un grand nombre d'utilisateurs de l'entrepôt de données.

Par conséquent, les règles existantes peuvent être facilement adaptées à l'évolution des besoins et peuvent être utilisées comme modèles pour résoudre de nouveaux problèmes [28], [6].

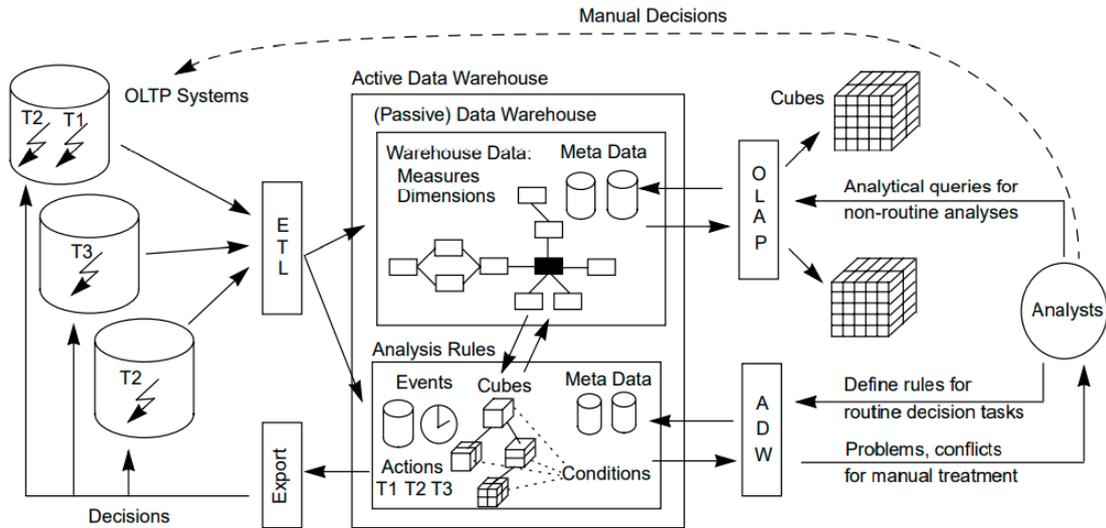


FIGURE 1.9 – Architecture conceptuelle d'un entrepôt de données actif [6].

1.7.3 Règles d'analyses

Les règles actives ECA, vues par Bailey et al., généralisent la notion de déclencheur. Elles utilisent généralement le formalisme Événement - Condition - Action [28].

Les règles d'analyse ont la même structure que les règles ECA, mais utilisent des opérateurs OLAP pour effectuer une analyse plus complexe sur les données d'entrepôt au lieu d'évaluer des conditions simples comme les règles ECA [6][28].

Le paradigme général ECA (événement-condition-action) des systèmes de bases de données actives :

ON event E

IF condition C

DO actin A

Selon Thalhammer [27] une règle d'analyse comprend les éléments suivants :

- **Dimension primaire** : Représente une instance d'un niveau de dimension spécifique pour lequel des règles sont définies. La dimension primaire détermine aussi la partie action de la règle d'analyse.
- **Événement** : spécifié par des points temporels pour le lancement de la règle d'analyse.
- **Condition primaire** : permet d'analyser uniquement le sous-ensemble d'instances de niveau qui appartiennent au niveau de la dimension primaire. Cette condition est booléenne et se rapporte aux attributs décrivant le niveau de la dimension primaire.
- **Graphe d'analyse** : représente tous les cubes qui sont utiles pour l'analyse, reliés entre eux par des opérateurs OLAP (RollUp, DrillDown, Intersection, union).

- **Etapas de décision** : représentent les conditions pour lesquelles une décision peut être prise.
- **Action** : représente la tâche de la décision prise par la règle effectuée sur le système OLTP.

La spécification des règles d'analyse nécessite de visualiser les données dans l'entrepôt de données dans une perspective orientée objet et multidimensionnelle[27].

1.7.3.1 Approche orientée objet

L'approche orientée objet permet d'identifier facilement les objets que les règles d'analyse doivent analyser et si des transactions doivent être effectuées pour ces objets dans la base de données OLTP. Pour référencer une transaction OLTP dans la partie action d'une règle, l'interface de transaction doit faire partie des métadonnées de l'entrepôt de données. Selon le paradigme orienté objet, ces transactions sont modélisées comme des méthodes du type d'objet OLTP et peuvent donc être invoquées sur des instances du type d'objet OLTP correspondant.

Alors La principale raison d'utiliser une approche orientée objet est qu'un objet à analyser à l'aide d'une règle active doit être associé à une méthode qui sera utilisée comme action pour la règle d'analyse[27].

1.7.3.2 Approche multidimensionnelle

En utilisant une approche multidimensionnelle, les analystes spécifient l'analyse multidimensionnelle requise pour la prise de décision. Un aspect important d'une approche multidimensionnelle est que les règles d'analyse modélisent avec précision la manière dont les analystes examinent les données multidimensionnelles "de haut en bas". Lors de l'analyse de données multidimensionnelles, les analystes créent plusieurs cubes à l'aide des opérateurs OLAP ROLLUP, DRILLDOWN, SLICE et INTERSECTION. Ces cubes décrivent des instances de niveaux de dimension auxquels des décisions doivent être prises. Lorsqu'il prend une décision, l'analyste examine d'abord le cube le plus grossier. Les instances d'un niveau où une décision peut être prise sont marquées comme "décidé" et seront retirées de l'analyse ultérieure. Les instances de niveau où une décision ne peut être prise seront analysées à l'aide d'un cube plus détaillé.

Ce modèle est utilisé pour les règles d'analyse dans la prise de décision. Les cubes requis pour analyser les instances de niveau du niveau de dimension principal seront spécifiés une fois par l'analyste. Ces cubes forment le graphique de la règle d'analyse pour chaque exécution de la règle d'analyse. Un cube de graphique d'analyse contient des cellules qui décrivent des instances de niveau du niveau de la dimension principale, sur lesquelles les règles d'analyse doivent être exécutées[27].

1.7.4 Analyse multidimensionnelle

Lorsqu'un analyste interroge un entrepôt de données pour décider d'exécuter des transactions dans un système OLTP, il adopte une approche descendante incrémentielle pour créer et analyser des cubes. Tout d'abord, l'analyste crée un cube très "grossier" qui décrit les objets pour lesquels des décisions doivent être prises (c'est-à-dire les instances de niveau de la dimension primaire).

Ensuite, en effectuant une analyse multidimensionnelle, les analystes effectuent des comparaisons, des agrégations, des transformations etc, entre Les cellules de ce cube.

En résumé, les analystes peuvent déterminer pour chaque instance de niveau d'une dimension principale si une décision peut être prise ou si une analyse plus poussée utilisant un cube plus détaillé est nécessaire.

Enfin, l'analyste crée un cube plus détaillé pour l'instance au niveau de la dimension primaire (nécessitant une analyse supplémentaire) puis poursuit son analyse. Il peut parfois y avoir plusieurs façons alternatives de créer et d'analyser des cubes plus détaillés. Dans ce cas, plusieurs cubes plus détaillés sont directement issus du même "point de départ".

Les règles d'analyse suivent la même approche. Pour automatiser la prise de décision, une règle d'analyse doit "connaître" quels cubes sont requis pour l'analyse multidimensionnelle. Ces cubes constituent le graphe d'analyse et sont spécifiés une seule fois par l'analyste. Les dimensions de chaque cube du graphe d'analyse sont classées en une "dimension primaire", qui représente les instances de niveau de la dimension primaire, et n-1 "dimensions d'analyse", qui représentent l'espace multidimensionnel pour l'analyse. Puisqu'une instance de niveau de la dimension primaire est décrite par une ou plusieurs cellules d'un cube, "l'analyse multidimensionnelle" comprend la comparaison, l'agrégation et la transformation les valeurs de mesure de ces cellules.

L'analyse multidimensionnelle effectuée sur le cube du graphe d'analyse est appelée étape de décision, Chaque étape de décision analyse les données d'exactly un cube du graphe d'analyse. Ainsi, le graphe d'analyse et les étapes de décision représentent la connaissance de l'analyse multidimensionnelle et la prise de décision d'une règle d'analyse[27].

1.7.5 Graphe d'analyse

Le graph d'analyse représente tous les cubes qui sont nécessaires à l'analyse multidimensionnelle, Ces cubes sont spécifiés par les analystes de la même manière descendante que les décisions manuelles, Nous distinguons les cubes locaux et les cubes globaux.

Les cubes qui composent le graphe d'analyse d'une règle d'analyse sont appelés

cube locaux car ils ne peuvent être utilisés que dans la règle d'analyse qui les spécifie, Le cube racine du graphe d'analyse est le cube le plus grossier, qui est nécessaire à la prise de décision d'une règle d'analyse. Ce cube peut être créé de bas en haut à partir d'un cube de base (en appliquant l'opérateur RollUp de OLAP sur un cube de base) ou de haut en bas à partir d'un cube global (en appliquant les opérateurs DrillDown, Slice ou Intersection sur les cubes globaux), qui représente des données pré-agrégées qui sont disponibles pour spécifier les cubes racines de plusieurs règles d'analyse [6][28].

- **L'opérateur Slice** : permet de sélectionner un sous ensemble de cellules dans un cube.
- **L'opérateur DrillDown** : permet de passer d'un niveau grossier de dimension à un niveau de dimension plus fin en présentant les mesures avec un niveau de granularité plus fin.
- **L'opérateur Intersection** : crée un cube plus spécifique qui représente un sous-ensemble de cellules, pour lesquelles la mesure possède un niveau de granularité plus fin.
- **L'opérateur RollUp** : permet de passer d'un niveau de dimension fine à un niveau de dimension plus gros.

Afin d'automatiser la prise de décision, les graphiques d'analyse sont une partie essentielle aux Règles d'analyse. Ces règles doivent connaître les cubes nécessaires à l'analyse multidimensionnel[6].

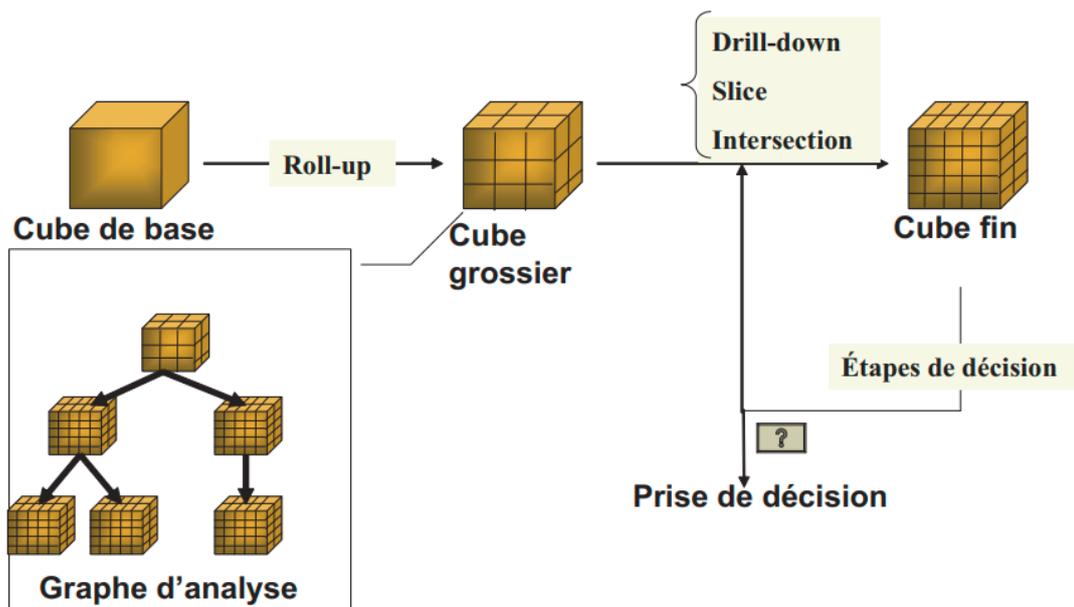


FIGURE 1.10 – Graphe d'analyse [6].

1.7.6 Taxonomie d'événements dans un entrepôt de données actif

On distingue deux catégories d'événements selon la classification de Bouattour, Les évènements OLTP et les évènements OLAP [29] :

1.7.6.1 Évènement OLTP

Les événements OLTP sont généralement des événements classiques rencontrés Dans une base de données transactionnelle, et qui concernent souvent la phase extraction du processus ETL(Extract-Transform-Load), ces événements peuvent concerner l'insertion, la modification et la suppression des données.

1.7.6.2 Évènement OLAP

Nous distinguons deux catégories d'événements OLAP [28] :

1. **Les événements résultats** : sont les événements qui permettent d'enrichir les schémas en étoile ou en constellation et diversifier ainsi les analyses.
2. **Les événements temporels** :qui Détermine l'intervalle d'observation auquel les données nécessaires à l'analyse sont collectées dans l'entrepôt. Les événements temporels sont utilisés pour déterminer quand les règles d'analyse doivent être exécutées. Il existe trois types d'événements temporels :
 - **Événements temporels absolus** : est une date du calendrier, exemple : 08 mai 2022.
 - **Événement temporel périodique** :peut-être la fin de chaque trimestre, chaque mois, etc.
 - **Événements temporels relatifs** : représentent des points temporels positionnés par rapport à un événement donné, exemple : 15 jours après le lancement d'un produit.

1.7.7 Comparaison entre l'entrepôt de données actif et passif

Les principales différences entre un entrepôt de données actif et passif selon Thalhammer et Bouattour resumer dans le tableau suivant :

Caractéristiques	L'entrepôt de données passif	L'entrepôt de données actif
Le but	Aide à la décision	La prise de décision
Le contenu	Mémoriser les données	Mémoriser les données et les connaissances
Analyse de données	Toutes les analyses de données sont faites explicitement par les analystes	Automatique pour les tâches de décision courantes et manuelle pour les tâches de décision non courantes

TABLE 1.2 – comparaison entre DW et ADW

1.8 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur les systèmes décisionnels, en effet, les fonctionnalités et leurs objectifs, en plus l'architecture des systèmes décisionnels. De même, les concepts de base de l'entrepôt de données (Data Warehouse), Nous avons également présenté les entrepôts de données actifs pour la prise de décision automatisée, Le chapitre suivant sera consacré pour l'automatisation de la conception de l'entrepôt de données.

CHAPITRE 2

AUTOMATISATION DE LA CONCEPTION ET D'ALIMENTATION DES ENTREPÔTS DE DONNÉES

2.1 Introduction

L'automatisation des entrepôts de données (DWA) remplace rapidement les approches conventionnelles pour créer des entrepôts de données, des référentiels de données centralisés utilisés par les entreprises pour obtenir des informations stratégiques basées sur les données.

Un entrepôt de données d'entreprise (EDW) est essentiel pour utiliser les données historiques à des fins de Business Intelligence et de reporting.

Cependant, les méthodes traditionnelles de compilation et de gestion de grandes quantités de données via le codage ETL manuel ne sont plus efficaces. Pour de telles exigences, le logiciel d'automatisation de l'entrepôt de données excelle dans la réduction du travail manuel impliqué dans la construction et le déploiement d'un entrepôt de données et la synthèse des données pour les rapports commerciaux[30].

2.2 Définition

Un entrepôt de données utilise une nouvelle technologie d'automatisation qui s'appuie sur des modèles et des processus de conception pour automatiser la planification, la modélisation et les différentes étapes d'intégration du cycle de vie complet d'un ensemble de données dans un entrepôt de données. Il offre une alternative efficace à la conception traditionnelle d'un entrepôt de données en réduisant les tâches chronophages telles que la génération et le déploiement de code ETL sur un serveur

de base de données[30].

2.3 Objectifs de l'automatisation des DW

- Augmentation du débit de l'entrepôt.
- Meilleure utilisation des ressources.
- Réduction des coûts de main-d'œuvre et d'exploitation.
- Amélioration du service client.
- Erreur humaine réduite.
- Améliore la fiabilité et augmente la vitesse de mise en œuvre des changements.
- Rendre les consommateurs de données heureux[8].

2.4 Processus d'automatisation

L'entreposage de données est un concept complexe qui implique plusieurs étapes différentes. La création de modèles de données efficaces et réutilisables et leur déploiement sont au cœur de tout bon entrepôt de données fonctionnel. Cependant, avant cette étape, il est essentiel de déterminer vos objectifs commerciaux et de les comprendre dans le cadre de vos processus et fonctions métier.

Une fois que cela a été fait, vous devez vous assurer que vous avez accès à toutes les sources de données à partir desquelles les données doivent être extraites et quelles transformations rendront ces données mieux adaptées à votre entrepôt de données.

Avec l'automatisation de l'entrepôt de données, vous pouvez supprimer l'exigence d'une intervention manuelle dans les étapes de planification, de modélisation et de déploiement pour accélérer l'analyse des données et la création de rapports. Avec des fonctionnalités et des capacités inhérentes pour tout, de l'amélioration de la qualité des données à l'orchestration des tâches, le logiciel d'automatisation de l'entrepôt de données permet aux entreprises de profiter pleinement des avantages d'un entrepôt de données sans passer des jours entiers à faire le travail acharné pour extraire des informations précieuses.

Les outils d'automatisation DWH modernes facilitent également l'intégration des données et les processus ETL à l'aide de connecteurs bidirectionnels intégrés vers des sources de données populaires, notamment des bases de données, des outils de visualisation et des plates-formes cloud. Cela élimine le besoin de répétitions cartographie des données et exécuter manuellement des projets et des processus.

Ces solutions d'entreposage de données intègrent également l'agilité et l'efficacité en permettant aux utilisateurs de créer rapidement un modèle dimensionnel qui incorpore tout, de Faits et dimensions aux clés primaires et de substitution, vous permettant de définir la structure de votre modèle tout en conservant l'intégrité.

Lors de la création de tels modèles dimensionnels, vous avez également la possibilité d'exploiter des dimensions à évolution lente, en vous assurant que seules les données pertinentes et à jour sont traitées pour l'analyse. En fin de compte, cela permet également aux entreprises de surfer sur la vague d'environnements commerciaux dynamiques, ce qui leur donne un avantage sur leurs concurrents[30].

2.5 Les différentes approches d'automatisation

Les deux types d'automatisation que l'on rencontre le plus fréquemment dans les environnements IT sont l'automatisation robotisée des processus (robotic process automation ou RPA) et l'automatisation de la charge de travail (workload automation ou WLA). La RPA est fondamentalement un logiciel programmé pour effectuer des tâches basiques imitant les comportements humains dans des applications. Généralement ce genre de solution est installé sur le poste de l'utilisateur qui la commande directement. Un logiciel de WLA sert à concevoir, programmer, surveiller et gérer des scripts et des exécutables dans l'entreprise. Installé sur les serveurs de l'entreprise, il est utilisé par les membres des équipes IT.

Ce sont deux approches complémentaires qui visent des scénarios très différents. Beaucoup d'entreprises sont en position de tirer profit des deux à la fois. Essentiellement, les solutions RPA et WLA libèrent du temps de travail humain qui peut alors être consacré à des tâches à plus forte valeur ajoutée. Si vous envisagez l'automatisation dans le cadre de la transformation digitale de votre entreprise, voici plusieurs points importants à prendre en compte[7].

2.6 Outils d'automatisation disponibles

Un logiciel d'automatisation d'entrepôt de données offre une approche fluide et sans code pour agréger et déplacer des données d'entreprise disparates des systèmes sources vers un entrepôt de données et au-delà. Contrairement à l'architecture d'entrepôt de données traditionnelle, le logiciel automatise l'exécution par lots et les exigences de déploiement de code ETL du processus d'entreposage de données. Construit sur des méthodologies agiles, certains des plus importants idées d'automatisation d'entrepôt de données utiliser une variété de fonctionnalités, notamment :

- Structures de données dénormalisées, normalisées et multidimensionnelles.
- Processus d'intégration de données ETL et ELT.
- Modélisation des données sources.
- Connectivité à divers fournisseurs de données

L'utilisation d'un outil d'automatisation des entrepôts de données peut améliorer la qualité et la précision des données. Cela permet aux entreprises d'accéder à des

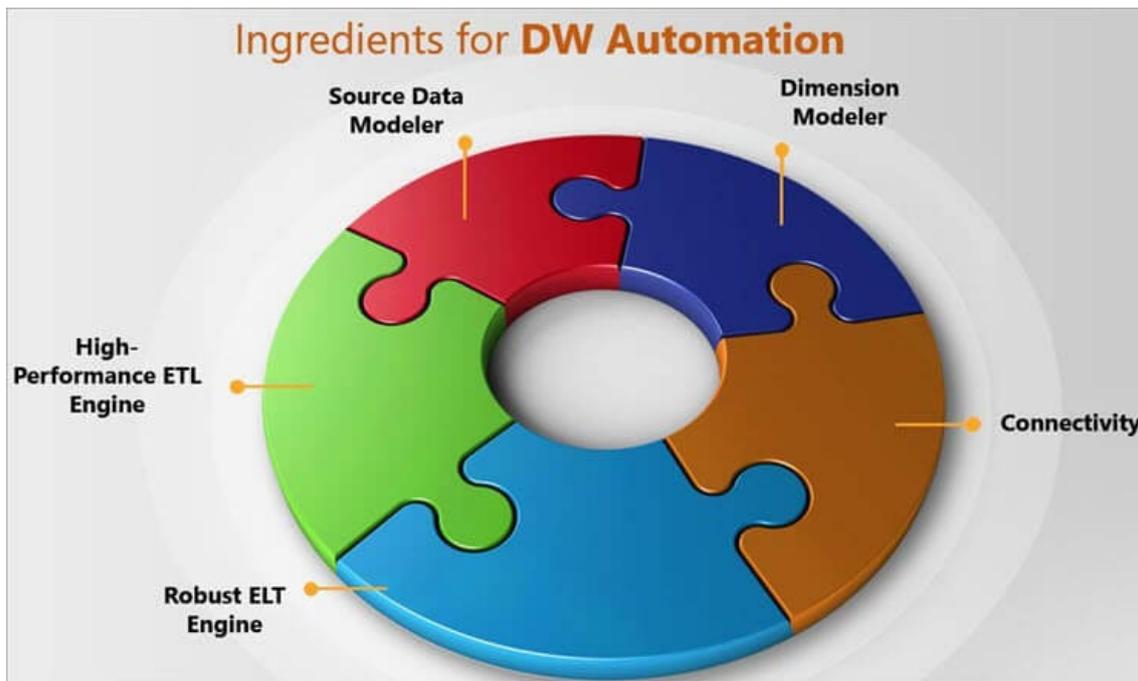


FIGURE 2.1 – ingrédients de l'automatisation de l'entrepôt de données[7].

données fiables et à des rapports et analyses avancés et précis. Il aide les entreprises à réagir rapidement aux conditions changeantes du marché. Cela augmente l'agilité de l'entreprise.

Ce sont des outils sans code que vous pouvez automatiser en deux fois moins de temps sans script et online. Il fournit des informations en temps réel et dispose de capacités d'alerte personnalisables[7].

Exemples des outils d'automatisation des entrepôts de données :

1. **ActiveBatch** :ActiveBatch est le meilleur pour l'optimisation de vos processus ETL.

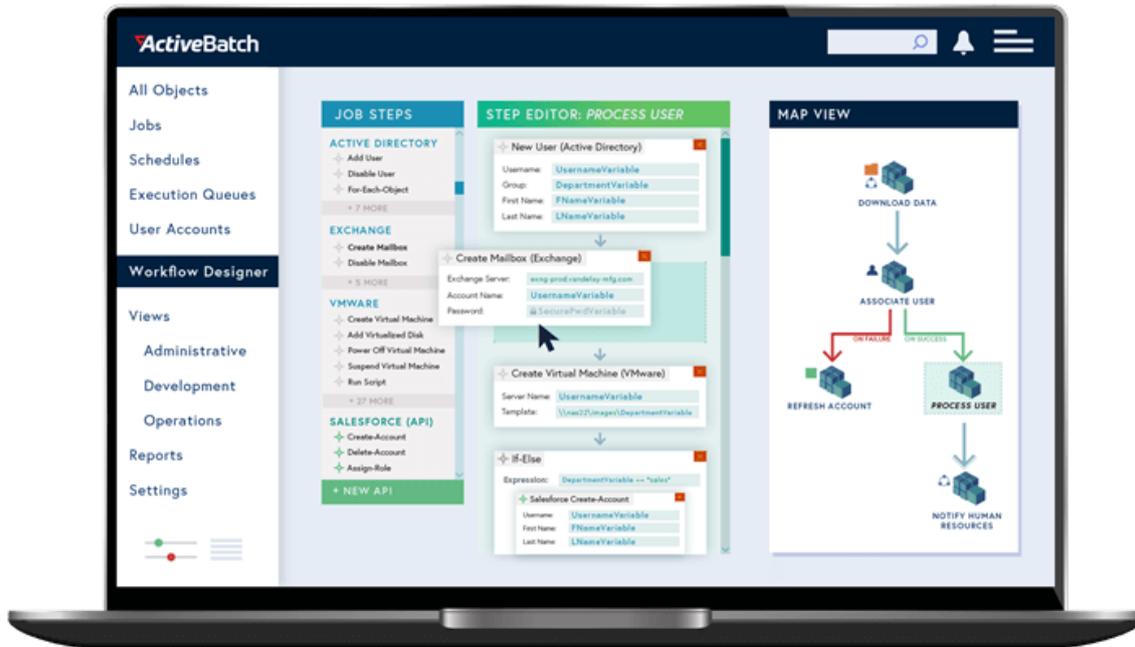


FIGURE 2.2 – l’outil ActiveBatch[8]

ActiveBatch Workload Automation contient les fonctionnalités Data Warehouse et ETL Automation qui vous aideront à optimiser vos processus ETL pour l’entreposage de données en temps réel.

Aussi, dispose d’une bibliothèque de travaux intégrée qui vous permettra de créer et d’automatiser des flux de travail fiables de bout en bout en deux fois moins de temps. Parmi ses caractéristiques [8] :

- fournit une bibliothèque de services pour tout intégrer virtuellement. Il fournit une accessibilité complète de l’API qui vous permettra de charger et d’exécuter des WSDL, des services Web SOAP, des services RESTful, etc.
- les fonctionnalités de planification avancées vous permettront de déclencher des processus d’entreposage de données et d’ETL en fonction de conditions externes. Il vous permettra de faire une planification basée sur des contraintes et une planification granulaire de date/heure.
- Ajoutez plusieurs points de contrôle dans les processus de l’entrepôt de données, de sorte que les étapes puissent être redémarrées sans affecter l’ensemble du processus par lots.
- empêcher tout accès non autorisé à l’aide d’autorisations granulaires, d’une authentification multifacteur et d’une gestion des accès privilégiés.

2. **Logiciel d’automatisation d’entrepôt de données Zapbi ETL** : Meilleur comme une solution facile à utiliser pour les données d’entreprise.

ZAP est le logiciel d’automatisation de l’entrepôt de données ETL compatible avec plusieurs systèmes et bases de données ERP, CRM et financiers.

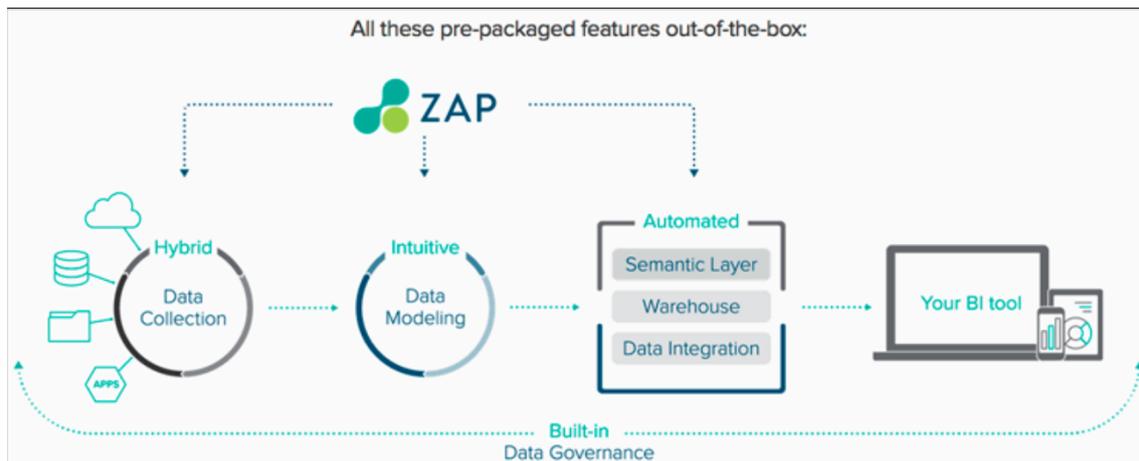


FIGURE 2.3 – l’outil Zapbi ETL [8].

Il peut être déployé, dans le cloud, sur site ou dans une combinaison hybride. ZAP Data Hub est le fournisseur de la gestion des données essentielles pour tous les utilisateurs de tous les logiciels de Business Intelligence et donne un accès sécurisé, efficace et précis à votre entrepôt de données.

Il collecte, intègre et prépare automatiquement les données pour les utilisateurs de BI via des fonctionnalités de collecte de données, d’intégration de données, de préparation de données et de gouvernance des données.

Parmi ces caractéristiques [8] :

- ZAP Data Hub possède une interface graphique sans code et suit une approche agile. Il fournit une automatisation basée sur des assistants.
- Il sera plus facile d’ajouter les sources de données même avec les besoins de reporting changeants.
- La plateforme est optimisée pour les bases de données Microsoft Dynamics, Salesforce, Sage, Oracle et SQL.
- ZAP Data Hub fournit les fonctionnalités de la collecte de données hybride.
- Il dispose de fonctionnalités de modélisation de données intuitives qui vous permettront de glisser- déposer des éléments.

3. **Logiciel Astera ETL** :Meilleur pour construire un écosystème de données intégré. Astera ETL Software fournit une solution pour créer un écosystème

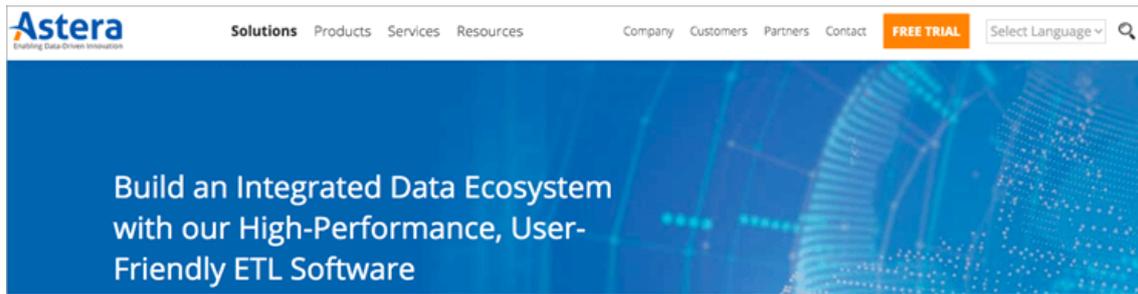


FIGURE 2.4 – l’outil astera ETL [9].

de données intégré. Il dispose de capacités d’intégration et de transformation de données pour des données de toutes complexités. Il prend en charge les données de divers formats, des fichiers hiérarchiques complexes et des documents structurés aux formats industriels tels que l’EDI et les données héritées.

Il fournit une connectivité transparente aux bases de données sur site, aux applications cloud et aux outils de visualisation. Il dispose de fonctionnalités pour écrire des données dans des outils de BI et de visualisation

Astera fournit des connecteurs intégrés pour PowerBI et Tableau qui faciliteront l’intégration et la transformation des données en insights. Il s’agit d’un logiciel ETL très performant et convivial.

Parmi ces caractéristiques [9] :

- Astera ETL Software dispose d’une interface sans code et conviviale.
 - Il a une architecture basée sur des clusters et accélère donc l’intégration et la transformation de grands ensembles de données en répartissant les tâches sur plusieurs nœuds.
 - Il prend en charge tous les formats de données.
 - Il possède une vaste bibliothèque de connecteurs.
 - Vous pourrez facilement intégrer, analyser et créer des fichiers XML, JSON et EDI complexes.
4. **WhereScape** :Meilleur pour automatisation des conceptions et des projets accélérés.

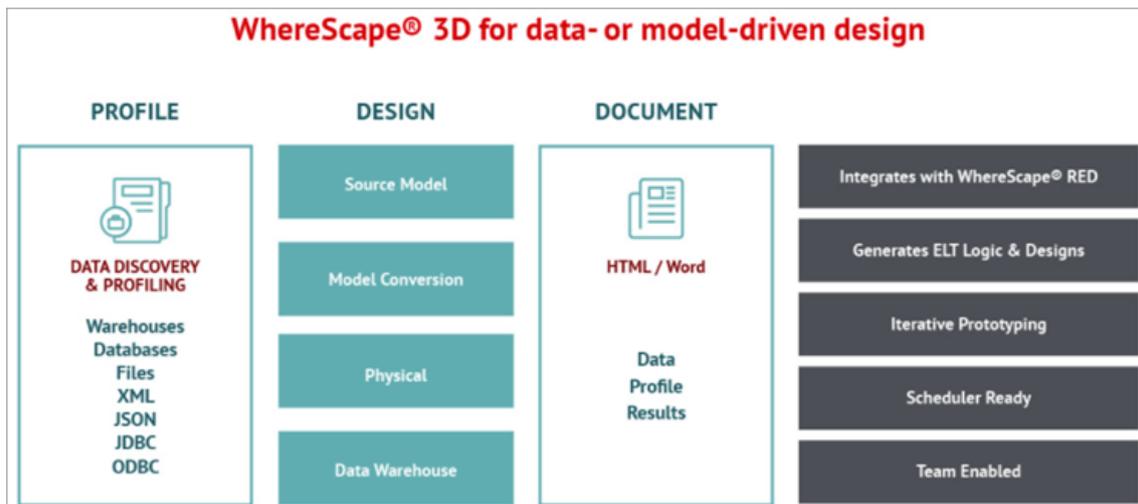


FIGURE 2.5 – l’outil WhereScape[8]

WhereScape propose le produit WhereScape 3D pour planifier, modéliser et concevoir tout type de projets d’infrastructure de données. Il dispose de capacités de découverte et de profilage de données.

WhereScape Automation est la plate-forme pour concevoir, développer, déployer et exploiter. C’est l’outil pour les plates-formes de données sur site et cloud. Il intègre les méthodologies dimensionnelles, 3NF et Data Vault 2.0.

Parmi ces caractéristiques [8] :

- WhereScape a des fonctionnalités pour Amazon Redshift, Apache Kafka, Exasol, Microsoft SQL Server, Microsoft Azure, Oracle, Snowflake, Teradata, etc.
 - Il fournit une automatisation pour Snowflake qui combine les fonctions natives de Snowflake, les assistants et les meilleures pratiques.
 - WhereScape Automation for Teradata a des capacités de Teradata qui minimiseront la complexité du développement et vous aideront à livrer plus rapidement les projets d’infrastructure Teradata.
5. **QualiDI au niveau du bit** :Meilleur pour test centralisé d’un ou plusieurs outils ETL.

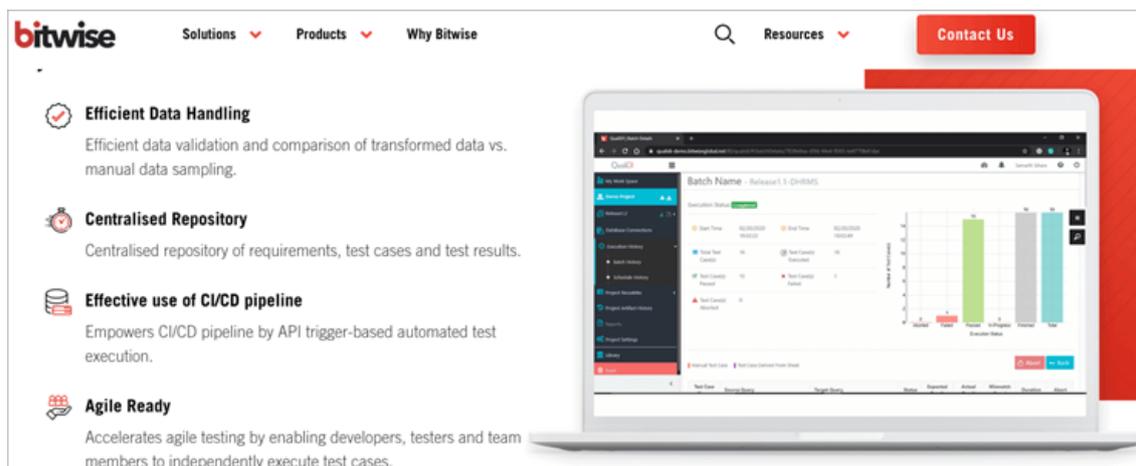


FIGURE 2.6 – l’outil Bitwise QualiDI [9]

Bitwise QualiDI est un outil de test ETL. Il centralise les tests d’un ou plusieurs outils ETL. Il effectue la validation des données tout en transformant des ensembles de données de plusieurs systèmes opérationnels en un entrepôt de données. Il peut être utilisé pour automatiser les tests ETL de n’importe quelle source vers n’importe quelle cible.

Il peut gérer le cycle de test ETL complexe. Bitwise QualiDI garantira la qualité tout au long du cycle de vie. Il dispose d’un système de gestion de version intégré pour les exigences et les cas de test.

Parmi ces caractéristiques[9] :

- QualiDI fournit des fonctionnalités avancées de test de Big Data, d’intégration JIRA, de capacité de planification intégrée, de prise en charge de l’intégration continue et de règles de transformation complexes, de la génération de données de test logique et d’une architecture évolutive selon les besoins.
- Pour les tests, il fournit les fonctionnalités de création de tests automatisés, de comparaison automatisée des données, de planification des tests, de validation des métadonnées, etc.
- Il prend en charge un ensemble hétérogène de magasins de données.
- Il fournit des fonctionnalités d’administration, de reporting et de suivi.

6. **Qlik Compose** :Meilleur pour raffinement automatisé et continu.

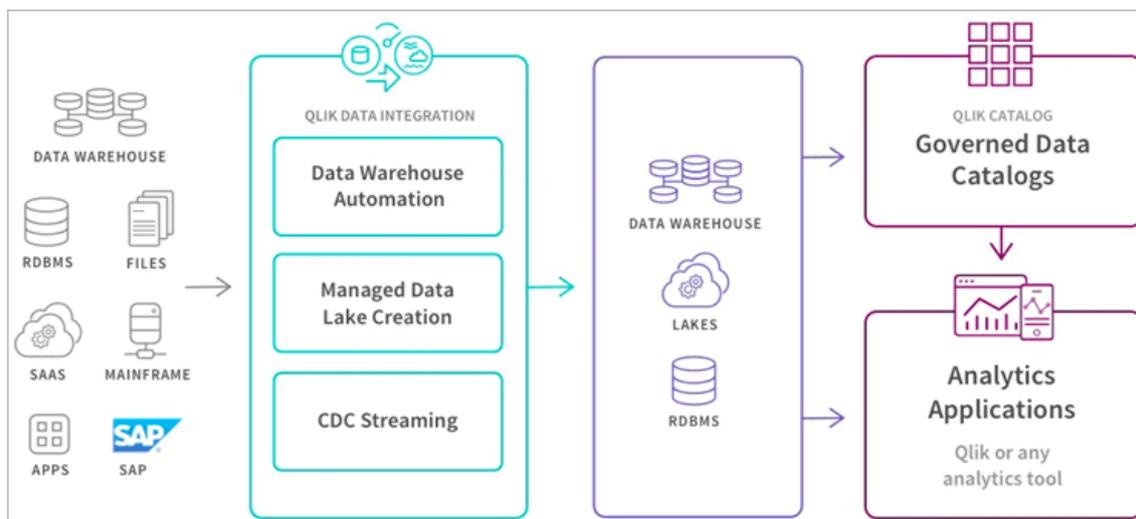


FIGURE 2.7 – l’outil Bitwise QualiDI [8]

Qlik Compose était auparavant connu sous le nom d’Attunity Compose. Il responsabilise les architectes de données et les équipes d’entrepôt de données. Il contient des fonctionnalités pour la conception, la construction et l’exploitation de l’entrepôt de données d’entreprise. Vous pourrez générer des commandes ETL sans codage manuel.

Prmi ces caractéristiques [8] :

- Qlik Compose dispose de fonctionnalités permettant de lancer un nouvel entrepôt de données et de nouveaux data marts, sur site et dans le cloud.
- Au fur et à mesure que les exigences métier évoluent rapidement, vous pourrez mettre à jour les modèles d’entrepôt de données et les nouvelles sources de données.
- Il possède des fonctionnalités qui vous permettront d’exécuter des travaux ETL selon les délais ou à la demande. Vous pouvez surveiller ces travaux en temps .

2.7 L’automatisation des entrepôts de données actifs

Il existe un certain nombre de travaux sur les entrepôts de données actifs. Selon les travaux de (Thalhammer et al., 2001), les entrepôts de données actifs appartient à une nouvelle catégorie de systèmes d’aide à la décision, (inspirées par les base de données actives qui utilisent le paradigme ECA) qui offre la possibilité d’automatiser la prise de décision, ensuite, les décisions sont automatiquement exportées vers les systèmes de traitement des transactions en ligne (OLTP).

Les règles ECA (Event-Condition-Action) dans un entrepôt de données actif sont imitent le travail d’un analyste lors de la prise de décision donc en l’appelées

les règles d'analyse[31].

2.8 Conclusion

Ce chapitre présente l'automatisation de la création des entrepôts de données passifs /actifs, mais en conclusion aucun des travaux existants ne fournit une solution complète pour la conception des entrepôts de données actifs qui tient compte des deux aspects : conception automatisée des entrepôts et automatisation de la génération des règles ECA.

CHAPITRE 3

ANALYSE ET CONCEPTION

3.1 Introduction

L'automatisation de la conception de l'entrepôt de données actif est une tâche complexe, qui comprend deux parties. La première partie traite des entrepôts de données passifs visant à automatiser la conception des entrepôts de données, et la deuxième partie concerne les règles ECA qui visent à automatiser la prise de décision.

Dans Ce chapitre, nous allons montrer la présentation et la description globale de notre système ADADW « automatic design of active data warehouse », ensuite nous mettons en évidence le côté conceptuel de notre application.

Dans la première partie de ce chapitre, nous décrivons l'idée de base de passage d'un modèle relationnel vers le modèle multidimensionnel, ensuite nous allons présenter l'algorithme de génération du modèle multidimensionnel à partir du modèle relationnel avec un exemple sur le déroulement de l'algorithme, la deuxième partie consiste à présenter l'idée de base de la conception des règles ECA, finalement, nous décrivons les composantes principales de la conception de l'application.

Lors de la conception de l'application, nous serons guidés par le processus 2 TUP (2 track unified process), que nous allons représenter à l'aide de diagrammes UML. Dans un premier temps, nous allons essayer d'exprimer les besoins fonctionnels et techniques sous forme de diagrammes de cas d'utilisation et construire les diagrammes de séquence pour détailler chaque cas d'utilisation.

A la fin nous allons donner le diagramme de classes d'application et le modèle relationnel résultat.

3.2 Architecture du système

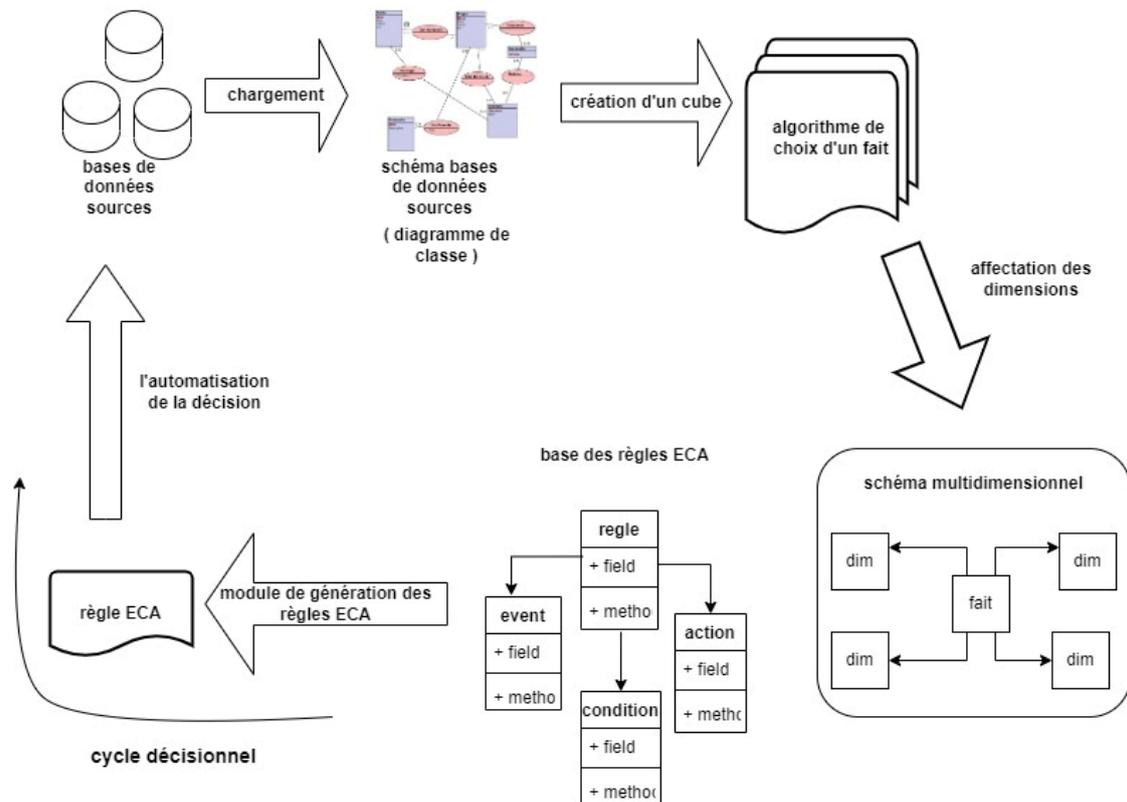


FIGURE 3.1 – Architecture générale du système

3.3 Partie 01 : L'entrepôt de données passif (la partie passive de DW)

3.3.1 Le passage du modèle relationnel vers le modèle multidimensionnel

Cette méthode est principalement basée sur une liste de règles pour identifier les différents éléments du schéma multidimensionnel, et consiste en deux étapes. La première vise à identifier la table de fait en calculant le nombre de clés étrangères dans chaque table du modèle relationnel, et la seconde permet d'identifier les dimensions, qui sont directement liées à la table de fait.

3.3.2 Règles pour la conception d'entrepôts de données à partir de données relationnelles

Cette section présente l'ensemble des règles qui sont développées par Nawfal El Moukhi et kram El Azami et Abdelaaziz Mouloudi et Abdelali Elmounadi, à partir de travaux précédents (Khouri et al., 2014) (Elmoukhi et al., 2015)(Khnaisser et al., 2015)(Santos et al., 2016) (Dahlan et Wibowo, 2016). Ces règles constitueront la base pour normaliser la conception des entrepôts de données[32].

3.3.2.1 Règles pour les faits :

- Les tables de faits sont les concepts d'intérêt principal pour le processus de prise de décision. Ils correspondent à des événements qui se produisent toujours dans l'organisation ou l'entreprise. (Chandwani and Uppal, 2015)
- Une table de faits représente toujours une activité particulière et doit être interrogée à partir d'un contexte particulier (une ou quelques dimensions).
- Aucune ligne du tableau des faits ne peut contenir une valeur vide.
- Une table de faits contient uniquement les clés étrangères qui représentent les clés primaires des dimensions et ces clés doivent être numériques, afin de garantir que la table de faits soit plus efficace. (Rudra and Nimmagadda, 2005).
- Chaque combinaison de valeurs de dimension définit une instance de la table de faits, qui est caractérisée par une et une seule valeur pour chaque mesure.
- Les mesures des tables de faits doivent être numériques et additives (au pire semi-additives). (Akbar et al., 2013).
- Les données d'une table de faits sont fixes et ne peuvent pas être modifiées. (Bliujute et al., 1998).

3.3.2.2 Règles pour les dimensions et les attributs

- Les dimensions déterminent comment les instances de faits peuvent être agrégées de manière significative pour le processus de prise de décision.
- Une table de faits doit toujours contenir la dimension temps.
- Les dimensions doivent avoir des clés primaires numériques.
- La clé primaire de chaque table de dimension doit être unique, et les champs doivent avoir une valeur non composée.
- Les hiérarchies des dimensions doivent de préférence avoir une forme simple de type 1-n et éviter les relations de type n-n.
- Un attribut non dimensionnel contient des informations supplémentaires sur un attribut de la hiérarchie, et est lié par une relation de type to-one (Golfarelli et al., 1998) .

- Les attributs non dimensionnels ne peuvent pas être utilisés pour l’agrégation (Golfarelli et al., 1998).
- La relation entre une table de faits et une dimension est toujours many-to-one (Cavalheiro et Carreira, 2016).

3.3.3 Algorithme de génération du modèle multidimensionnel à partir du modèle relationnel

l’idée de l’algorithme est inspiré de[32] :

— **Les Entrées :**

BD source : la base de données source.

Fichier xml : le fichier xml de la base de données source.

Tableau_t :String : nom de la table que nous cherche à calculer leur nombre clés étrangères.

T1, T2, . . . , Ti : liste des tableaux, chaque tableau contient des attributs {primary key, foreign key,..}.

— **Les Sorties :**

Liste P : liste des tableaux avec le nombre de clé étranger du chaque tableau.

Fait : le nom de la table qui contient le plus nombre des clés étrangers.

— **La déclaration des structures des données :**

L’enregistrement tableauI

tableauI :enregistrement

nom_tableau :String;

nbr_clé :int;

Fin d’enregistrement

La déclaration de la liste

Type Liste = ^Element;

Type Element :enregistrement

Valeur : tableauI;

Suivant : liste;

Fin d’enregistrement

— **La procédure d'insertion dans la liste****Procédure** inserer_element(l : liste, val : valeur)

var P : liste

Début

nouveau(P);

P^.valeur ← val;

P^.suivant ← l;

l ← P;

fin— **La procédure de comparaison****Procédure** comparaison (l : liste)

var Max : Element;

Fait : String;

Début

Element_courant := l.tete;

Max := Element_courant;

Element_courant := Element_courant.suivant;

while (Element_courant ≠ NULL) **DO****if** (Max.nbr_clé < Element_courant.nbr_clé) **Then**

Max := Element_courant;

End

Element_courant := Element_courant.suivant;

End

Fait := Max.nom_tableau;

fin

3.3.4 Algorithme :

Algorithm 1: calculer le nombre de clé étranger

```
1 Déclaration;
2 count :int ;
3 TI :tableauI ;
4 nom_tab :String ;
5 Liste p : liste ;
6 DEBUT ALGORITHMME;
7 count←0 ;
8 for chaque tableau ∈ BDsource do
9   | nom_tab :=tableauT.nom ;
10  | if tableauT.attribut.type is "FK" then
11    | count←count+1;
12  | end
13  | TI.nom_tableau←nom_tab;
14  | TI.nbr_clé←count ;
15  | Inserer_element(listep, TI);
16 end
17 comparaison(liste p);
18 FIN ALGORITHMME;
19
```

3.3.5 Exemple sur le déroulement de l'algorithme

Nous prenons comme exemple le modèle ci-dessous. Il représente un modèle d'une base de données des films, Le modèle est composé d'une table film liée à quatre tables :

- la table film contenant id_film, le titre, l'année, le genre et le résumer.
- La table des acteurs qui contient les informations sur les acteurs.
- La table de producteur qui contient les informations sur le producteur.
- la table ville contient les villes du provenir le film, la table ville est liée à la table pays.
- la table salle où se présenter le film.

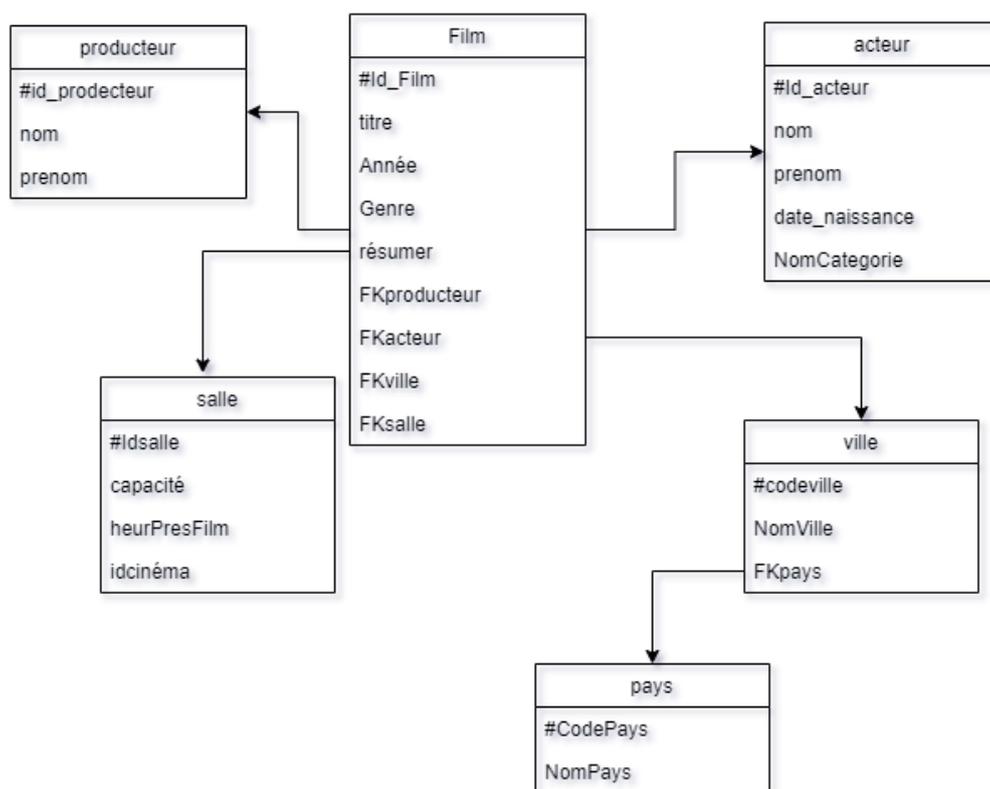


FIGURE 3.2 – Exemple d’un modèle d’une base de données des films.

En suivant l’algorithme et les règles présentées en précédent, la table de faits sera la table film, Il s’agit de la table qui représente une activité particulière de l’entreprise et le principal intérêt pour les décideurs. Cette table contient le plus grand nombre de relations many-to-one (le plus grand nombre de clés étrangères), qui est le type privilégié pour les relations entre les tables de faits et les tables de dimensions.

En ce qui concerne les tables de dimension, il s’agira respectivement de la table acteur, de la table producteur, de la table sale, et de la table ville.

La table ville est liée à une table représentant le lieu, et elles peuvent donc toutes être regroupées dans une seule table (Dim_Place), Enfin, notre exemple ne contient pas de tableau date, il est donc nécessaire d’en ajouter un qui représentera la dimension temporelle.

Le modèle multidimensionnel ci-dessous est le résultat final de cette transformation :

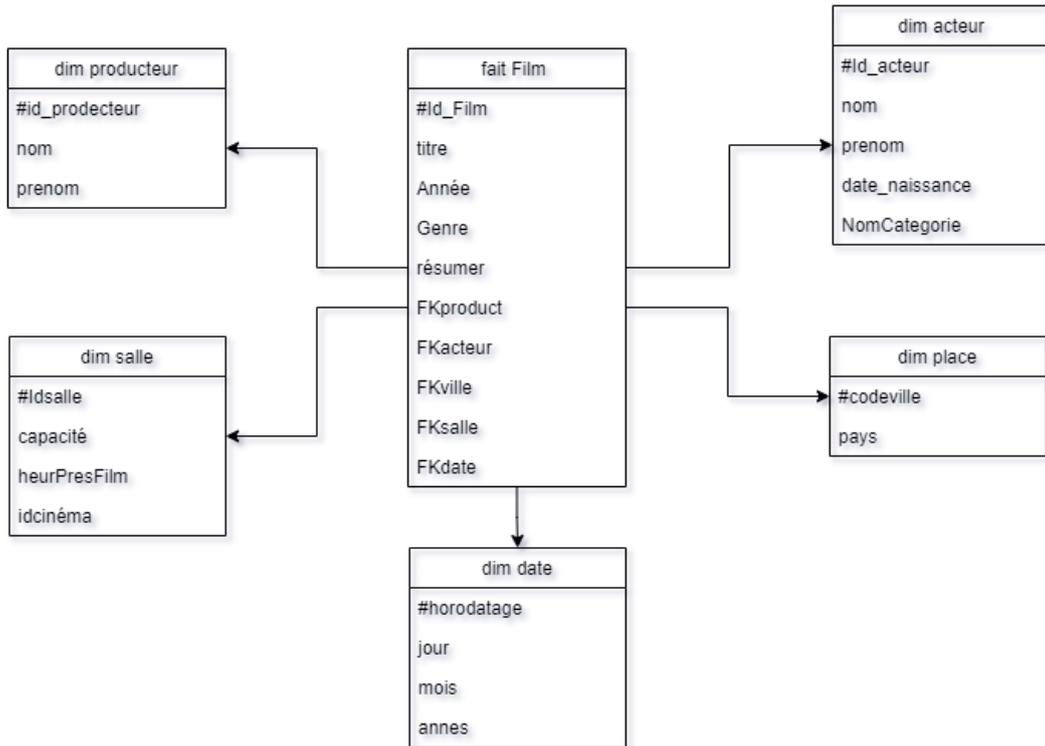


FIGURE 3.3 – le modèle de la base de données des films après appliqué l’algorithme

Dans l’exemple ci-dessous, Le schéma flocon de neige du modèle précédent qui contient une seule table de faits avec des dimensions qui peuvent être liées à d’autres dimensions.

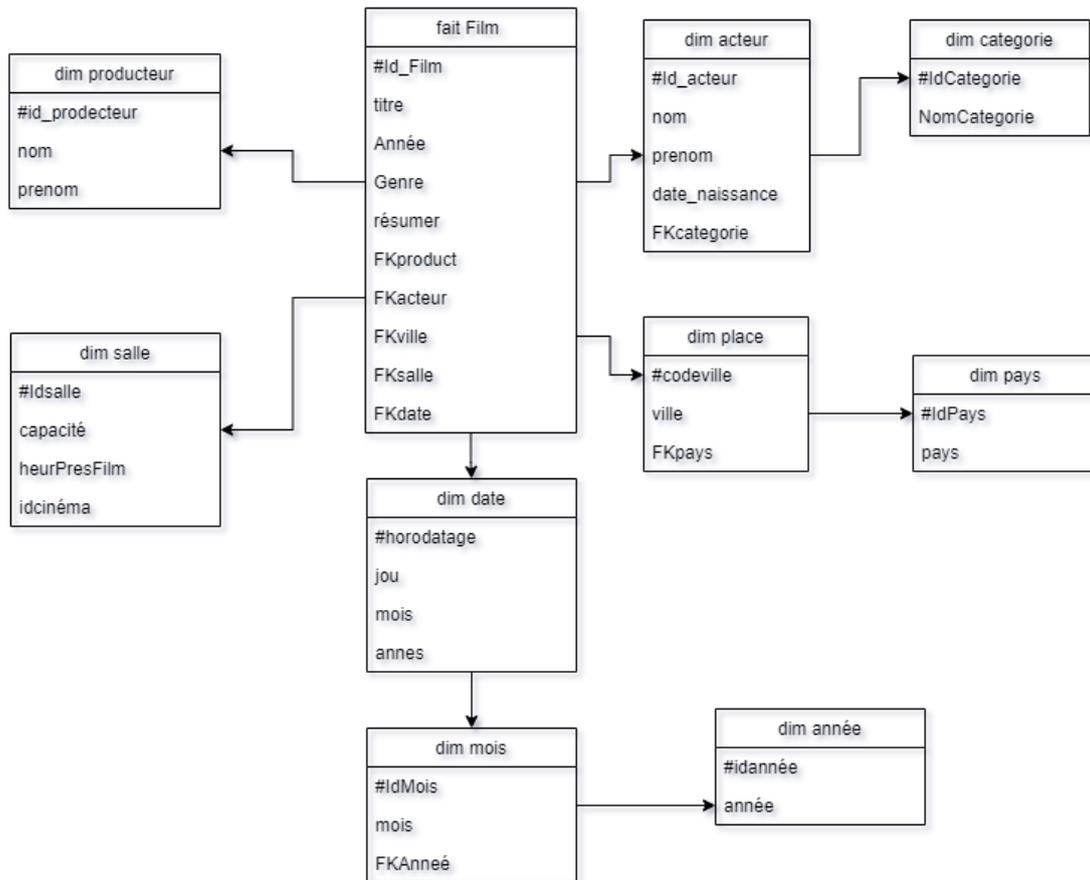


FIGURE 3.4 – Le schéma flocon de neige pour l'exemple BD film

3.4 partie 2 : les règles ECA : le module de génération des règles ECA

3.4.1 Introduction

Les règles Event-Condition-Action (ECA) sont un langage largement utilisé pour la spécification de haut niveau des contrôleurs dans les systèmes adaptatifs. L'évaluation et l'exécution de chaque règle ECA sont considérées comme indépendantes des autres, mais les interactions des actions de règle peuvent rendre les comportements du système imprévisibles ou dangereux. Les problèmes typiques sont la redondance des règles, les incohérences, la circularité ou les problèmes de sécurité liés à l'application. Par conséquent, il est nécessaire de coordonner les systèmes fondés sur des règles ECA afin de garantir les objectifs de sécurité[33].

3.4.2 L'idée

Les règles événement-condition-action (ECA) sont définies comme un ensemble de règles où chacune d'entre elles réagit de manière autonome à des événements simples ou complexes détectés activement ou passivement en évaluant une condi-

tion ou un ensemble de conditions et en exécutant une réaction chaque fois que l'événement se produit et la condition est vraie. La forme de la règle est :

ON Evènement IF Condition DO Action. Certaines caractéristiques sont que[33] :

- une règle n'est activée que par des événements.
- son exécution est autonome et indépendante des autres règles du système
- il implémente une réaction à l'événement entrant ;
- il contient une condition de garde pour exécuter de telles actions.
- La recherche sur les règles ECA est souvent liée aux systèmes de gestion de bases de données actives, où les événements représentent les modifications produites dans la base de données, et les règles ECA sont utilisées pour contrôler l'intégrité. Mais ils ont également été utilisés dans différents environnements de contrôle ou cadres d'adaptation, et il existe de nombreuses implémentations différentes de systèmes basés sur des règles ECA.
- Les règles de l'ECA sont un langage dérivé de la pratique et non construit à partir d'une définition formelle au départ. Ceci est comparable à d'autres cas comme StateCharts et ses multiples variantes et implémentations (par exemple en UML, ou dans les outils StateMate d'iLogix , ou dans Stateflow/ Simulink), ou comme des travaux sur la vérification de langages d'implémentation comme Java ou C. Il n'y a donc pas de référence unique ni de sémantique formelle, et elles ne peuvent être soumises à une analyse formelle en tant que telle.

3.5 La conception de système ADADW :

3.5.1 Langage de modélisation UML

UML (en anglais « Unified Modeling Language », en français « Langage de Modélisation Unifié ») est un langage de modélisation objet permettant de modéliser les classes et leurs interactions, la première version de UML a été publiée en novembre 1997 par l'OMG (Object Management Group), instance de normalisation internationale du domaine de l'objet.

Aujourd'hui, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet[34].

UML dans la dernière version propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles[34] :

1. **Les diagrammes structurels** ces diagrammes représentent l'aspect statique d'un système (classes, objets, composants. . .).
 - **Diagramme de classe** : ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements.

- **Diagramme d'objet** : le diagramme d'objet permet la représentation d'instances des classes et des liens entre instances.
 - **Diagramme de composant** : ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système.
 - **Diagramme de déploiement** : ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.
 - **Diagramme de paquetage** : ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants. . .).
 - **Diagramme de structure composite** : ce diagramme permet de décrire la structure interne d'un ensemble complexe composé. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent.
2. **Les diagrammes de comportement** : ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs.
- **Diagramme des cas d'utilisation** : ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système.
 - **Diagramme d'état-transition** : ce diagramme montre les différents états des objets en réaction aux événements.
 - **Diagramme d'activités** : ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données.
 - **Diagramme de séquence** : ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.
 - **Diagramme de communication (collaboration)** : ce diagramme est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.
 - **Diagramme global d'interaction** : ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
 - **Diagramme de temps** : ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

3.5.2 Processus de développement d'application :

2TUP (2 Track Unified Process) ou le processus en Y : est un processus de développement logiciel construit sur UML. Il apporte une réponse aux contraintes

de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Ce sont les chemins fonctionnels et techniques, qui correspondent aux deux axes de changement imposés au système informatique.

2TUP est réponde aux caractéristiques suivantes[35] :

- **Il est itératif et incrémental** : La définition d'itérations de réalisation est en effet la meilleure pratique de gestion des risques d'ordre à a fois technique et fonctionnel. Chaque itération garantit que les équipes sont capables d'intégrer l'environnement technique pour développer un produit final et fournit aux utilisateurs un résultat tangible de leurs spécifications.
- **Il est construit autour de la création et de la maintenance d'un modèle** : Le volume d'informations de ce modèle nécessite une organisation stricte qui présente les différents points de vue du logiciel à différents degrés d'abstraction.
- **Il est orienté composant** : Tant au niveau modélisation que production, c'est une garantie de souplesse pour le modèle lui-même et le logiciel qu'il représente. Cette pratique constitue le support nécessaire 'a la réutilisation logicielle et offre des perspectives de gains non négligeables.
- **Il est orienté utilisateur** : la spécification et la conception sont construites à partir des modes d'utilisation attendus par les acteurs du système.

La figure suivante présente le processus de développement en Y :[35]

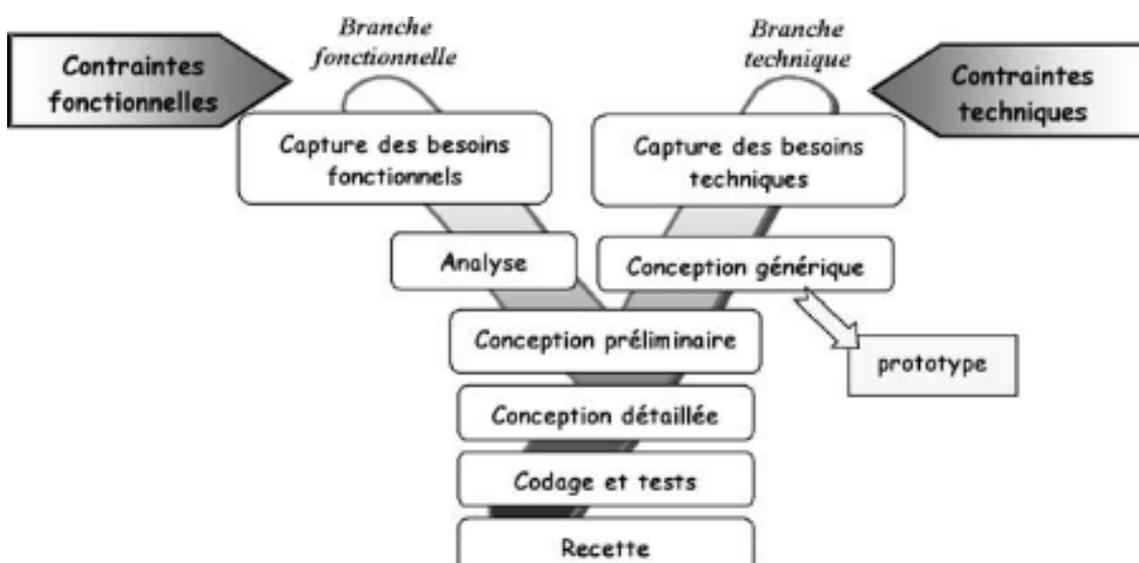


FIGURE 3.5 – Le processus de développement en Y

3.5.3 Etude préliminaire

L'étude préliminaire est la première étape de processus de développement en Y à l'objective de délimiter le périmètre du projet. Les principes de cette étape sont :

- Etablir un recueil initial des besoins fonctionnels et opérationnels.
- Modéliser le système comme une boîte noire en identifier les acteurs et les messages échangent entre les acteurs et le système.

3.5.3.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages éventuellement porteurs de données [35] .

Les acteurs d'une système peut être des utilisateurs humains et/ou des autres systèmes qui interagissent avec le système. Dans le cas de notre système, nous avons identifié un seul acteur principal qui agit directement avec le système, c'est le concepteur.

Concepteur de système :le rôle de concepteur est la création de schéma multidimensionnel, générer et consulter et géré les règles d'analyse.

3.5.3.2 Identification des messages

Un message est associé à deux occurrences d'évènements : un événement d'envoi et un événement de réception. Cette notion de message est également applicable pour décrire les interactions de plus haut niveau entre les acteurs et le système[35] .

- **Les messages acteur/system**

M1 :demande d'authentification.

M2 :demande d'inscription.

M3 :l'ajout des classes,attributs,classes associations, fonctions.

M4 :le calcul de la classe fait.

M5 :la validation de la classe fait.

M6 :éditer un Nom pour le cube résultat.

M7 :le chargement d'un schéma-BD.

M8 :l'ajout d'une règle d'analyse.

M9 :le choix d'un type évènement.

M10 :la modification d'une règle.

M11 :la suppression d'une règle.

M12 :demande de consulter les règles.

- **Les messages system/acteur**

M'1 :validation d'authentification.

M'2 :validation d'inscription.

M'3 :validation l'ajout d'une classe, un attribut, une association,une fonction.

M'4 :Affiche la classe fait.

M'5 :Confirmer la validation de la classe fait.

M'6 :valider le nom de cube.

M'7 :la validation d'un shéma-BD,et afficher les informations sur le schéma (le fait, les dimensions, le schéma multidimensionnel).

M'8 :valider l'ajout de la règle.

M'9 :valider le type d'évènement.

M'10 :validation de la modification d'une règle.

M'11 :validation de suppression d'une règle.

M'12 :la liste des règles.

3.5.3.3 Modélisation en contexte de notre système ADADW « automatic design of active data warehouse »

Tous les messages et les acteurs identifiés précédemment (dans l'identification des acteurs et des messages) peuvent être représentés de manière synthétique sur un diagramme suivant qui appelé le diagramme de contexte dynamique.

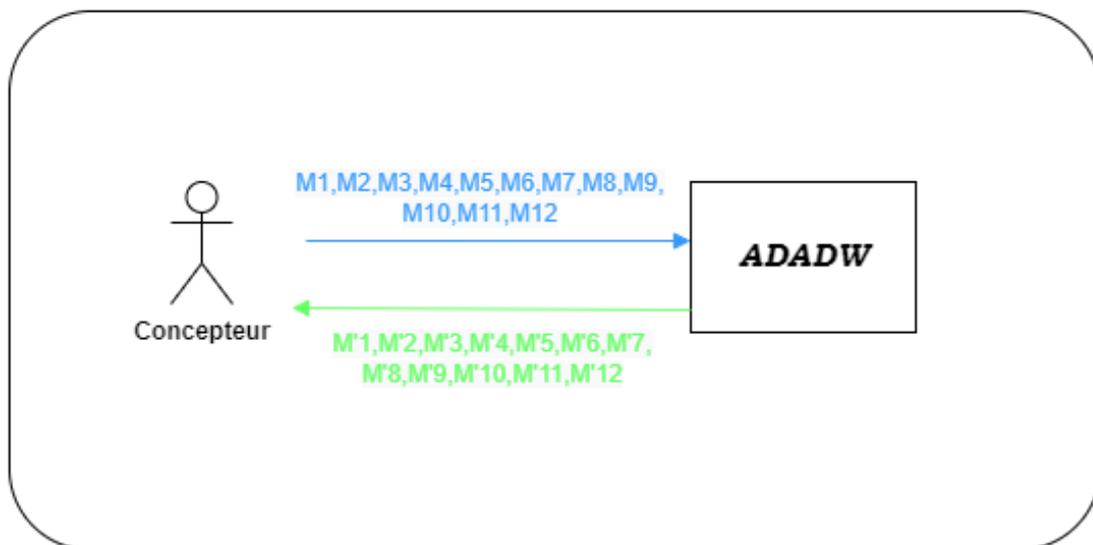


FIGURE 3.6 – le diagramme de contexte dynamique

3.5.3.4 Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. qui formalise et détaille ce qui a été ébauché au cours de l'étude

préliminaire.[35]

Un Besoin fonctionnel est un besoin lié au métier et au domaine adressé par le système, les exigences fonctionnelles du système sont traduites en détail par un ensemble de cas d'utilisation.

3.5.3.4.1 Identification et description des cas d'utilisations Chaque cas d'utilisation peut être structuré avec une fiche descriptive, nous allons présenter la fiche descriptive de chaque cas d'utilisation :

1. **Le cas « s'inscrire » :**

Le titre de cas d'utilisation : s'inscrire.

But : Permettre aux utilisateurs de créer des comptes pour accéder et utiliser l'application.

Acteur : concepteur.

Pré-condition : Authentification.

Scenario nominal :

- L'utilisateur demande l'inscription.
- Le système affiche le formulaire d'inscription.
- L'utilisateur remplit le formulaire et valide.
- Le système vérifie la validité des informations.
- Le système crée un compte pour l'utilisateur.

Exception A1 : Si un champ d'information n'est pas valide ou l'utilisateur existe déjà le système affiche un message d'erreur contenant l'erreur et demande à l'utilisateur de ressayer.

2. **Le cas «Crée le cube » :**

Le titre de cas d'utilisation : créer un cube .

But : Permet aux concepteurs de créer un modèle multidimensionnel pour un cube de données.

Acteur : concepteur.

Pré-condition : Authentification.

Scenario nominal :

- le concepteur édite le schéma-BD de la base de données.
- Le système vérifie la validité de la classe.
- Le système affiche un message de confirmation.
- Le système enregistre les données.
- le système calcul la classe fait.
- le concepteur valide le résultat.
- le système affiche les dimensions possibles.

— le concepteur édite un nom pour le cube résultat.

Exception A1 : Si le champ de la classe n'est pas valide ou la classe déjà existe : le système affiche un message d'erreur.

3. **Le cas « Gérer les règles d'analyses » :**

Le titre de cas d'utilisation : supprimer une règle.

But : Contrôler les règles.

Résumer : : supprimer une règle.

Acteur : concepteur.

Pré-condition : authentification.

Scenario nominal :

- le système affiche la liste des règles.
- l'utilisateur sélectionne une règle pour le supprimer.
- le système affiche un message de confirmation.

4. **Le cas « générer une règle » :**

But : la création d'une règle .

Résumer : : Enrichir la base de données.

Acteur : concepteur

Pré-condition : authentification.

Scenario nominal :

- le concepteur édite un nom pour la règle
- le système demande la sélection d'un type d'évènement.
- Le concepteur choisit un type d'évènement.
- le système affiche la liste des conditions.
- le concepteur choisit une condition.
- le système affiche une action possible pour la règle.

5. **Le cas « consulter les règles » :**

But : Consulter les différents listes.

Résumer : le concepteur peut consulter la liste d'évènement, liste des conditions, liste des actions.

Acteur : concepteur.

Pré-condition : authentification.

Scenario nominal :

- Le système affiche les icônes des listes.
- L'utilisateur demande d'afficher le contenu d'une liste.
- Le système affiche le contenu.

3.5.3.5 Capture des besoins non fonctionnels

En complément des besoins fonctionnelles, la capture des besoins techniques couvre toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification logicielle concerne donc les contraintes techniques. La spécification technique est une activité de la branche droite du Y, elle est essentielle pour la conception architecturale. [35]

Concernant notre application, nous avons identifié le besoin technique suivante :

La sécurité : L'application doivent être sécurisée, chaque utilisateur a un identifiant et un mot de passe(authentifier) pour accéder à son compte dans l'application. L'authentification d'utilisateur pour accéder à son compte est traduire par un cas d'utilisation « authentifie » qui nous allons expliquer dans la suit :

Le cas « authentifier » :

But :L'authentification sécurise les comptes et permet à l'utilisateur d'accéder à son compte pour utiliser l'application.

Acteur :concepteur.

Pré-condition :Application accessible.

Scenario nominal :

- L'utilisateur accède à l'application.
- Le système affiche le formulaire d'authentification.
- L'utilisateur saisit son nom d'utilisateur et son mot de passe.
- Le système vérifie l'existence du compte [A1].
- Le système renvoie l'interface correspondante.

Exceptioin A1 : Si un champ d'information n'est pas valide ou le compte n'existe pas, le système affiche un message d'erreur et demande à l'utilisateur de ressayer.

3.5.3.6 Diagramme de cas d'utilisation globale du système

Le diagramme suivant représenter le diagramme de cas d'utilisation de système de l'acteur « concepteur ».

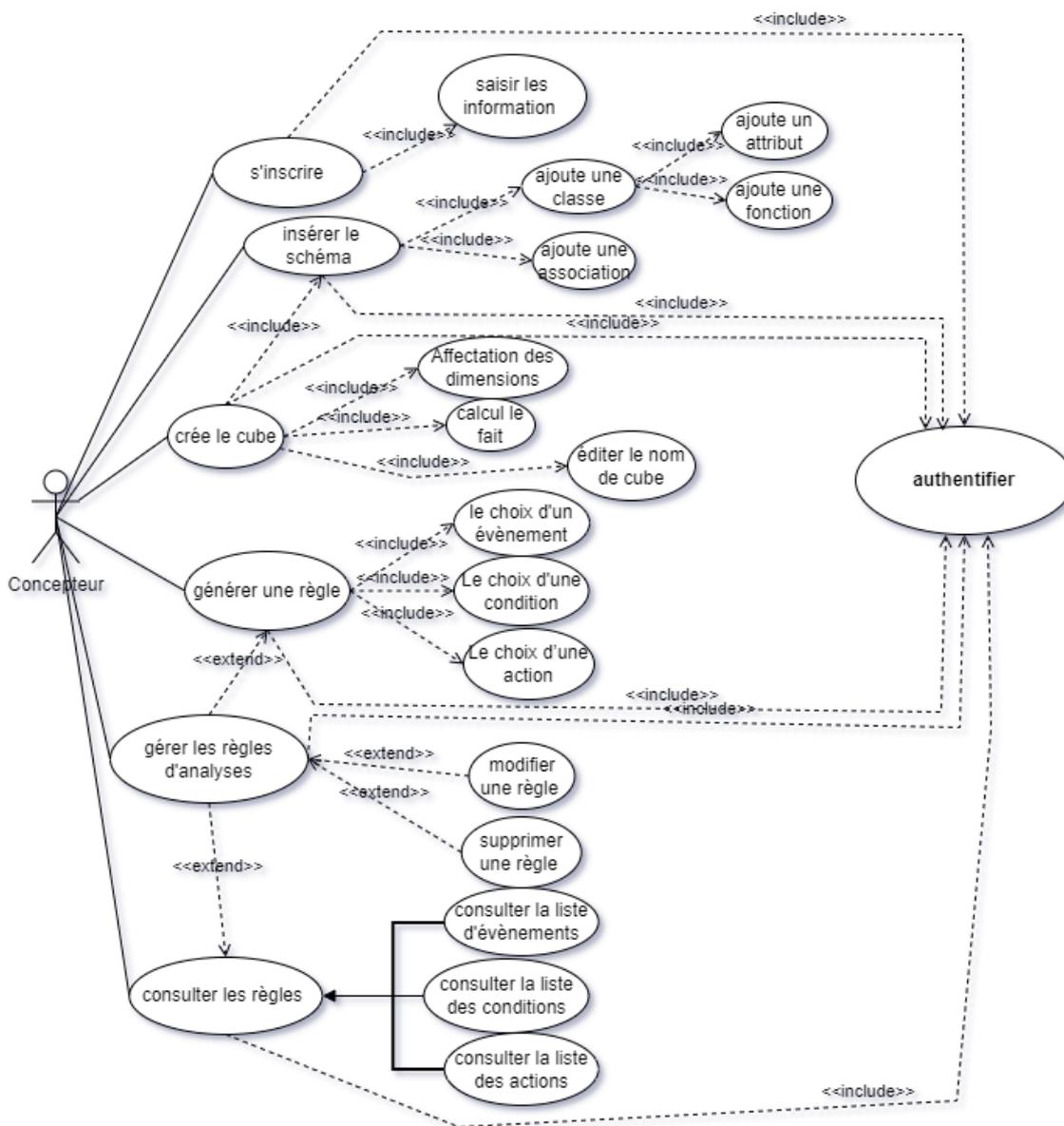


FIGURE 3.7 – Diagramme de cas d’utilisation de notre système [ADADW]

3.5.3.7 L’analyse

L’analyse est l’étape principale du processus 2TUP et suit l’étape de capture de besoin fonctionnel. Le but de l’analyse est de décrire précisément et correctement les besoins du client afin de concevoir une solution.

3.5.3.7.1 Diagramme de classe

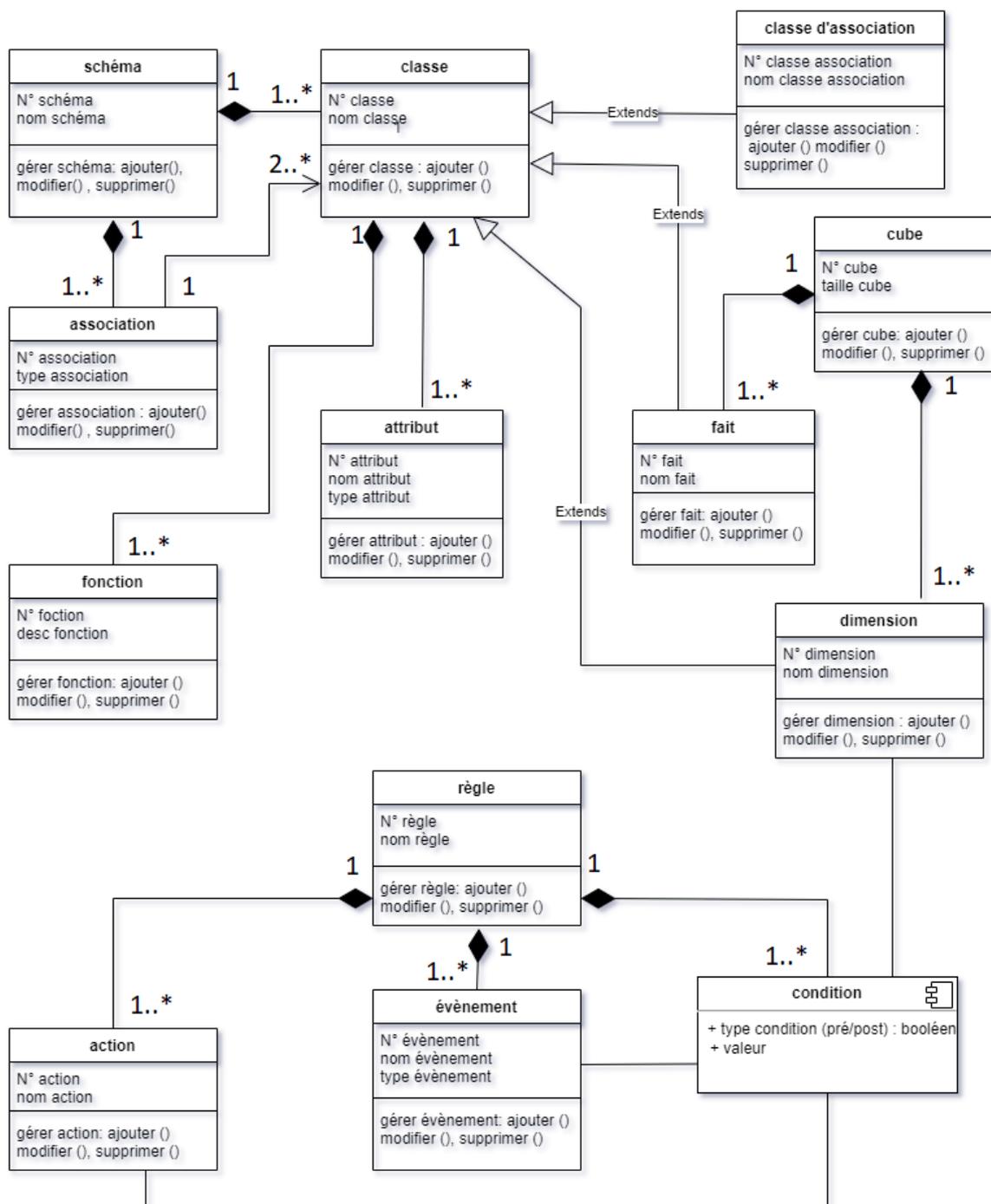


FIGURE 3.8 – Diagramme de classe de système

3.5.3.7.2 Diagrammes de séquence des cas d'utilisation Les diagrammes suivants représentent le diagramme de séquence de chaque cas d'utilisation :

1. Authentifier

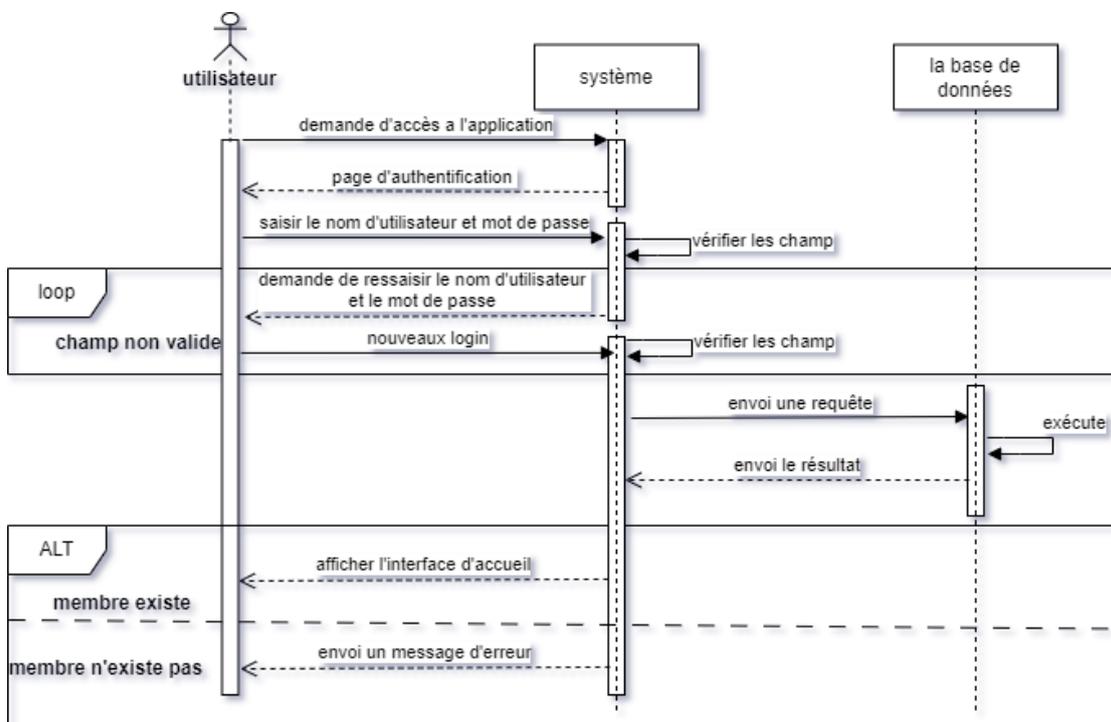


FIGURE 3.9 – Diagramme de séquence de cas d'utilisation authentifier

2. S'inscrire

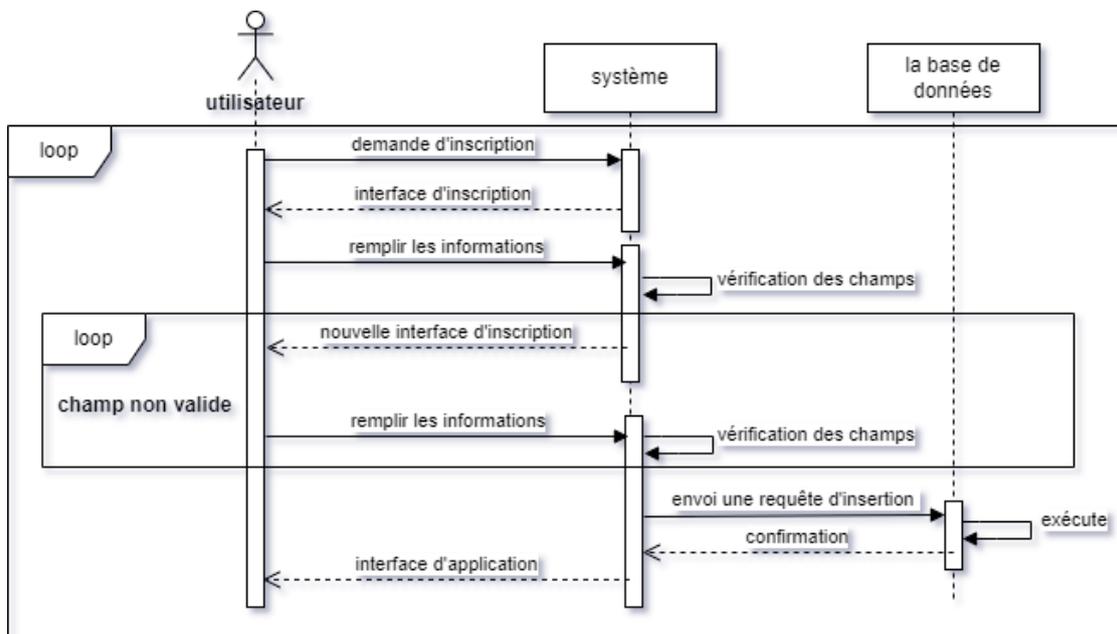


FIGURE 3.10 – Diagramme de séquence de cas d'utilisation s'inscrire

3. Consulter les règles

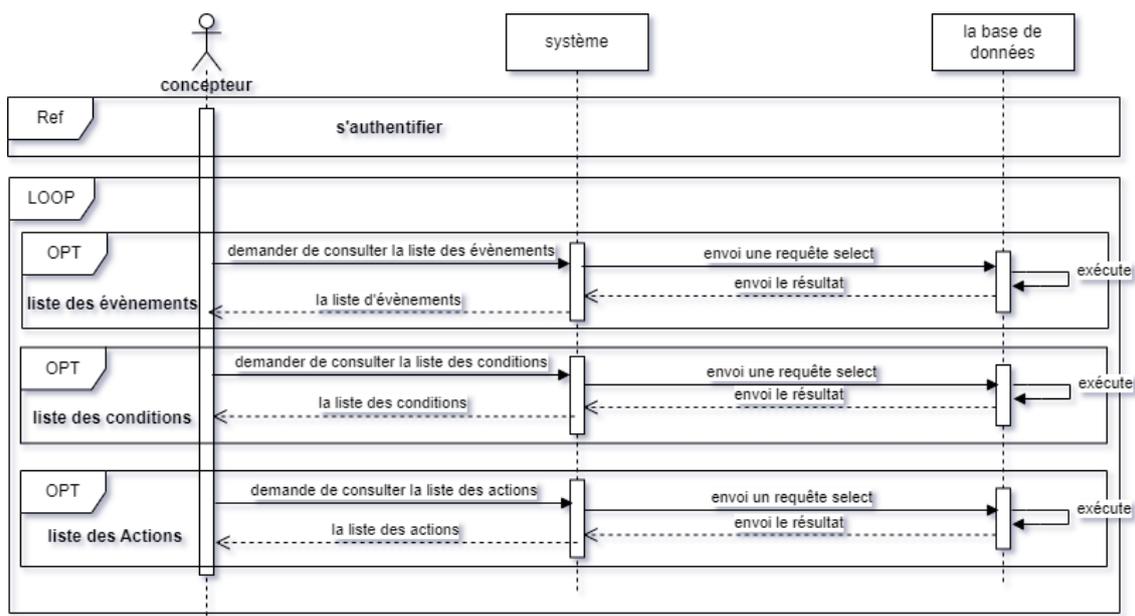


FIGURE 3.11 – Diagramme de séquence de cas d'utilisation consulter les règles

4. Supprimer une règle

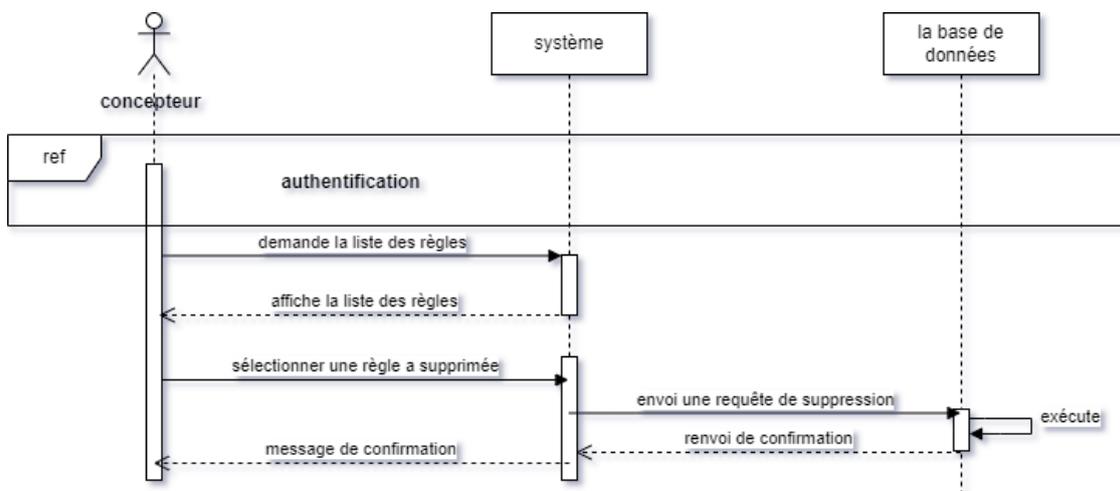


FIGURE 3.12 – Diagramme de séquence de cas d'utilisation supprimer une règle

5. créé le cube :

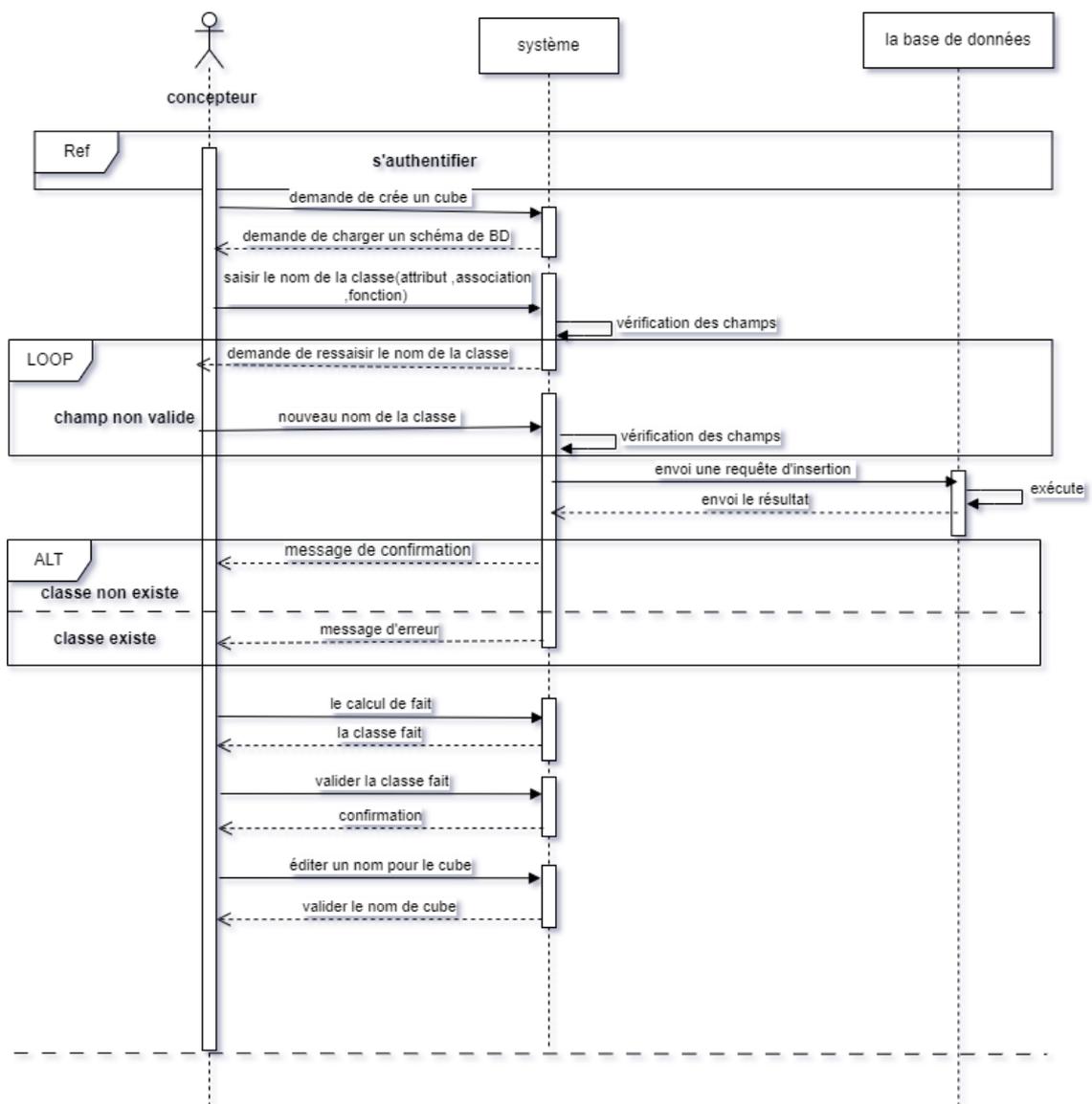


FIGURE 3.13 – Diagramme de séquence de cas d'utilisation créer le cube

6. Générer une règle ECA :

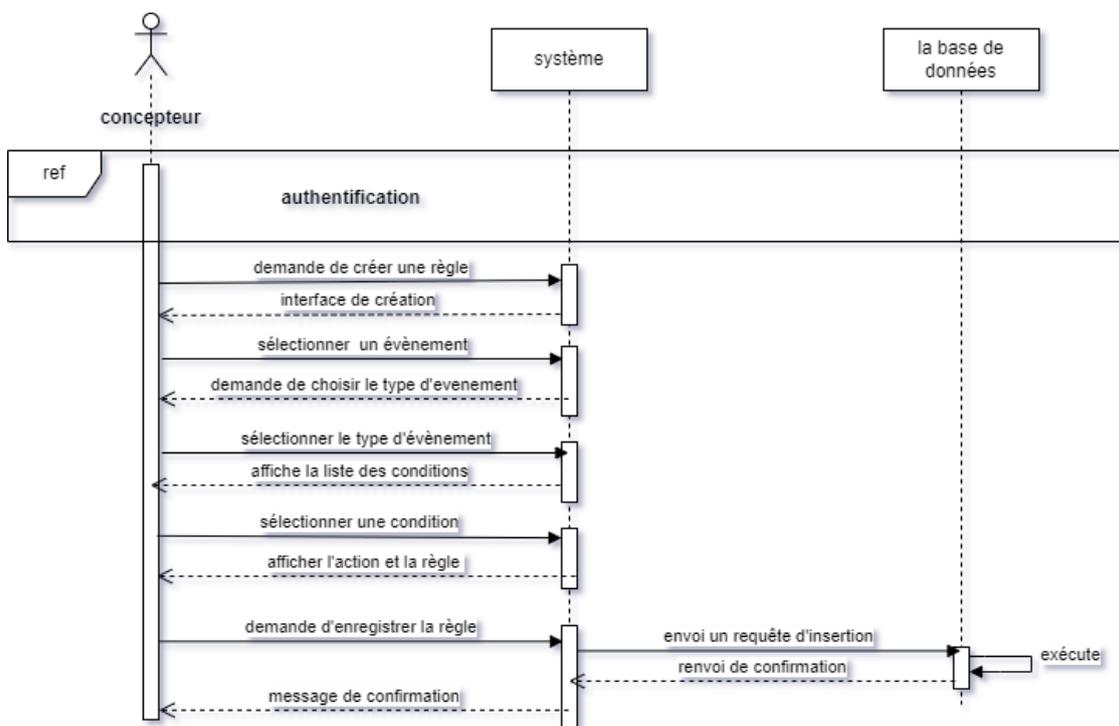


FIGURE 3.14 – Diagramme de séquence de cas d’utilisation générer une règle ECA

3.6 Conclusion

Dans ce chapitre, nous avons présenté l’architecteure générale de notre système, nous avons décrit l’idée de base de passage d’un modèle relationnel vers le modèle multidimensionnel ensuite nous avons présenté l’algorithme de génération du modèle multidimensionnel à partir du modèle relationnel avec un exemple sur le déroulement de l’algorithme, En suite , nous avons décrit de façon détaillée les étapes faites pour la conception de l’application, et spécifier les processus utilisés pour obtenir des différents modèles conceptuels. Enfin, ce chapitre nous a permis de préparer la phase de mise en œuvre qui concrétisera tout ce qui a été présenté jusque-là.

CHAPITRE 4

IMPLÉMENTATION ET RÉALISATION

4.1 Introduction

Ce chapitre est dédié à la réalisation et la mise en œuvre de notre application, nous commençons par une présentation des différents outils, langages, et environnement de développement utilisés, ainsi qu'un aperçu général sur la phase pratique qui consiste à décrire les différentes fonctionnalités disposées lors de l'utilisation de notre logiciel, ce qui permet d'éclaircir chaque étape de la réalisation.

4.2 Choix Technique

4.2.1 Langages de développement utilisés

4.2.1.1 Language JAVA

JAVA est un langage de programmation orienté objet est né officiellement en 1995, ce dernier présente de nombreux avantages comme par exemple le fait d'être portable. C'est-à-dire qu'un programme réalisé en JAVA peut être exécuté sur différentes plateformes comme Windows, Mac, Linux[36].

4.2.2 JavaFX

JavaFX étend la puissance de Java en permettant aux développeurs d'utiliser n'importe quelle bibliothèque Java au sein des applications JavaFX. il permet aux développeurs Java de créer une interface graphique pour des applications de bureau, des applications internet riches et des applications smartphones et tablettes tactiles. Les développeurs peuvent ainsi multiplier les possibilités dans Java et utiliser la technologie de présentation offerte par JavaFX pour créer des expériences

visuelles époustouflantes. En tant qu'utilisateur, vous pourrez exécuter les applications JavaFX dans un navigateur ou les déplacer sur le Bureau par un simple glisser-déplacer. ainsi il Améliore l'efficacité du travail entre concepteurs et développeurs, les premiers pouvant utiliser les outils de leur choix tout en collaborant avec les seconds[37].

4.2.3 Environnement de développement

4.2.3.1 IntelliJ IDEA

IntelliJ IDEA est un IDE intelligent et tenant compte du contexte qui permet de travailler sur toutes sortes d'applications en Java et dans d'autres langages de la JVM tels que Kotlin, Scala et Groovy. De plus, IntelliJ IDEA Ultimate vous aide à développer des applications web full-stack grâce à ses puissants outils intégrés, à la prise en charge de JavaScript et de ses technologies connexes et à la prise en charge avancée de frameworks populaires tels que Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus et Helidon. IntelliJ IDEA peut être complété par des plugins gratuits développés par JetBrains afin de pouvoir travailler avec d'autres langages de programmation, parmi lesquels Go, Python, SQL, Ruby et PHP.

Il fournit un accès rapide à toutes les fonctionnalités et à tous les outils intégrés qu'ils sont utiles pour un travail, il comprend de nombreux autres langages de programmation, tels que Groovy, Kotlin, Scala, JavaScript, TypeScript et SQL, et fournit une assistance au codage intelligente pour chacun d'eux. L'indexation initiale du code source permet à l'IDE de créer une carte virtuelle de votre projet. En utilisant les informations de la carte virtuelle, il peut détecter les erreurs à la volée, suggérer des variantes de saisie semi-automatique du code avec une connaissance précise du contexte, il propose un ensemble d'outils puissants pour définir les paramètres d'exécution et de build de votre application, déboguer votre code et appliquer et développer des tests JUnit directement dans l'IDE[38].

4.2.3.2 PHPmyadmin

phpMyAdmin est un outil logiciel gratuit écrit en PHP , destiné pour gérer l'administration de MySQL sur le Web. phpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tandis que vous avez toujours la possibilité d'exécuter directement n'importe quelle instruction SQL[39].

4.2.4 Outils logiciel

4.2.4.1 L'éditeur Draw.io

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter[40].

4.3 Présentation de l'application

4.3.1 Page d'authentification

Au démarrage du logiciel, l'interface d'authentification apparaîtra à l'écran. Cette étape consiste à protéger les données et à protéger l'application. L'utilisateur doit s'identifier par un nom d'utilisateur et un mot de passe pour pouvoir accéder à page d'accueil.

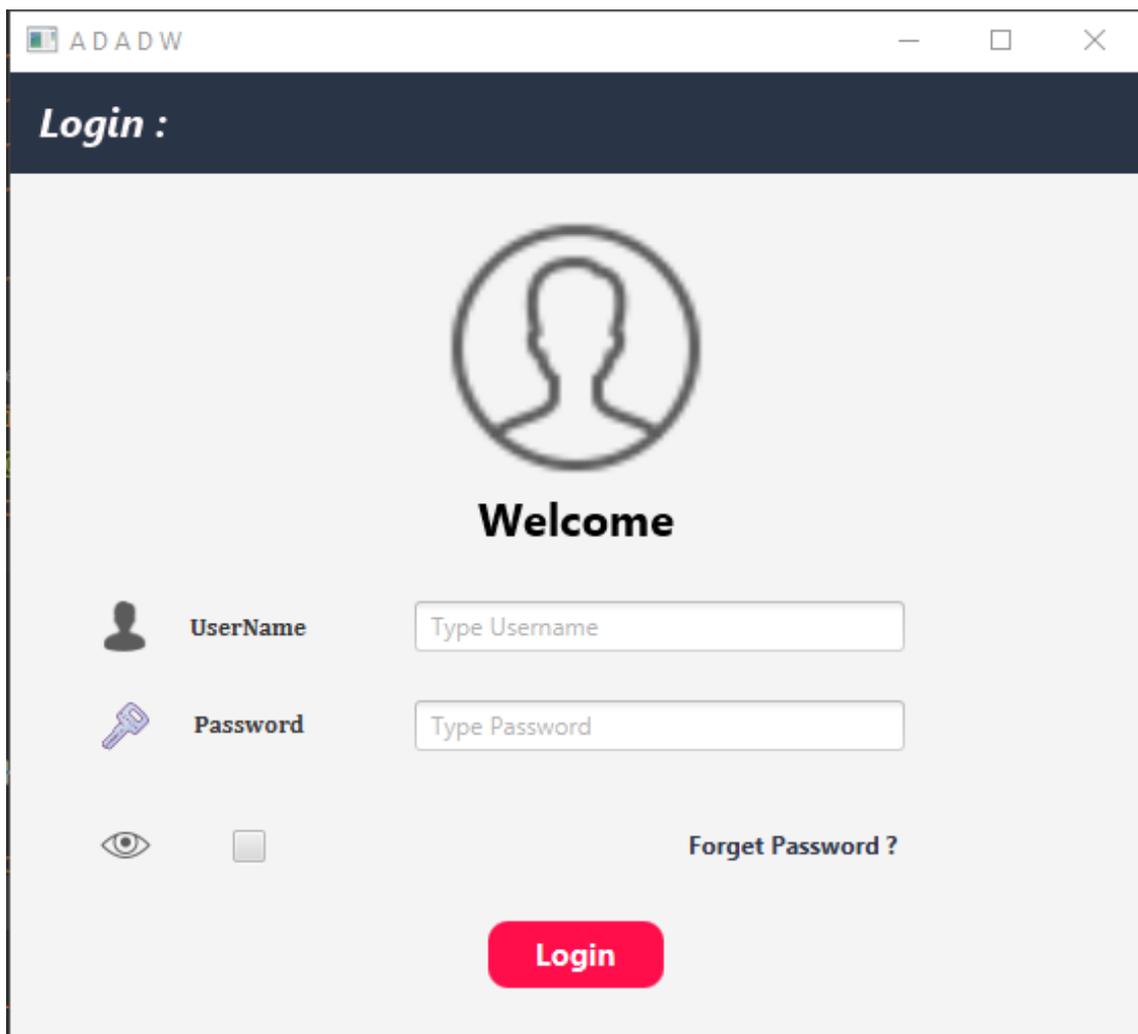


FIGURE 4.1 – Capture d'écran de l'interface authentification.

4.3.2 Page d'accueil

Si le nom d'utilisateur et le mot de passe sont vérifiés par le système, l'interface principale s'affiche. L'interface est composée de six boutons :

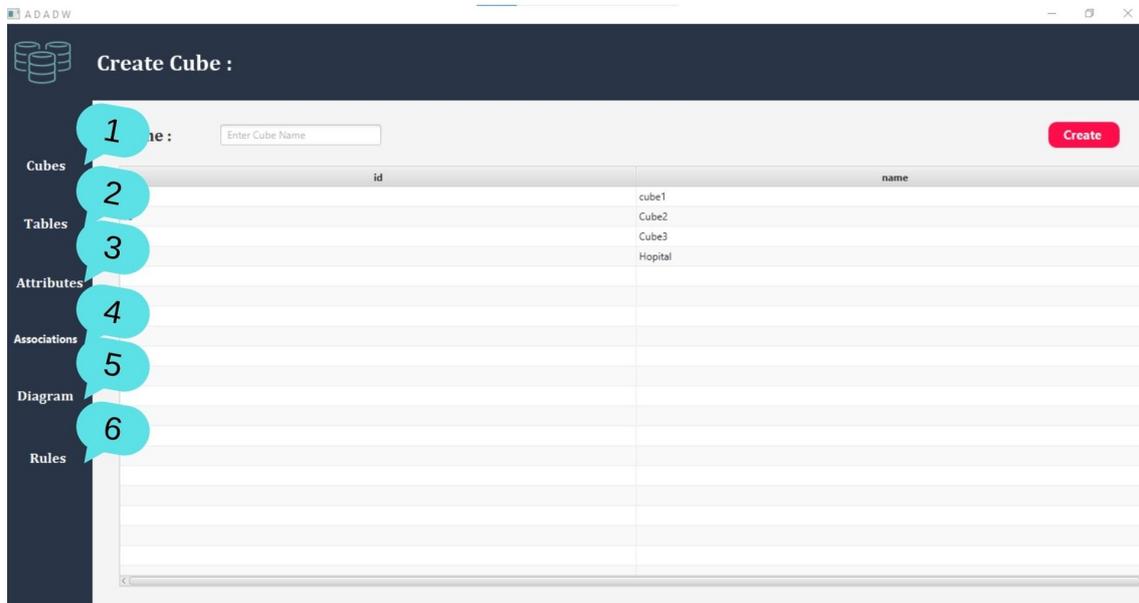


FIGURE 4.2 – Capture d'écran de l'interface d'accueil.

1. pour accéder à la page de création du cube.
2. pour accéder à la page d'ajout des tables.
3. pour accéder à la page d'ajout des attribut.
4. pour accéder à la page d'ajout des associations.
5. pour accéder à la page d'affichage du schéma multidimensionnel.
6. pour accéder à la page d'ajout des règles.

4.3.2.1 Cube

Si l'utilisateur choisi « cube », l'interface suivante s'affiche :

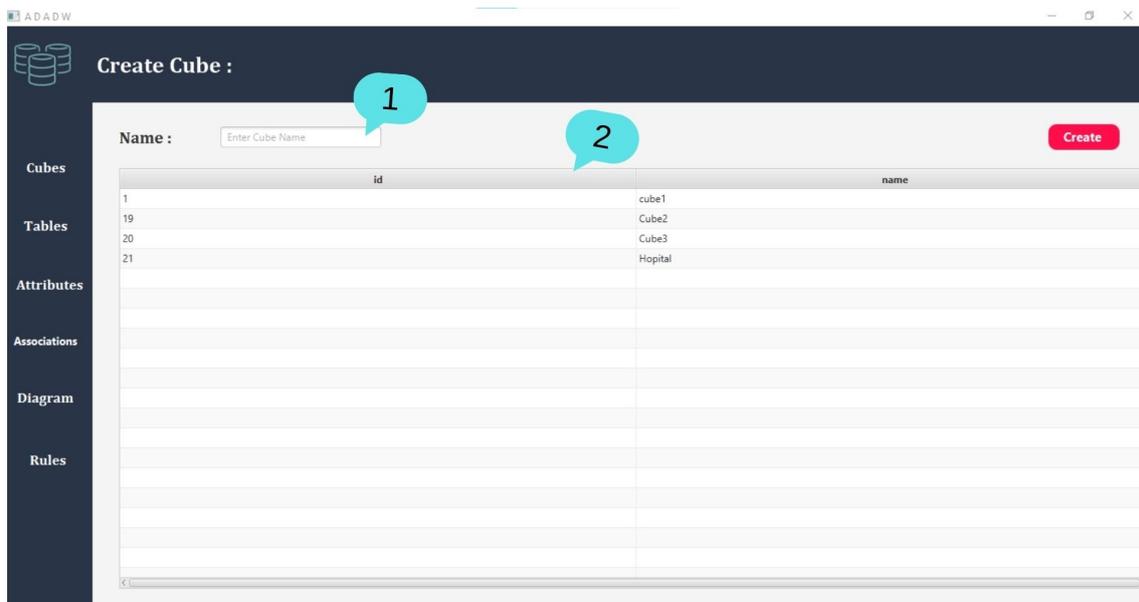


FIGURE 4.3 – Capture d’écran de l’interface crée cube.

1. pour ajouter le nom de cube.
2. pour visualiser les cubes qui sont déjà créés.

4.3.2.2 Tables

Si l’utilisateur choisi « tables », l’interface suivante s’affiche :

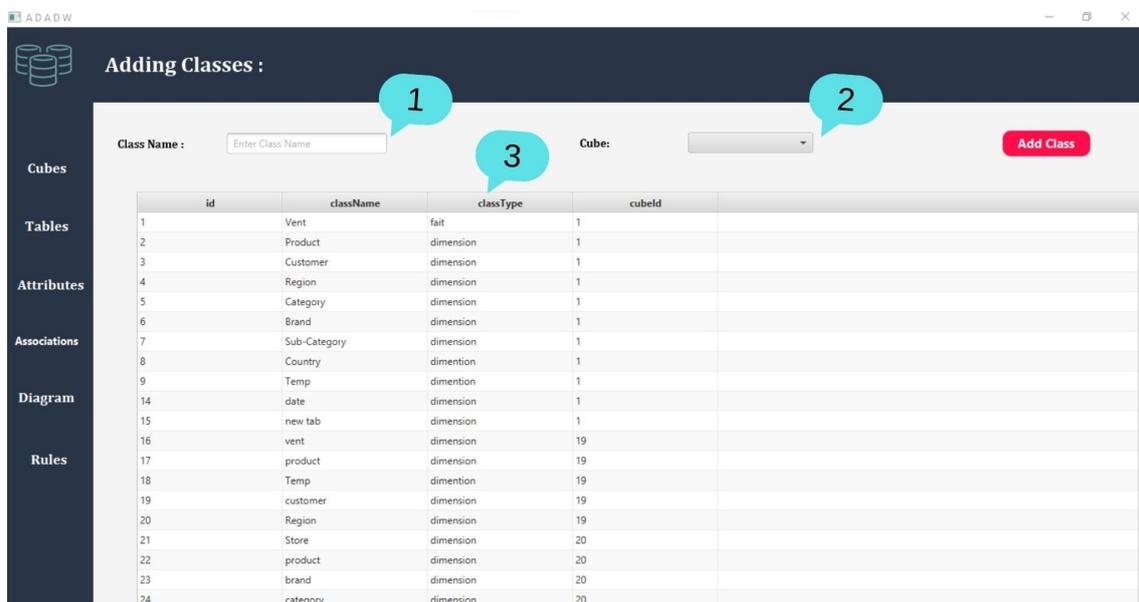


FIGURE 4.4 – Capture d’écran de l’interface ajoute classe.

1. pour ajouter le nom de la classe.
2. pour sélectionner le cube.
3. pour vésualiser les classes qui sont déjà crée.

4.3.2.3 attributes

Si l'utilisateur choisi « attributes », l'interface suivante s'affiche :

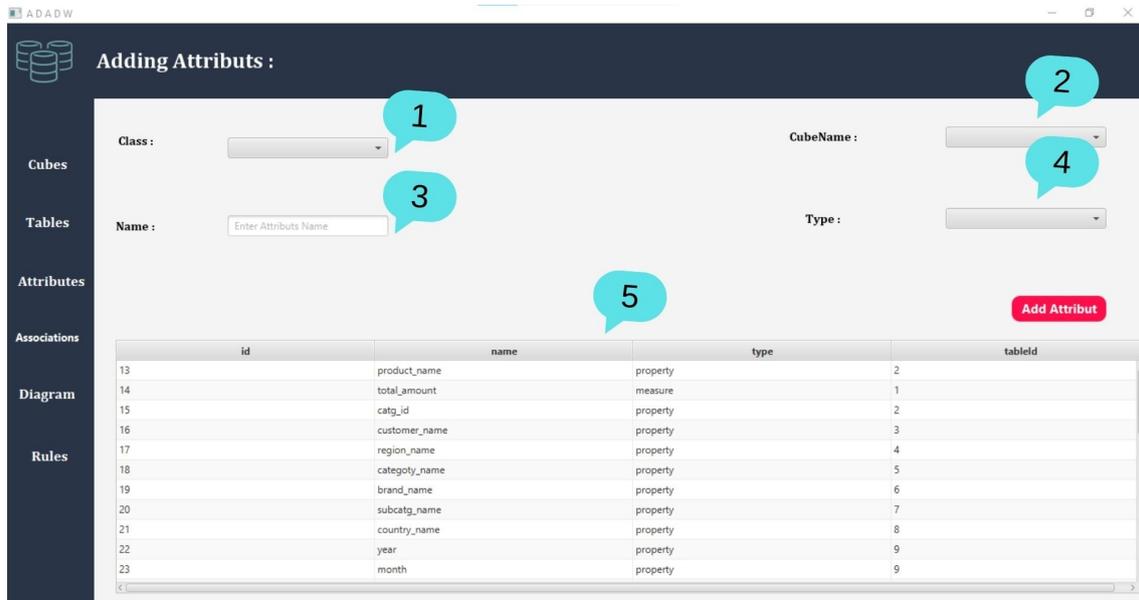


FIGURE 4.5 – Capture d'écran de l'interface ajoute attribut.

1. pour sélecionner la classe.
2. pour sélectionner le cube.
3. pour ajouter le nom de l'attribut.
4. pour sélectionner le type d'attribut {primary key, foreign key, simple}.
5. pour visualiser les attributs qui sont déjà créés.

4.3.2.4 Associations

Si l'utilisateur choisi « Associations », l'interface suivante s'affiche :

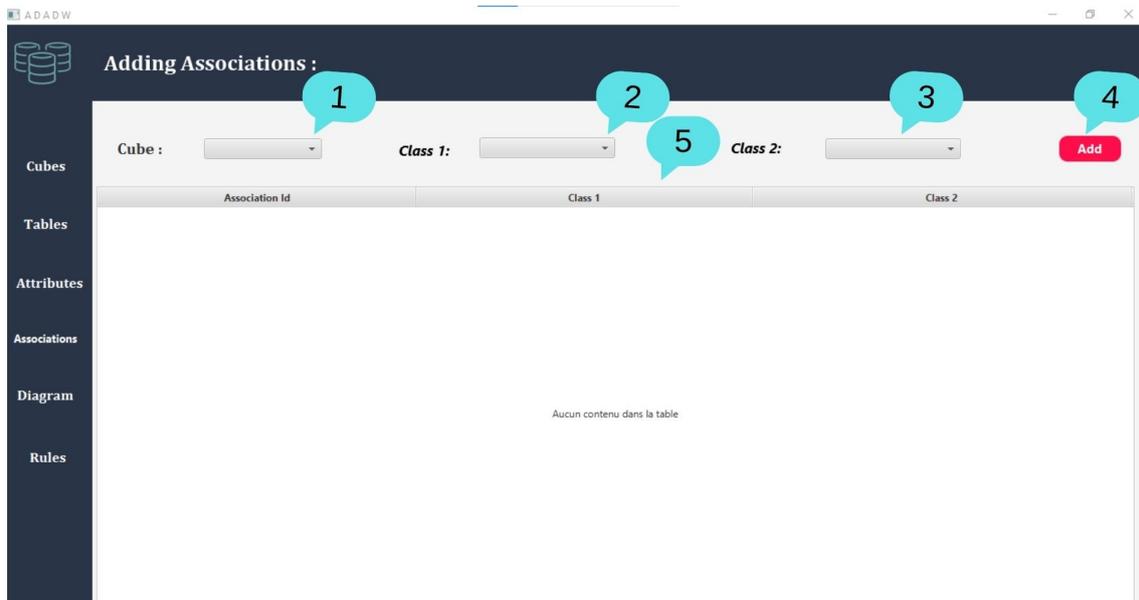


FIGURE 4.6 – Capture d’écran de l’interface ajoute association.

Dans cette étape nous pouvons ajouter des associations et des relations entre tables pour rendre le modèle relationnel :

1. pour sélectionner le cube.
2. pour sélectionner la première classe de l’association.
3. pour sélectionner la deuxième classe de l’association.
4. la validation de l’ajout.
5. affichage des associations disponibles (sont déjà ajoutées avant).

4.3.2.5 Diagram

Si l’utilisateur choisi « Diagram », l’interface suivante s’affiche :

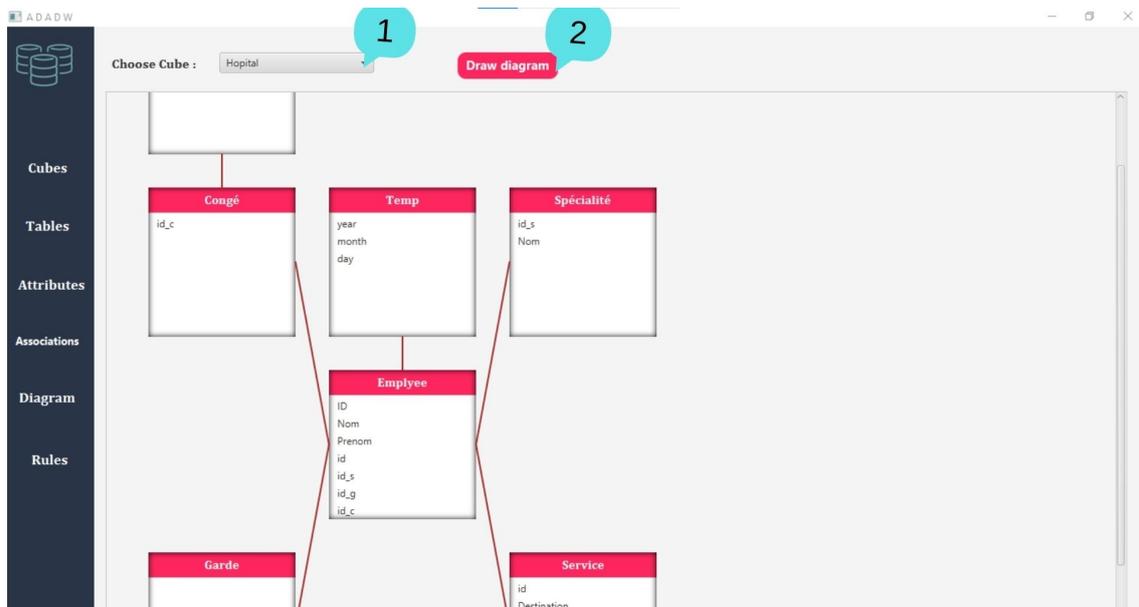


FIGURE 4.7 – Capture d’écran de l’interface schéma multidimensionnel.

1. le choix d’un cube dans une liste des cubes disponibles.
2. pour le schéma multidimensionnel du cube sélectionné.

4.3.2.6 Add rules

Si l’utilisateur choisi « Add Rule », l’interface suivante s’affiche :

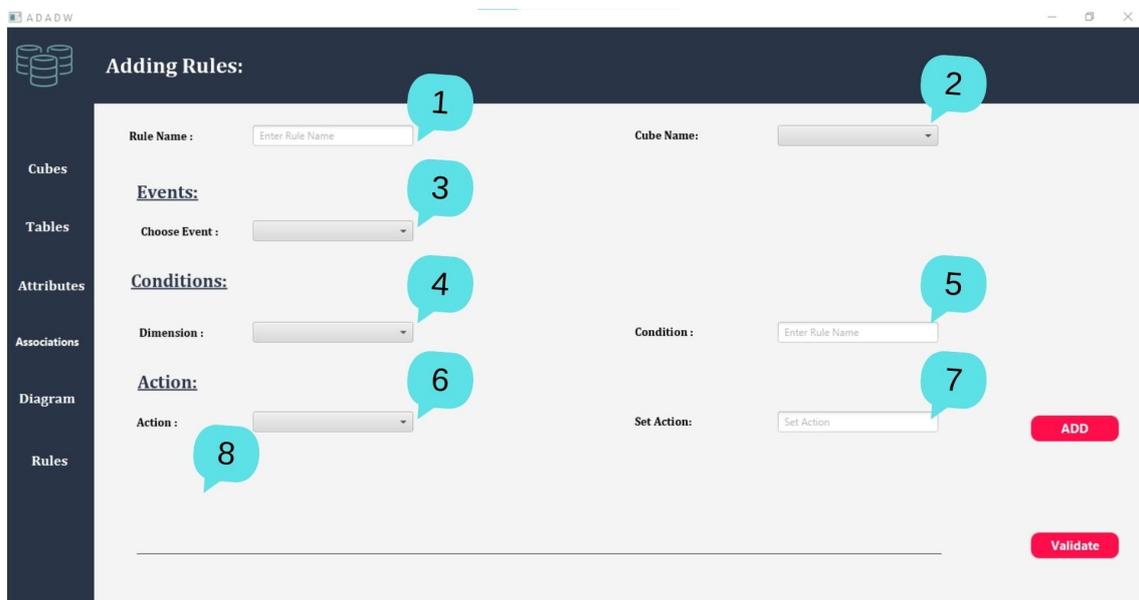


FIGURE 4.8 – Capture d’écran de l’interface ajoute règle.

Cette étape nous donne la possibilité d'ajout des règles ECA, et on les enregistre dans la base des règles ECA. Cette interface fonctionne comme suit :

1. insérer le nom de la règle.
2. Le choix de cube concerné pour notre règle.
3. Le choix d'évènement dans une listes des évènements possibles.
4. Choix de la dimension concerné pour le déclenchement de cet évènement.
5. l'insertion de la condition requise.
6. Choisissez le type de l'action(insert,delete, update).
7. L'insertion de l'action a exécutée.
8. pour visualiser la règle.

4.4 Conclusion

Dans cette dernière partie de notre travail, Nous avons brièvement décrit les outils utilisés pour la réalisation de notre application, et nous avons présenté aussi une vue de l'application finale via quelques interfaces et ses fonctionnalités.

CONCLUSION GÉNÉRALE

Au cours de ce mémoire, nous avons abordé les systèmes décisionnels, ou plutôt les systèmes et les outils d'aide à la décision, dont les entrepôts de données, où nous avons détaillé le concept général des entrepôts de données et leurs conceptions. En raison de la croissance exponentielle des données, nous avons cité le concept d'automatisation dans l'entreposage de données. Nous avons donc examiné les objectifs et les implications de l'automatisation des données et de l'entreposage des données. En plus nous avons présenté quelques outils d'automatisation qui permet de construire un entrepôt (ou cube) à partir d'une base de données source et donne la possibilité de la manipulation de cette base multidimensionnelle.

Objectifs atteints :

Enfin, nous avons implémenté une approche d'automatisation de la conception des entrepôts de données notre système repose sur une distinction de deux espaces de stockage : l'entrepôt de données passif et les règles ECA.

la première partie de travail basée sur la création d'un cube de données à partir d'une base de données source, ou la sélection de la table fait et calculée par un algorithme prédéfini, afin de présenter le schéma multidimensionnel adéquat. La deuxième partie consiste à ajouter des règles ECA dynamique.

Perspectives :

les différentes perspectives que l'on peut adopter :

- Au cours de ce travail, nous avons utilisé une seule base de données source, aussi nous avons travaillé sur un seul cube et une seule table de fait. Il est nécessaire de prévoir l'extension de ce système pour prendre en compte plusieurs bases de données sources.
- Donc notre but est d'améliorer ce travail par l'ajout de plusieurs sources de données, et nous essayons de travailler sur plusieurs cubes et tables de fait.
- De plus, une façon possible est d'intégrer la dynamique des schémas multidimensionnels.

BIBLIOGRAPHIE

- [1] N. Hassi, “Conception et réalisation d’un système décisionnel pour les collectivités locales (étude de cas : Des décisions financières),” 2017.
- [2] O. Teste, *Modélisation et manipulation d’entrepôts de données complexes et historisées*. PhD thesis, Université Paul Sabatier-Toulouse III, 2000.
- [3] S. Khouri, “Modélisation conceptuelle à base ontologique d’un entrepôt de données,” *Mémoire de Magistère. Université Oued-Smar Alger*, 2009.
- [4] “L’architecteur d’un entrepôt de donnée..” <https://slideplayer.fr/slide/9445149/>.
- [5] K. Boukhalfa, *De la conception physique aux outils d’administration et de tuning des entrepôts de données*. PhD thesis, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d’Aérotechnique-Poitiers, 2009.
- [6] S. Bouattour, “Les entrepôts de données actifs : automatisation des scénarii d’analyse,” *Université Lumière Lyon*, vol. 2, 2007.
- [7] <https://www.journaldunet.com/solutions/dsi/1490565-quelle-est-la-meilleure-approche-de-l-automatisation-pour-votre-entrepot/>.
- [8] <https://www.softwaretestinghelp.com/data-warehouse-etl-automation-tools/>.
- [9] <https://fre.myservername.com/11-best-data-warehouse-etl-automation-tools>.
- [10] S. Bouattour, R. Ben Messaoud, and O. Boussaid, “Modélisation de règles d’analyse dédiées aux entrepôts de données actifs,” *Deuxième Atelier des Systèmes Décisionnels (ASD’2007), Sousse, Tunisie. Octobre*, 2007.
- [11] R. Kimball, *The data warehouse toolkit : practical techniques for building dimensional data warehouses*. John Wiley & Sons, Inc., 1996.
- [12] A. Devisy, *Livre blanc e-Business intelligence, l’effet internet sur le décisionnel*. 2002.
- [13] A. Estella, *Eléments méthodologiques pour le développement des systèmes décisionnels dans un contexte de réutilisation*. PhD thesis, Université Paul Sabatier-Toulouse III, 2007.

- [14] B. Burquier, *Business Intelligence avec SQL Server 2008 : Mise en oeuvre d'un projet décisionnel*. Dunod, 2009.
- [15] S. HOUARI and A. KARIM, "Conception et réalisation d'un data warehouse pour la mise en place d'un système décisionnel au sein de la sntr-groupe," *Mémoire de fin d'étude, Ecole national Supérieur d'Informatique*, 2011.
- [16] A. Meguireche, *Conception et implémentation d'un système décisionnel TAOUAB-Bousaâda*. PhD thesis, FACULTE : Mathématique et Informatique DEPARTEMENT : Informatique-OPTION . . . , 2020.
- [17] W. H. Inmon, *Building the Data Warehouse 3ème édition*. John Wiley & Sons, Inc., 2002.
- [18] A. Daâs, *Optimisation des requêtes dans le data were house*. PhD thesis, 2018.
- [19] S. SAICHI, *Optimisation de requêtes dans les entrepôts de données*. PhD thesis, Université d'Oran1-Ahmed Ben Bella, 2009.
- [20] A. B. Charef, "Systèmes d'information décisionnels," 2019.
- [21] C. Ghezzi, "Designing data marts for data warehouses," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 10, no. 4, pp. 452–483, 2001.
- [22] M. O. Benali and M. N. Rechoum, *La conception et la réalisation d'un système décisionnel*. PhD thesis, Université Saad Dahleb - Blida1, 2020.
- [23] E. Ziyati, "Optimisation de requêtes olap en entrepôts de données : Approche basée sur la fragmentation génétique," 2010.
- [24] S. Rivest, "Investigation des modes d'intégration physique entre un serveur de base de données multidimensionnelle et un sig," *Rapport de DEA Informatique, Laval : Université Laval, Canada*, p. 84, 2000.
- [25] R. Kimball and M. Ross, *Entrepôts de données : guide pratique de modélisation dimensionnelle*. Vuibert informatique, 2003.
- [26] E. F. Codd, "Providing olap (on-line analytical processing) to user-analysts : An it mandate," <http://www.arborsoft.com/papers/coddTOC.html>, 1993.
- [27] T. Thalhammer, M. Schrefl, and M. Mohania, "Active data warehouses : complementing olap with analysis rules," *Data & Knowledge Engineering*, vol. 39, no. 3, pp. 241–269, 2001.
- [28] S. Bouattour and O. Boussaïd, "Les entrepôts de données actifs : modélisation des règles d'analyse.," in *EDA*, pp. 139–154, 2009.
- [29] S. Bouattour, J. Feki, H. Ben Abdallah, R. Ben Messaoud, and O. Boussaïd, "Proposition d'une taxonomie d'événement dans les entrepôts de données actifs," *Les huitième journée de GEI 2008, Sousse, Tunisie*, 2008.
- [30] <https://www.astera.com/fr/type/blog/data-warehouse-automation-a-complete-guide/>

- [31] S. Bouattour, O. Boussaid, H. Ben-Abdallah, and J. Feki, “Modélisation et analyse dans les entrepôts de données actifs,” *TSI-Technique et Science Informatiques*, vol. 30, no. 8, p. 975, 2011.
- [32] N. El Moukhi, I. El Azami, A. Mouloudi, and A. Elmounadi, “Towards a new automatic data warehouse design method,” *Electronic Journal of Information Technology*, no. 11, 2018.
- [33] J. Cano, G. Delaval, and E. Rutten, “Coordination of eca rules by verification and control,” in *International Conference on Coordination Languages and Models*, pp. 33–48, Springer, 2014.
- [34] J. Gabay and D. Gabay, *UML 2 Analyse et conception : Mise en œuvre guidée avec études de cas*. Dunod, 2008.
- [35] P. Roques and F. Vallée, *UML 2 en action : de l’analyse des besoins à la conception*. Editions Eyrolles, 2011.
- [36] <https://www.techno-science.net/glossaire-definition/Java-langage.html>.
- [37] [://www.java.com/fr/download/help/javafx.html](http://www.java.com/fr/download/help/javafx.html).
- [38] <https://www.jetbrains.com/fr-fr/idea/features/>.
- [39] <https://www.phpmyadmin.net/>.
- [40] <http://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>.

RÉSUMÉ

Les entrepôts de données actifs est une nouvelle catégorie des systèmes d'aide à la décision, qui offre la possibilité d'automatiser la prise de décision, Les entrepôts de données actifs étendent les entrepôts de données traditionnels avec des règles analyses qui simulent le travail des analystes dans le processus de prise de décision, ce mémoire contient une partie théorique sur les entrepôts de données passifs et actifs, la conception automatisée des entrepôts de données, puis une partie pratique (conception et réalisation).

L'objectif de notre travail consiste à réaliser un système d'aide à la conception des entrepôts de données actifs qui regroupe deux parties, une implémentation d'une approche de conception des entrepôts de données passifs basée sur un algorithme de comparaison des clés étrangères, et la deuxième partie concerne le développement d'un système de génération des règles ECA dynamique.

Mots-clés : Entrepôts de données, Entrepôts de données, Règle d'analyse, règles ECA.

ABSTRACT

Active data warehouses are a new category of decision support systems, which offer the possibility to automate decision making. Active data warehouses extend traditional data warehouses with analytical rules that simulate the work of analysts in the decision making process, this dissertation contains a theoretical part on passive and active data warehouses, the automated design of data warehouses, and a practical part (design and implementation).

The objective of our work is to realize a system of assistance to the design of active data warehouses which gathers two parts, an implementation of an approach of design of passive data warehouses based on an algorithm of comparison of the foreign keys, and the second part concerns the development of a system of generation of the dynamic ECA rules.

Keywords :Data warehouse, Active data warehouse, Analysis rule, ECA rules.