

# Projet de fin D'étude

*Effectué à*



*Préparé par*

**Mohamed FRIKHA**

**Ismail ABIDA**

*En vue de l'obtention du diplôme National*

**D'INGÉNIEUR EN GÉNIE INFORMATIQUE**

*Intitulé :*

---

**Mise en place d'une solution d'automatisation de  
la migration d'un cube Oracle en cube Microsoft**

---

*Soutenu le 12 Juillet 2016, devant la commission d'examen :*

**M. Ahmed KALLEL**

**Président**

**M. Ahmed JEMAL**

**Rapporteur**

**M<sup>me</sup>. Manel BEN SALAH**

**Encadrante académique**

**M. Hatem AROUS**

**Encadrant Industriel**

# **Avant-Propos**

Dans le cadre de l'obtention de notre diplôme national d'ingénieur en génie informatique au sein de l'institut international de technologie de Sfax, nous avons été appelés à réaliser un mémoire de fin d'étude afin de clôturer notre formation du deuxième cycle universitaire.

C'est ainsi que nous projetons concrétiser nos connaissances acquises durant nos études supérieures par la conception et le développement d'une solution d'automatisation de la migration d'un cube Oracle en cube Microsoft.

# *Dédicaces*

*De plus profond de mon cœur je dédie ce fructueux travail*

*A ma chère mère Neila*

*Pour la grande joie que tu attends impatiemment de ma réussite. Rien n'égale ta tendresse, ton amour et tes sentiments d'affection dont tu n'as cessé de me prodiguer durant les plus durs moments de la préparation de ce travail. Que dieu te garde et te protège.*

*A mon cher père Sami*

*Pour tout l'amour avec lequel il m'a entouré, son encouragement moral durant mes études, sa diplomatie et son esprit compréhensif au moment difficile. Il a été pour moi le meilleur gage de ma réussite. Que dieu te protège et te procure bonne santé et longue vie.*

*A mon frère Moez*

*Pour le soutien moral et l'aide qu'il m'a apporté. Qu'il trouve dans ce travail l'expression de ma grande affection et un témoignage de ma profonde gratitude. Que la vie lui offre ce qu'elle a de mieux,*

*A mes chers amis*

*Je n'oublierais jamais votre présence à mes côtés durant la préparation de ce projet. Sans oublier mon binôme Ismail pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

*A tout ce qui m'ont encouragé*

*Pour leur encouragement illimité, leur tendresse et leur soutien moral.*

*Mohamed Frikha*

# *Dédicaces*

*Je dédie ce modeste travail et ma profonde gratitude*

*A mes chers parents Mohamed et Souhaila*

*Qui m'ont toujours soutenu et qui ont fait tout possibles pour m'aider et sans qui je ne serai pas là aujourd'hui. J'implore Dieu le plus puissant pour m'aider à vous prouver mes reconnaissances pour vos sacrifices afin de me guider vers le chemin de la connaissance et du savoir.*

*A ma sœur Hajer et son Mari Riadh*

*Pour leurs encouragements et la motivation qu'ils m'ont donnés. Que Dieu vous protège et vous procure une bonne santé et une longue vie.*

*A mon Petit Frère Alys*

*Présent dans tous mes moments d'examens. Que Dieu te gratifie le succès et le bonheur dans ta vie.*

*A la petite princesse Ranim*

*Dont nous sommes très heureux de sa naissance. Que Dieu te bénisse et te protège.*

*A mes chers amis et collègues*

*Je n'oublie jamais les moments magnifiques passés ensemble. Je leur souhaite une bonne continuation professionnelle. Sans oublier mon binôme Mohamed pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

*À tous ceux que j'aime, tous ceux qui m'aiment et qui me sont chers.*

*Que dieu vous bénisse tous*

*Ismail Abida*

# Remerciements

*Nous tenons tout d'abord à remercier notre encadrante **Madame Manel BEN SALAH**, technologue en informatique à l'institut Supérieur des Etudes Technologiques de Kairouan, pour sa disponibilité, ses conseils et sa collaboration pour la réalisation et la direction de ce travail.*

*Nous adressons nos sincères remerciements à **Monsieur Hatem AROUS**, gérants de la société « **TODEOS** », pour son accueil, sa disponibilité et son aide à la réalisation de ce projet.*

*Par la même occasion, nous remercions tous les personnels de la société «**TODEOS**».*  
*Nous tenons aussi à remercier vivement **Monsieur Ahmed KALLEL** et **Monsieur Ahmed JEMAL** qui ont bien voulu accepter de juger ce modeste travail.*

*Nous remercions également tous nos enseignants et le cadre administratif de l'institut international de la technologie de Sfax pour leur contribution à notre formation durant nos études universitaires.*

*Nous ne manquerons pas de remercier toutes les personnes qui nous ont encouragé et aidé de près ou de loin à l'élaboration de ce travail.*

**Mohamed & Ismail**

# Acronymes

<b>DSI</b>	Direction des Systèmes d'Information
<b>BI</b>	Business Intelligence
<b>ETL</b>	Extract Transform Load
<b>SSIS</b>	SQL Server Integration Services
<b>SSAS</b>	SQL Server Analysis Services
<b>SSRS</b>	SQL Server Reporting Services
<b>SSMS</b>	SQL Server Management Studio
<b>SSDT</b>	SQL Server Data Tools
<b>UML</b>	Unified Modeling Language
<b>OLAP</b>	On-Line Analytical Processing
<b>OLTP</b>	On-Line Transaction Processing
<b>3NF</b>	Third Normal Form
<b>EPM</b>	Enterprise Performance Management
<b>XMLA</b>	Extensible Markup Language for Analysis
<b>ESSBASE</b>	Extended SpreadSheet dataBASE
<b>SGBDR</b>	Système de Gestion de Base de Données Relationnel
<b>SQL</b>	Structured Query Language
<b>BIDS</b>	Business Intelligence Développement Studio
<b>T-SQL</b>	Transact Structured Query Language
<b>CRM</b>	Customer Relationship Management
<b>ERP</b>	Entreprise Resource Planning

# Table des matières

<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>CHAPITRE 1: PRESENTATION DU CADRE DE PROJET .....</b>	<b>3</b>
1 INTRODUCTION.....	3
2 PRESENTATION DE L'ORGANISME D'ACCUEIL.....	3
3 PRESENTATION DU PROJET .....	4
3.1 <i>Cadre du projet</i> .....	4
3.2 <i>Problématiques</i> .....	4
3.3 <i>Objectifs du projet</i> .....	4
4 LA DEMARCHE SUIVIE.....	5
4.1 <i>Définition de l'approche itérative et incrémentale</i> .....	5
4.2 <i>Les avantages attendus des itérations</i> .....	6
4.3 <i>Planning prévisionnel</i> .....	7
5 CONCLUSION .....	7
<b>CHAPITRE 2: ETAT DE L'ART .....</b>	<b>9</b>
1 INTRODUCTION.....	9
2 LES CONCEPTS DE BASE DE L'INFORMATIQUE DECISIONNELLE.....	9
2.1 <i>L'informatique décisionnelle</i> .....	9
2.2 <i>La chaîne décisionnelle</i> .....	9
2.3 <i>Stockage des données</i> .....	11
2.4 <i>Caractéristiques des données</i> .....	11
2.5 <i>Cube</i> .....	12
2.6 <i>Les schémas multidimensionnels</i> .....	13
3 LES TECHNOLOGIES DE L'INFORMATIQUE DECISIONNELLE.....	15
3.1 <i>Microsoft Business Intelligence</i> .....	15
3.2 <i>Oracle Hypérion Business Intelligence</i> .....	17
3.3 <i>Autres technologies</i> .....	18
4 ETUDE DE L'EXISTANT .....	19
4.1 <i>Description de l'existant : Migration Manuelle</i> .....	19
4.2 <i>Critique de l'existant</i> .....	19
4.3 <i>Les solutions disponibles</i> .....	19
4.4 <i>Solution proposée</i> .....	20
5 CONCLUSION .....	21
<b>CHAPITRE 3: SPECIFICATION DES BESOINS ET CONCEPTION .....</b>	<b>22</b>

1	INTRODUCTION.....	22
2	SPECIFICATION DES BESOINS.....	22
2.1	<i>Besoins fonctionnels.....</i>	22
2.2	<i>Besoins non fonctionnels.....</i>	22
2.3	<i>Diagramme de cas d'utilisation Globale .....</i>	23
3	CONCEPTION DETAILLEE .....	25
3.1	<i>Présentation UML .....</i>	26
3.2	<i>Conception générale de l'application.....</i>	26
3.3	<i>Conception de la phase d'intégration des données.....</i>	28
3.4	<i>Conception de la phase d'analyse des données .....</i>	31
4	CONCLUSION .....	32
	<b>CHAPITRE 4: REALISATION.....</b>	<b>33</b>
1	INTRODUCTION.....	33
2	ENVIRONNEMENT DE REALISATION .....	33
2.1	<i>Environnement matériel.....</i>	33
2.2	<i>Environnement logiciel.....</i>	33
2.3	<i>Langages de programmation.....</i>	35
2.4	<i>Techniques de développement.....</i>	35
3	VUE ARCHITECTURALE DU SYSTEME .....	36
4	PRESENTATION DES INTERFACES GRAPHIQUES DE L'APPLICATION .....	36
4.1	<i>Interface d'accueil.....</i>	36
4.2	<i>Interface de connexion à Oracle Hypérion .....</i>	37
4.3	<i>Création de l'entrepôt de données.....</i>	38
4.4	<i>Création du cube SSAS.....</i>	41
4.5	<i>Résultat attendu de notre application .....</i>	44
5	CONCLUSION .....	44
	<b>CONCLUSION GENERALE .....</b>	<b>45</b>
	<b>BIBLIOGRAPHIE .....</b>	<b>46</b>
	<b>WEBOGRAPHIE.....</b>	<b>46</b>
	<b>ANNEXE : SOLUTION SSIS .....</b>	<b>48</b>



# Liste des tableaux

<b>Tableau 1.</b> <i>Description textuelle du cas d'utilisation « Créer l'entrepôt de données »</i> .....	24
<b>Tableau 2.</b> <i>Description textuelle du cas d'utilisation « créer le cube Microsoft»</i> .....	25
<b>Tableau 3.</b> <i>Description textuelle du cas d'utilisation « Consulter l'historique de conversion »</i> .....	25
<b>Tableau 4.</b> <i>Configuration Techniques des Ordinateurs</i> .....	33

# Liste des figures

<b>Figure 1.</b> <i>Concept général de l'application</i> .....	5
<b>Figure 2.</b> <i>Le cycle itératif incrémental</i> .....	6
<b>Figure 3.</b> <i>Diagramme de GANTT de notre projet</i> .....	7
<b>Figure 4.</b> <i>La chaîne décisionnelle</i> .....	10
<b>Figure 5.</b> <i>Exemple de table de faits Vente</i> .....	12
<b>Figure 6.</b> <i>Exemples de table de dimensions</i> .....	12
<b>Figure 7.</b> <i>Hiérarchie de la dimension Date</i> .....	13
<b>Figure 8.</b> <i>Schéma en étoile comprenant une table de faits et trois tables de dimension</i> .....	14
<b>Figure 9.</b> <i>Schéma en flocon</i> .....	14
<b>Figure 10.</b> <i>Schéma en constellation</i> .....	15
<b>Figure 11.</b> <i>La Plateforme MS BI</i> .....	16
<b>Figure 12.</b> <i>L'architecture d'Oracle Hypérion EPM 11.1.2.3</i> .....	18
<b>Figure 13.</b> <i>Diagramme de cas d'utilisation Globale</i> .....	24
<b>Figure 14.</b> <i>Diagramme de séquence</i> .....	27
<b>Figure 15.</b> <i>Diagramme de navigation</i> .....	28
<b>Figure 16.</b> <i>Diagramme de flux de contrôle de l'entrepôt de données</i> .....	29
<b>Figure 17.</b> <i>Diagramme d'activité de package de création des tables de dimensions</i> .....	30
<b>Figure 18.</b> <i>Diagramme d'activité de package de création de table de fait</i> .....	31
<b>Figure 19.</b> <i>Diagramme de séquence « création d'un cube SSAS »</i> .....	32
<b>Figure 20.</b> <i>Architecture globale du système</i> .....	36
<b>Figure 21.</b> <i>L'interface d'accueil</i> .....	37
<b>Figure 22.</b> <i>Configuration de la connexion à Oracle Hypérion EPM</i> .....	37
<b>Figure 23.</b> <i>Configuration de la connexion au serveur moteur de la base de données</i> .....	38
<b>Figure 24.</b> <i>Création de l'entrepôt de données</i> .....	39
<b>Figure 25.</b> <i>Conversion des dimensions</i> .....	40
<b>Figure 26.</b> <i>L'état de progression de l'exécution du flux SSIS</i> .....	40
<b>Figure 27.</b> <i>Conversion de(s) fichier(s) de fait</i> .....	41
<b>Figure 28.</b> <i>Interface de configuration de connexion</i> .....	42
<b>Figure 29.</b> <i>Interface de création du cube</i> .....	43
<b>Figure 30.</b> <i>Interface de traitement du cube</i> .....	43
<b>Figure 31.</b> <i>Cube Microsoft finale</i> .....	44

# Introduction Générale

De nos jours, l'informatique décisionnelle est un sujet en pleine évolution, s'adressant à la direction générale tout comme aux métiers. Outil d'aide à la décision, l'informatique décisionnelle permet d'avoir une vue d'ensemble des différentes activités de l'entreprise, et de son environnement. Cette vue transversale nécessite de connaître les différents métiers de l'entreprise et implique certaines spécificités organisationnelles et managériales.

La mise en place de projets Business Intelligence ne peut se faire sans avoir défini préalablement une stratégie décisionnelle globale de l'entreprise elle-même. Plus que pour des projets système d'information classiques, les projets BI nécessitent une grande maturité dans les relations DSI-Métiers, s'inscrivent dans une démarche d'amélioration continue. De ce fait, les stratégies des sociétés de services en BI est de répondre au plus près aux demandes et besoins de leurs clientèles.

Aujourd'hui, le marché BI propose des solutions assez complètes concernant les aspects de consolidation, d'analyse et de rapports des données, tant du domaine propriétaire que de l'open source. Les évolutions possibles, à court-moyen terme, porteraient sur les outils d'analyse proactive et de simulation, ainsi que sur l'interactivité et la convivialité des accès aux données, et sur la combinaison de données structurées et non structurées issues des sources de données internes et externes. Mais le manque de visibilité des clients sur le marché BI crée une grande incertitude quant à la pérennité des offres actuelles et nuit grandement à la définition d'une stratégie décisionnelle à moyen-long terme concernant le choix de la technologie BI utilisé.

C'est dans ce contexte que se situe notre projet de fin d'étude. Nous visons à automatiser la migration des cubes entre la technologie Oracle Hypérion Essbase vers la technologie Microsoft SQL Server Analysis Services et ceci en développant une plateforme de migration qui touche ces deux technologies BI concurrentes.

Le présent rapport s'articule autour de quatre chapitres décrivant notre travail en détail.

Nous commençons par la présentation du cadre du travail, les problématiques reliées au sujet traité et la méthodologie utilisée pour réaliser ce projet.

Dans un deuxième temps, nous représentons l'état de l'art du sujet. En fait, dans cette partie nous décrivons les concepts de base de l'informatique décisionnelle. Nous nous intéressons à la description et la critique des solutions existantes ainsi qu'aux solutions proposées.

Ensuite, nous allons spécifier les besoins fonctionnels et non fonctionnels de notre future solution, nous illustrons les différents diagrammes conceptuels en utilisant le langage de modélisation UML tout en expliquant notre choix.

Finalement, nous clôturons ce rapport par un dernier chapitre qui décrit la réalisation de notre solution toutes en présentant les interfaces développées ainsi que les fonctionnalités associées.

# Présentation du cadre de projet

## 1 Introduction

Dans ce chapitre, nous exposons le cadre général du projet. Nous allons commencer en premier lieu par la présentation de l'organisme d'accueil, le travail demandé et nous terminons par l'explication de la démarche de travail que nous allons suivre.

## 2 Présentation de l'organisme d'accueil

TODEOS NEARSHORE est une société de conseil et de services spécialisée dans l'externalisation, la mise en œuvre, l'intégration et le suivi des projets dans les nouvelles technologies et les systèmes d'information.

Cette société se spécialise notamment en technologie des systèmes d'informations décisionnelles et en développement d'applications et de logiciels. De plus, elle offre à ses clients des formations assurées par une équipe d'experts. [1]

TODEOS offre à ses clients plusieurs services riches tel que :

- L'assurance de la compréhension et la définition des besoins du client,
- L'accompagnement de ses clients dans la mise en œuvre et l'exploitation de leurs systèmes d'informations,
- L'intervention sur la totalité du cycle de vie des systèmes informatiques : études, réalisations, exploitation, maintenance,
- La donation des recommandations de grandes orientations opérationnelles et technologiques en cohérence avec la stratégie de l'entreprise.

### **3 Présentation du projet**

#### **3.1 Cadre du projet**

Ce stage s'inscrit dans le cadre d'un projet de fin d'étude pour l'obtention du diplôme d'ingénieur, spécialité génie logiciel et informatique décisionnelle.

Le sujet est intitulé « Mise en place d'une solution d'automatisation de la migration d'un cube ORACLE en cube MICROSOFT ». En effet, à travers cette solution nous allons automatiser la migration des cubes ORACLE en cubes MICROSOFT.

Ce projet était proposé par la société TODEOS NEARSHORE qui nous a intégrés au sein de son équipe afin de se rapprocher de ses ingénieurs et se familiariser avec la vie professionnelle.

#### **3.2 Problématiques**

Actuellement, les développeurs de l'entreprise sont menés à réaliser manuellement des opérations de migration des cubes oracles vers des cubes Microsoft.

Pour réaliser la migration, les développeurs doivent exporter les cubes de l'Oracle Hypérion sous formes des fichiers de données. Ils traitent ces données afin de construire un entrepôt de données.

Les données traitées généralement sont massives et de grande taille, ce qui implique une grande perte de temps, la plupart des cubes à traiter sont dans le domaine de comptabilité et finance qui sont relativement complexes. Une conversion manuelle peut facilement prendre trois à quatre personnes par mois pour un seul cube.

De plus, le besoin analytique est traité aussi d'une manière manuelle. Cette stratégie de travail réduit la productivité de la société et ne garantit pas des décisions adéquates et précises.

Pour remédier à ces problèmes, la société TODEOS NEARSHORE souhaite implémenter une solution d'automatisation de la migration d'un cube Oracle en cube Microsoft.

#### **3.3 Objectifs du projet**

Nous présentons dans ce qui suit les objectifs à atteindre pour maximiser la performance globale de notre future solution. Ces objectifs se résument dans les points suivants :

- Automatiser le processus de migration d'un cube Oracle vers un cube Microsoft,
- Mise en place d'une application simple qui joue le rôle d'un intermédiaire entre l'utilisateur final et les outils BI proposées par la technologie Microsoft,
- Accélérer l'opération de migration toutes en diminuant les coûts et les délais d'exécution,
- Réalisation d'une charte graphique qui assure une cohérence, une harmonisation dans toutes les étapes nécessaires pour assurer la migration entre ses deux différentes technologies,
- Une mise à disposition de nouveaux usages avec Microsoft BI.

La figure 1 représente le concept général de l'application. Elle est considérée comme une boîte noire qui prend en input un cube Oracle Hypérion Essbase pour offrir à l'utilisateur un cube "Microsoft Analysis services".



**Figure 1.** *Concept général de l'application*

#### **4 La démarche suivie**

Étant donné la nature de notre projet et ses objectifs qui sont bien définis, clairs et fixés dès le début, les méthodes classiques s'avèrent inadaptées. De ce fait, la gestion de ce type de projet requiert une flexibilité facile avec le changement et détection des bugs le plus tôt possible afin de minimiser l'impact de ses derniers sur notre projet. Par conséquent, nous opterons pour la méthode de développement « Processus unifié » qui est basée sur un cycle de vie itératif et incrémental.

Dans la suite, nous définirons l'approche utilisée dans notre projet, puis nous allons mentionner ses avantages.

##### **4.1 Définition de l'approche itérative et incrémentale**

Le développement d'un produit logiciel destiné à la commercialisation est une vaste opération qui peut s'étendre sur plusieurs mois, voire sur une année. Il est nécessaire de découper le travail en plusieurs parties qui sont autant de mini-projets (Concept systémique de système et sous-

systèmes). Chacun d'eux représente une itération qui donne lieu à un incrément. Les itérations désignent des étapes de l'enchaînement d'activités, tandis que les incréments correspondent à des stades de développement du produit. [2]

La figure 2 présente une itération. Elle comporte six étapes, dont quatre qui en constituent le cœur.

- **L'expression des besoins** : C'est la définition et la formalisation des besoins opérationnels attendus afin de réaliser un système qui répond parfaitement à ces besoins,
- **Le cœur du processus itératif** :
  - **Spécification** : C'est la traduction en langage technique des besoins fournis en entrée,
  - **Développement** : Il s'agit de la réalisation concrète de ce qui a été défini,
  - **Validation** : C'est l'ensemble des tests qui permettent de s'assurer que le développement effectué correspond bien à ce qui était attendu,
  - **Évaluation** : Cette étape sert à effectuer un retour sur les écueils rencontrés et les fonctionnalités abandonnées pendant les 3 étapes précédentes, et l'utiliser comme informations d'entrée pour un nouveau cycle,
- **Déploiement** : Les livrables qui ont été validés sont déployés pour que le client y ait accès,



**Figure 2.** *Le cycle itératif incrémental*

#### 4.2 Les avantages attendus des itérations

L'approche de développement par itérations offre les avantages suivants :

- Elle s'adapte mieux aux changements. En fait, cette approche considère le changement comme faisant partie du cycle de développement d'une application et non pas comme un événement intempestif,

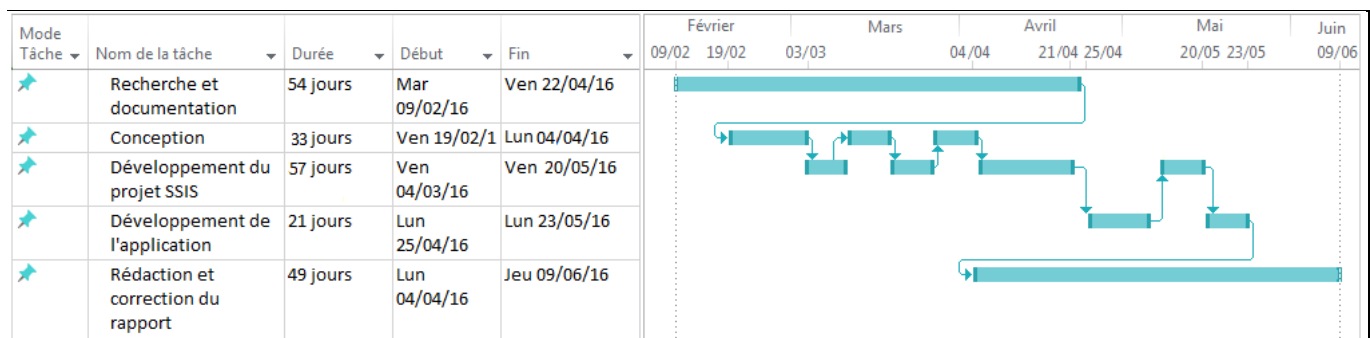


- Elle nous permet de détecter les risques très tôt dans la vie du projet,
- Elle permet d'ajuster les choix en termes d'architecture ou de conception graphique par exemple, très tôt dans le processus et non pas après que ces derniers aient été complètement réalisés,
- Chaque itération est une expérience qui nous permet d'apprendre d'avantage sur les challenges que représente le projet. Par exemple, il est fréquent de revoir les estimations faites au début du projet après la fin des premières itérations,
- On donne la chance au client de visualiser le résultat des itérations et donc l'occasion pour lui d'exprimer des ajustements au fur et à mesure que le projet avance et non pas à la fin uniquement lors des tests d'acceptation,
- Le client est rassuré car il peut voir concrètement la progression du projet à travers la manipulation ou l'exécution de cas d'utilisations réelles de son produit. [3]

### 4.3 *Planning prévisionnel*

Afin de réaliser notre projet dans les délais établis par la convention de stage, il a fallu définir les étapes essentielles et estimer le temps à consacrer pour chacune. Pour cela, nous avons utilisé l'outil de gestion des projets MS Project afin de représenter les différentes tâches et établir par la suite le diagramme de GANTT pour visualiser le planning prévisionnel.

La figure 3 présente le diagramme de GANTT de notre projet.



**Figure 3.** *Diagramme de GANTT de notre projet*

## 5 Conclusion

Dans ce chapitre introductif, nous avons présenté le contexte général dans lequel s'inscrit notre projet. Ce qui a permis de souligner les problématiques qui en découlent, de proposer une solution adéquate et de préciser la méthodologie adoptée.

Nous allons entamer dans le chapitre suivant les principaux concepts et les outils de l'informatique décisionnelle ainsi que la méthodologie de travail utilisée pour réaliser notre solution.

# Chapitre 2

## Etat de l'art

### 1 Introduction

L'état de l'art est la phase initiale dans les modèles du cycle de vie de développement des applications ou dans les recherches scientifiques. En effet, cette phase permet d'exprimer les spécifications et les outils reliés à ce domaine.

Dans ce qui suit, nous allons présenter les concepts de base de l'informatique décisionnelle, les principaux outils de BI qui sont reliés avec notre projet et la méthodologie de travail utilisée pour réaliser notre solution.

### 2 Les concepts de base de l'informatique décisionnelle

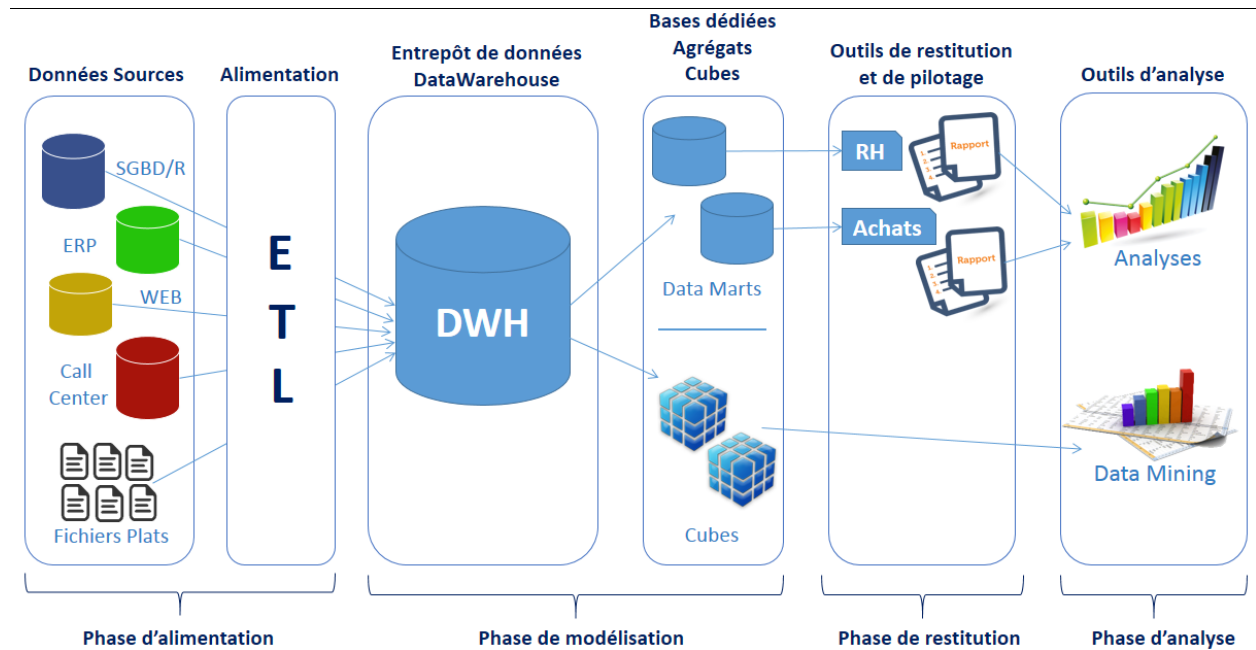
#### 2.1 *L'informatique décisionnelle*

L'informatique décisionnelle, (en anglais "Business Intelligence" ou BI), désigne un ensemble de méthodes, de moyens et d'outils informatiques utilisés pour piloter une entreprise et aider à la prise de décision : tableaux de bord, rapports analytiques et prospectifs.

Elle repose à la fois sur la collecte, la modélisation et la restitution des données éparses, déstructurées et hétérogènes que génère une entreprise : archives papier, bases de données, feuilles de calcul, données clients collectées via un service en ligne, etc. Le tout est traité par des outils d'extraction, de transfert et de consolidation mis en place pour normaliser ces sources et établir une cohérence entre elles. [4]

#### 2.2 *La chaîne décisionnelle*

Comme le montre la figure 4, la chaîne décisionnelle comprend différentes phases :



**Figure 4.** La chaîne décisionnelle

- **La phase d'alimentation :** Cette phase va faire intervenir le processus ETL (Extraction, Transformation et Chargement). L'ETL est un composant qui sert à alimenter un "datawarehouse" ou un système décisionnel. Il se base sur la collecte des données à partir des bases de données de production (ERP, CRM, SGBDR, fichiers plats,...). Ces données hétérogènes seront transformé (conversion, nettoyage, standardisation,...) et charger dans un entrepôt de données,
- **La phase de modélisation :** Cette phase permet de stocker les données nettoyées dans des bases de données dédiées à l'analyse appelées entrepôts de données. Un entrepôt de données est une collection de données thématiques, intégrées, non volatiles et historisées pour la prise des décisions. Ces données seront représentées sous forme des cubes OLAP multidimensionnels prévus à des fins d'analyses interactives,
- **La phase de restitution :** C'est dans cette phase que les différents outils de restitution vont intervenir. Nous trouverons des outils de rapports tels que BIRT et Crystal Reports, des portails d'accès à des tableaux de bord tels que Power BI et Tableau et des outils de classification et de navigation dans les cubes tels que tableaux croisés dynamiques,
- **La phase d'analyse :** C'est dans cette phase que les utilisateurs finaux interviennent et analysent les informations qui leurs sont fournies. Elle peut aussi faire intervenir des spécialistes en analyse pour utiliser des outils de statistique et ressortir des prévisions ou des estimations futures. Parmi ces outils, on peut citer : Clear Analytics et Board. [5]

## **2.3 Stockage des données**

Les bases de données de production ne sont pas utilisables pour une exploitation décisionnelle. Les données brutes ne sont pas prêtes à cet usage et les requêtes décisionnelles sont particulièrement gourmandes en ressources machines. C'est pour cette raison des nouvelles bases de données dédiées à l'analyse et les traitements décisionnels apparues.

### **2.3.1 Entrepôt de données**

Un entrepôt de données est une base de données dans laquelle sont recopiées des données opérationnelles, mais à laquelle aucune application opérationnelle n'accède. L'entrepôt de données (en anglais "datawarehouse") est le lieu de stockage intermédiaire des différentes données en vue de la constitution du système d'information décisionnel. Le créateur du concept de "datawarehouse", *Bill Inmon*, le définit comme suit : « Un "datawarehouse" est une collection de données thématiques, intégrées, non volatiles et historisées pour la prise de décision. »

### **2.3.2 Magasin de données**

Les entrepôts de données partielles, limités à un domaine particulier sont qualifiés de magasin de données (en anglais "datamart"). Les magasins de données, peuvent être considérés comme des sous référentiels de données ou vus par métier de l'entreprise. Cet éclatement permet aux outils d'analyse d'accéder plus facilement aux données.

## **2.4 Caractéristiques des données**

### **2.4.1 Orienté sujet**

Les données sont organisées pour répondre à des besoins métiers : elles sont classées par thèmes. Les données propres à un thème, les ventes par exemple, seront rapatriées des différentes bases OLTP "On Line Transaction Processing" de production et regroupées.

### **2.4.2 Intégré**

Les données proviennent de sources hétérogènes utilisant chacune un type de format. Elles sont intégrées avant d'être proposées à l'utilisation. Un nettoyage préalable des données est nécessaire dans un souci de rationalisation et de normalisation.

### **2.4.3 Non volatile**

Les données une fois intégrées dans l'entrepôt ne sont pas appelées à être supprimées.

#### 2.4.4 Historisé

Les données mémorisées sont datées. Il est ainsi possible de visualiser leur évolution dans le temps. Le degré de détail de l'archivage est bien entendu relatif à la nature des données. Toutes les données ne méritent pas d'être archivées.

### 2.5 Cube

Un cube est une structure multidimensionnelle composée par des tables de fait et des tables de dimensions qui contiennent des informations à des fins analytiques.

#### 2.5.1 Les faits

Une table de faits est constituée d'un ensemble de fait qui désigne soit une référence (clés étrangères) vers une table de dimension soit une mesure. Une mesure représente une valeur numérique à analyser. La figure 5 présente une table de faits nommée **Vente** qui contient deux mesures nommées **quantité** et **montant**. [6]

Vente
-Quantité
-Montant

**Figure 5.** Exemple de table de faits Vente

#### 2.5.2 Les dimensions

Une table de dimensions contient des différents axes d'analyse qui nous permettent d'étudier une mesure selon des différentes perspectives. La figure 6 montre des exemples de table de dimensions.

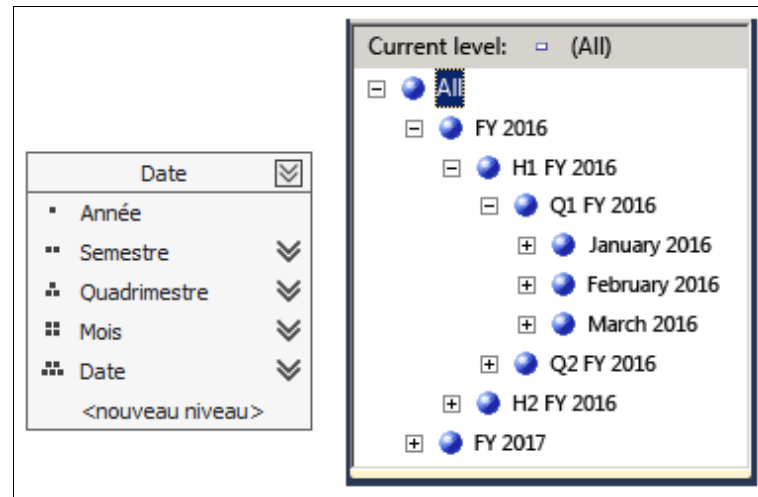
Temps	Géographie	Catégorie
-Année	-Région	-TypeProd
-Trimestre	-Département	-Gamme
-Saison	-Ville	-NomProd
-Mois		-Couleur
-Jour		

**Figure 6.** Exemples de table de dimensions

### 2.5.3 Hiérarchie

Les éléments d'une dimension peuvent être organisés en une hiérarchie, un ensemble de relations parent-enfant, généralement où un membre du parent résume ses enfants. Éléments parents peuvent en outre être agrégés comme étant enfant d'un autre parent.

Par exemple le mois de mars 2016 a comme parent le premier quadrimestre de l'année 2016, qui est à son tour l'enfant du premier semestre de l'année 2016. [7]



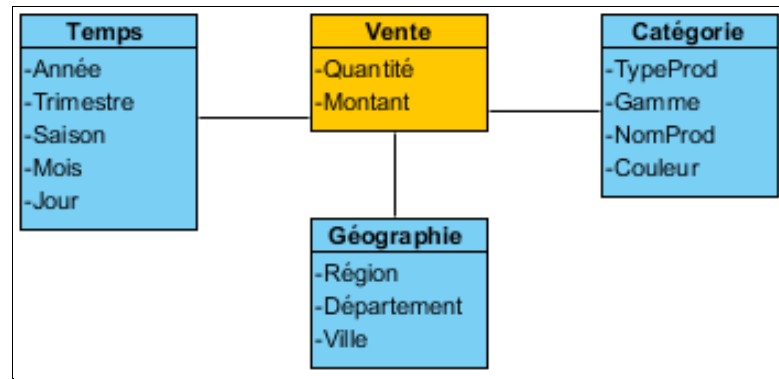
**Figure 7.** *Hiérarchie de la dimension Date*

## 2.6 Les schémas multidimensionnels

Les schémas multidimensionnels donnent des indications sur la manière de représenter les données en base. Il s'agit d'organiser les tables de manière logique, afin d'optimiser les requêtes.

### 2.6.1 Schéma en étoile

Le schéma en étoile est une façon de représenter un modèle multidimensionnel au sein d'une base de données. Comme la montre la figure 8, un schéma en étoile se constitue d'une table de faits associée à un ensemble de tables de dimension. Ces dernières sont toujours directement reliées à la table de faits.

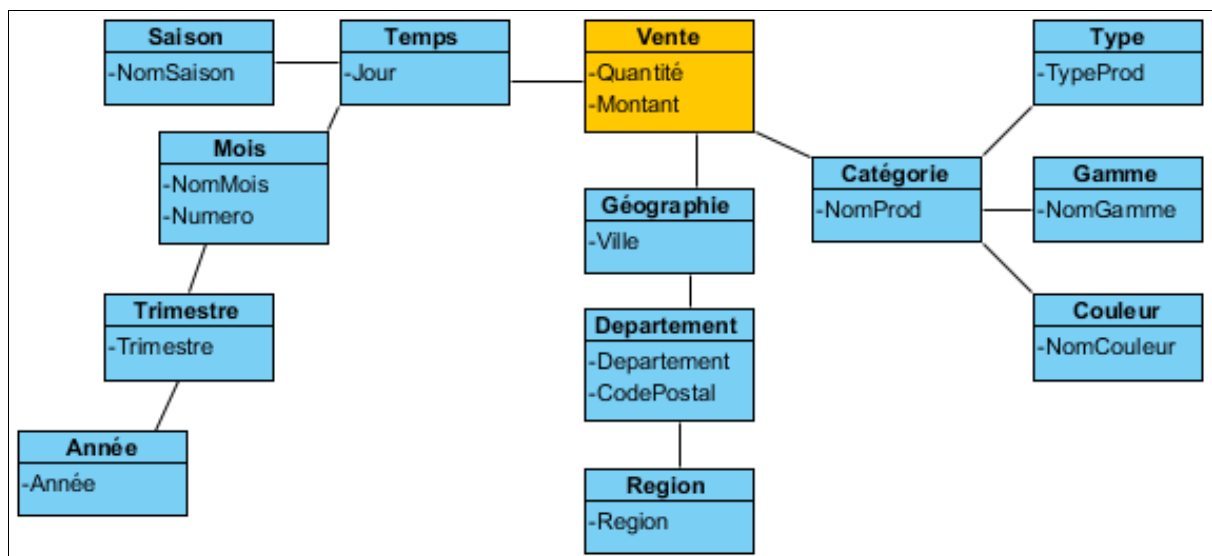


**Figure 8.** Schéma en étoile comprenant une table de faits et trois tables de dimension

### 2.6.2 Schéma en flocon

Un schéma en flocon est une représentation normalisée (3NF) d'une unique table de dimension. Cette table, qui peut parfois contenir un grand nombre d'attributs, est scindée en plusieurs entités.

La figure 9 reprend le modèle en étoile en décomposant les dimensions. [6]

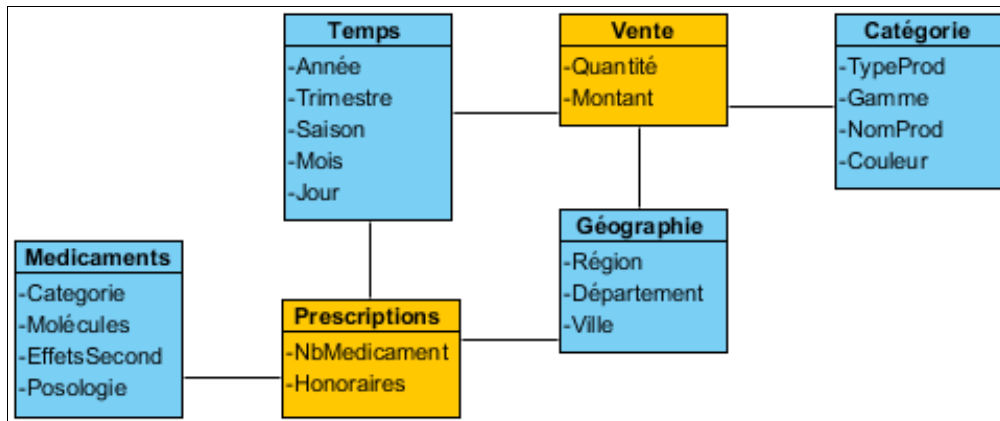


**Figure 9.** Schéma en flocon

### 2.6.3 Schéma en constellation

Le schéma en constellation est un ensemble de schémas en étoiles et/ou en flocons dans lesquels les tables de faits se partagent certaines tables de dimensions. La figure 10 présente un exemple de schéma en constellation, constitué par deux tables de fait **Vente** et **Prescriptions** qui partagent entre eux les deux dimensions **Temps** et **Géographie**. [6]





**Figure 10.** *Schéma en constellation*

### 3 Les technologies de l'informatique décisionnelle

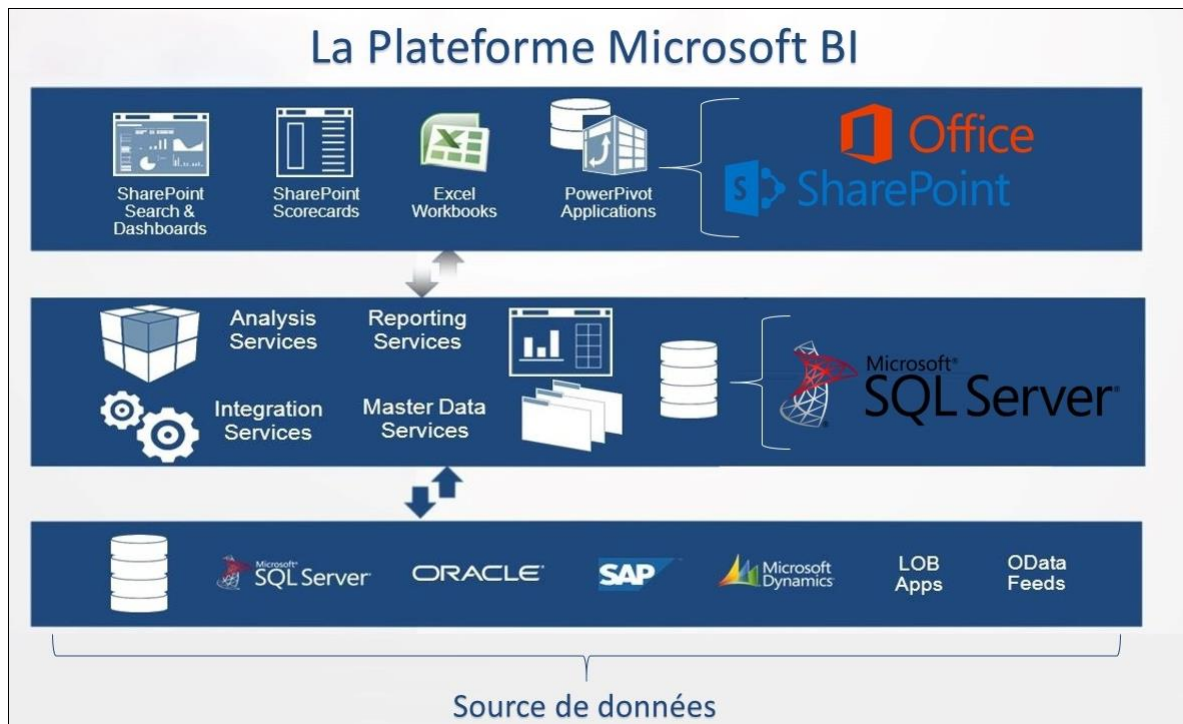
Il existe une multitude de solutions BI sur le marché, tel qu'IBM Cognos Business Intelligence, Information Builders WebFOCUS Platform, Outils BI de la gamme Microsoft, Oracle BI Foundation Suite, Outils BI de la gamme Infor, Qlik Sense, QlikView, SAP Business Objects Business Intelligence...

Nous nous focalisons, dans la partie qui suit, sur les suites Oracle et Microsoft dédiées aux développements d'applications BI.

#### 3.1 *Microsoft Business Intelligence*

Le Business Intelligence de Microsoft tire parti des investissements existants en termes de productivité (Office), d'infrastructure de données (SQL Server) et d'outils de collaboration (SharePoint) pour former une plateforme complète et performante aux utilisateurs. La solution de Business Intelligence intègre les forces des trois produits sur une plateforme unique qui permet aux utilisateurs de prendre les bonnes décisions pour l'entreprise.

La figure 11 présente la plateforme décisionnelle de la suite Microsoft Business Intelligence qui est présenté en trois couches : Intégration, gestion et présentation des données.



**Figure 11.** La Plateforme MS BI

Dans ce qui suit nous présentons le système de gestion de base de données SQL Server :

- **SQL Server** avec ses avantages en termes d'intégration et de stockage de grands volumes de données en toute sécurité : SQL Server rassemble toutes les données dans un moteur de gestion puissant et peut stocker un nombre impressionnant de données. Doté de services d'analyse et de reporting, il permet aux entreprises de gérer ses données en toute sécurité. Ce produit intègre l'ensemble des briques nécessaire à la construction d'un système d'information décisionnel. Les services associés au SQL Server sont :
  - **SSIS (SQL Server Integration Services)** : Il représente l'ETL (Extract, Transform and Load), c'est l'outil qui vous permet d'extraire les données de votre SI ou d'autres sources de données et de le charger dans l'infocentre.
  - **SSAS (SQL Server Analyses Services)** : L'objectif de cet outil est de créer des cubes OLAP. Cette technologie, que nous détaillons par la suite, structure une base de données multidimensionnelle. Ces dimensions permettent ensuite au décideur d'explorer instantanément des volumes importants de données et d'obtenir l'ensemble des vues agrégées qu'il souhaite.
  - **SSRS (SQL Server Reporting Services)** : cette brique offre un outil complet pour la réalisation des rapports et leur diffusion. Les rapports Reporting Services sont naturellement intégrables dans SharePoint pour en faciliter la diffusion. [8]

Les outils de développement intégrés à SQL Server sont :

- **SQL Server Data Tools (SSDT)** est un outil de conception, de développement de base de données moderne et de modélisation des données pour SQL Server, Azure SQL Database, les modèles de données Analysis Services, les packages “Integration Services” et les rapports “Reporting Services”. Avec SSDT, nous pouvons concevoir, développer, générer, tester et publier des bases de données et d’autres types de contenu avec la même facilité que celle des développements d’applications dans Visual Studio. [9]
- **SQL Server Management Studio (SSMS)** est un environnement intégré pour l'accès, la configuration, la gestion, l'administration et le développement de tous les composants de SQL Server. SSMS combine un large groupe d'outils graphiques avec un certain nombre de riches éditeurs de script pour fournir aux développeurs et aux administrateurs de tous les niveaux de compétence accès à SQL Server.

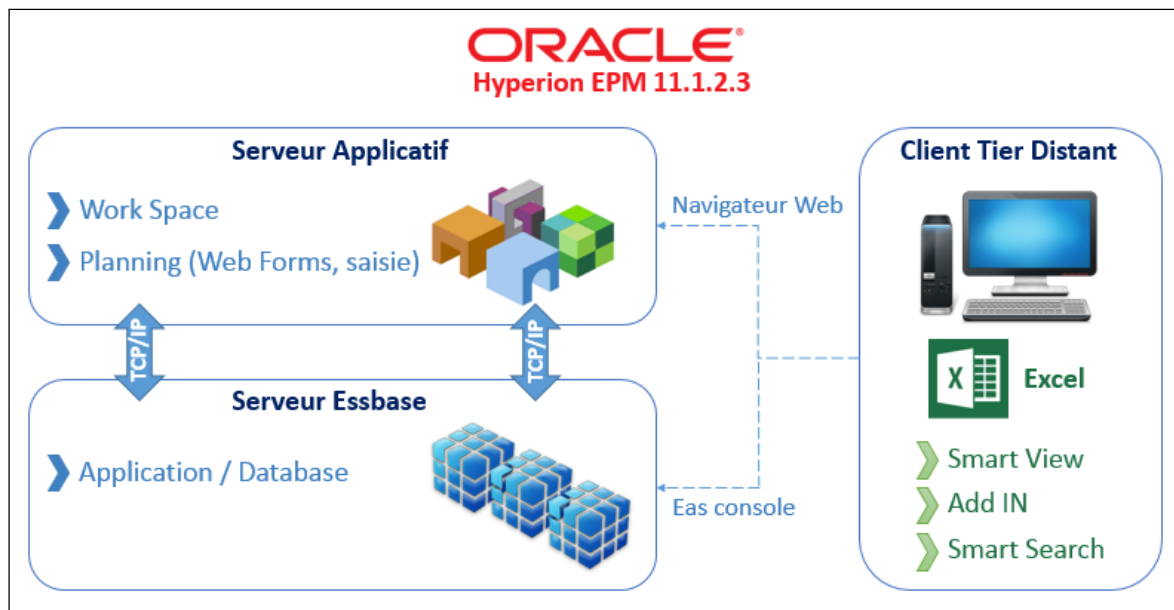
### 3.2 *Oracle Hypérion Business Intelligence*

Hypérion Solution, éditeur historique de la Business Intelligence, spécialiste des solutions de gestion financière et de planification a été racheté par Oracle Corporation en 2007. Il s’agissait alors pour ce dernier de compléter son offre afin de proposer l’ensemble de la chaîne décisionnelle en s’appuyant sur des produits matures et de qualité. Après ces opérations de concentration, les solutions technologiques d’Oracle couvrent l’ensemble du processus d’aide à la décision et de management de la performance.

Les principaux produits sont :

- **Oracle Essbase** : Serveur multidimensionnel de données, un produit historique en matière d’analyse dimensionnelle conçue à l’origine par Arbor Software avant d’être racheté par Hyperion Solution.
- **Oracle Enterprise Performance Management (EPM)** : Disponible aussi en mode “Cloud”, délivre les fonctions d’aide au pilotage stratégique, budget, planning, prévision, reporting, gestion des coûts. [10]

La figure 12 présente l’architecture d’Oracle Hypérion EPM 11.1.2.3.



**Figure 12.** *L'architecture d'Oracle Hypérion EPM 11.1.2.3*

### 3.3 Autres technologies

#### 3.3.1 Pentaho

Pentaho est une suite logicielle décisionnelle complète qui permet la création et la diffusion de documents décisionnels à un grand nombre de destinataires via une interface web. Pentaho est une solution open source entièrement développée en Java. Elle porte sur toute la chaîne décisionnelle et utilise différents outils et composants pour :

- La collecte et l'intégration : l'outil ETL Kettle (Pentaho Data Integration),
- La diffusion : un serveur d'application Pentaho BI Server, déployé sur un serveur d'application tel que JBoss ou Tomcat,
- La présentation : l'outil de design de rapports Pentaho Report Designer.[11]

#### 3.3.2 IBM Cognos

Le logiciel IBM Cognos offre toutes les fonctionnalités de l'informatique décisionnelle nécessaire pour répondre aux questions métiers essentielles : Requête, Analyse, Reporting, Tableaux de bord, Monitoring en temps réel. Les différentes fonctionnalités offertes par ce logiciel sont :

- Reporting : créer et diffuser tous types de rapports dont vous avez besoin de façon centralisée et accessibles en libre-service,
- Analyse : analyser les tendances et l'évolution de l'activité. Placer les résultats et décisions dans leur contexte,

- Tableaux de bord : surveiller, mesurer, et piloter les performances de l'entreprise d'un seul coup d'œil,
- Scorecarding : connecter la stratégie aux opérations. [12]

## **4 Etude de l'existant**

L'étude de l'existant permet de déterminer les points faibles et les points forts d'un produit actuel pour pouvoir déterminer les besoins du client, en vue d'en prendre compte dans la conception et la réalisation de l'application.

### **4.1 Description de l'existant : Migration Manuelle**

Actuellement, la migration entre les plateformes Oracle Hyperion Essbase et Microsoft BI s'effectue d'une façon manuelle. En fait, cette méthode de migration manuelle comporte deux principales phases. Une phase d'extraction et de nettoyage des données à partir de la plateforme Oracle Hyperion Essbase afin de sauvegarder ces dernières dans un entrepôt de données via les outils SSIS de SQL Server et une deuxième phase de création du cube SSAS via l'outil SQL Server Data Tools en se basant sur l'entrepôt de données déjà crée.

### **4.2 Critique de l'existant**

La méthode de migration manuelle, décrite plus haut, est une opération compliquée et pénible vu la taille énorme des données à traiter (la taille des entrepôts est à l'ordre de quelques To). Cette voluminosité des données implique aussi la longue durée du temps de traitement.

Nous notons aussi que cette méthode consiste à répéter les mêmes fastidieuses étapes pour chaque application déployée sous la plateforme Oracle à convertir en cube SSAS.

La création manuelle de l'entrepôt de données peut engendrer aussi des erreurs telles que l'altération de la structure de base d'un cube, des rapports erronés,...

### **4.3 Les solutions disponibles**

Après une recherche dans les solutions de migration des cubes disponible sur le marché et qui répond à notre besoin, nous avons constaté qu'il y a un seul logiciel ou outil de migration.

#### **4.3.1 CubePort <sup>TM</sup>**

Le portage d'application Analytics, est la seule solution disponible sur le marché pour les clients migrants ou de réplication à partir d'Oracle / Hyperion Essbase vers Microsoft SQL Server

Analysis Services. Le logiciel emploie les pratiques pour convertir presque tous les aspects d'une application Essbase / cube pour le produit Microsoft. CubePort <sup>TM</sup> utilise sa technologie de cartographie analogie en attente de brevet pour réaliser une opération de migration facile et propre. Depuis son introduction, CubePort a gagné les sceptiques et dépassé les attentes des adoptants éventuels. [13]

Ce produit est lancé en 2005, depuis le 31 mars 2010 sa société propriétaire « ExoLogic Corporation » a stoppé tout genre d'évolution pour ce produit. Actuellement, il n y a aucune version téléchargeable disponible sur internet. Parmi les inconvénients de ce produit :

- Des interfaces non ergonomiques et non compréhensibles,
- La documentation est inefficace et non claire pour les utilisateurs novices au domaine de l'informatique décisionnelle,
- Pour un nombre réduit des cubes à migrer, le prix de l'opération de migration serait 5000 \$ pour chaque cube, mais avec un nombre élevé de cubes, le prix par cube peut être aussi peu que 1000 \$ par cube,
- Ce produit ne répond pas exactement aux besoins de notre client.

#### **4.4 Solution proposée**

Notre solution est de mettre en place une application d'automatisation de la migration d'un cube Oracle Hypérion Essbase vers un Cube Microsoft Analysis services qui offre à ses utilisateurs des services richement cohérents et qui emploie les meilleurs pratiques pour convertir presque tous les aspects d'une application Essbase (ou cube pour le produit Microsoft).

Cette solution traite pratiquement toutes les caractéristiques d'une application Essbase telles que les agrégations, hiérarchies, membre calculé ...

Après quelques réunions avec le staff technique, nous finirons par identifier les principaux points de réalisation à prendre en compte :

- Création d'une application simple à utiliser par l'utilisateur final,
- Développement d'une solution qui nous permet d'extraire les données d'une Base de données source, les traiter et les charger dans une Base de données dédié à l'analyse,
- Création des cubes Microsoft.

## **5 Conclusion**

Dans ce chapitre, nous avons présenté les principaux concepts de base de l'informatique décisionnelle. Ensuite, nous avons décrit les différentes technologies de l'informatique décisionnelle et élaborer une étude et une critique de l'existant qui nous ont conduit à proposer notre solution.

Dans le prochain chapitre, nous définissons les spécifications des besoins et présentons la modélisation conceptuelle de notre future solution.

# Chapitre 3

## Spécification des besoins et conception

### 1 Introduction

L'identification des besoins du projet est l'une des étapes importantes dans le cycle de vie d'un logiciel. Elle mène à la détermination des besoins fonctionnels et non fonctionnels avec lequel les utilisateurs s'interagissent avec le système.

Nous consacrerons donc ce chapitre à dégager les différentes fonctionnalités de l'application à réaliser. Nous commencerons par l'identification des besoins fonctionnels et non fonctionnels et nous clôturerons ce chapitre par la conception détaillée.

### 2 Spécification des besoins

Dans ce qui suit, nous allons définir les fonctionnalités que le système doit l'offrir à ses utilisateurs et les interactions qui peuvent exister entre elles. Ces spécifications ont pour but de mieux comprendre les besoins fonctionnels et les besoins non fonctionnels de l'application.

#### 2.1 *Besoins fonctionnels*

L'application devra principalement permettre à l'utilisateur de réaliser les fonctionnalités suivantes :

- Automatiser le processus de migration manuelle des cubes,
- Simplifier les différentes étapes de migration d'un cube, afin de les accélérer et minimiser le risque d'erreur généré par la migration manuel,
- Créer des interfaces simple et ergonomique pour simplifier et clarifier les étapes de migration.

#### 2.2 *Besoins non fonctionnels*

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application ce résumant dans les points suivants :



### 2.2.1 *L'ergonomie*

L'application doit offrir une interface simple et facile à utiliser. Elle doit assurer une facilité de navigation.

### 2.2.2 *La portabilité*

On parle de portabilité de l'application pour désigner la capacité de cette dernière à pouvoir s'exécuter sur des plateformes et des environnements différents, comme par exemple : un ordinateur personnel sur Windows 7, 8, 10 ... ou un serveur sur Windows Server.

### 2.2.3 *Temps de Réponse raisonnable*

On désigne par le temps de réponse le délai au bout duquel le système parvient à répondre à une requête d'entrée. Ce terme traduit la rapidité de traitement réalisé par le système et on a toujours tendance à réduire son coût. De toute évidence, les utilisateurs interagissant avec le système désirent avoir un traitement rapide de l'information tout en gardant valide et correcte.

### 2.2.4 *Documentation*

Pour assurer la bonne utilisation de l'application, cette dernière possède un guide d'utilisation afin de clarifier les étapes de construction d'un entrepôt de données et la création d'un cube MICROSOFT.

## 2.3 *Diagramme de cas d'utilisation Globale*

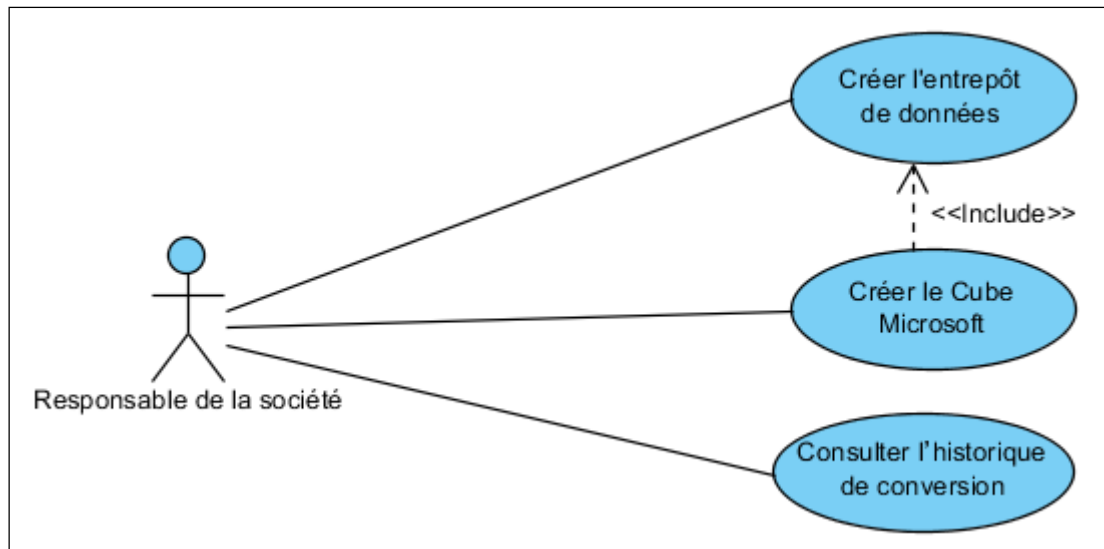
Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel du système logiciel. Ce diagramme décrit les différentes actions qui peuvent être effectuées par un acteur dans un système.

### 2.3.1 *Identification des acteurs*

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou un système). L'acteur du système est le responsable de la société.

- **Le responsable de la société :** il effectue l'opération de migration tout en respectant les étapes et les configurations de connexion requis, en cas où l'entrepôt de données est déjà créé, le responsable peut accéder directement à l'étape de création de cube.

La figure 13 présente le diagramme de cas d'utilisation global relatif à notre système.



**Figure 13.** Diagramme de cas d'utilisation Globale

### 2.3.2 Description des cas d'utilisation

En vue de bien expliquer les cas d'utilisation de notre application, nous commençons par la description des principaux cas d'utilisation dans les tableaux suivants :

**Tableau 1.** Description textuelle du cas d'utilisation « Créer l'entrepôt de données »

CU1 : Créer l'entrepôt de données
<b>Résumé :</b> Ce CU permet à l'utilisateur de créer l'entrepôt de données.
<b>Acteur :</b> Responsable de la société
<b>DESCRIPTION DU SCENARIO NOMINAL :</b>
« DEBUT »
1 : le système invite l'acteur à entrer la configuration de connexion Microsoft (nom et type de serveur, mode d'authentification).
2 : Après une vérification des paramètres, le système invite l'acteur à vérifier le chemin de la source de données des fichiers de dimensions (fichier Zip), faire l'extraction des fichiers de dimensions et créer la base de donnée.
3 : Après l'extraction, le système invite l'acteur à choisir les dimensions qui vont être converties en des table SQL et réalise la conversion et le chargement des données.
4 : La dernière étape, le système invite l'acteur à vérifier le chemin des fichiers de faits et réalise la conversion.
« FIN »

**Tableau 2.** Description textuelle du cas d'utilisation « créer le cube Microsoft »

CU2 : Créer le Cube Microsoft
<b>Résumé :</b> Ce CU permet à l'utilisateur de créer le cube Microsoft.
<b>Acteur :</b> Responsable de la société
<b>Précondition :</b> L'entrepôt de données doit être créé
<b>DESCRIPTION DU SCENARIO NOMINAL :</b> « DEBUT » 1 : le système invite l'acteur à faire entrer la configuration de connexion Microsoft : configuration du serveur moteur de base de données (nom et type de serveur, mode d'authentification et la base de données source) et configuration du serveur analysis services (nom et type de serveur, mode d'authentification). 2 : Après une vérification de ces paramètres, le système invite l'acteur à choisir la table de fait et les tables de dimensions relié et de faire la création du cube. 3 : Pour mettre l'acteur en cour, le système affiche l'état de progression de création de cube. 4 : Une fois le cube créé, le système invite l'acteur de traiter le cube afin d'être afficher, ainsi, il sera prêt pour le parcourir. « FIN »

**Tableau 3.** Description textuelle du cas d'utilisation « Consulter l'historique de conversion »

CU3 : Consulter l'historique de conversion
<b>Résumé :</b> Ce CU permet à l'utilisateur de consulter l'historique de toutes les opérations de migration effectué par le système.
<b>Acteur :</b> Responsable de la société
<b>Précondition :</b> Au minimum une opération de migration déjà réalisé.
<b>DESCRIPTION DU SCENARIO NOMINAL :</b> « DEBUT » 1 : l'acteur clique sur l'onglet « help » et choisir « log File ». 2 : le système ouvre le fichier log de l'historique. « FIN »

### 3 Conception détaillée

Dans cette section, nous détaillons la conception de notre application en se basant sur trois phases principales. Dans la première, nous décrivons la conception générale de l'application.

Dans la deuxième, nous étudions conception de la phase d'intégration des données. Et dans la troisième, nous finissons avec conception de la phase d'analyse des données.

Pour une modélisation conceptuelle parfaite, nous avons choisi le standard UML "Unified Modeling Language".

### **3.1 *Présentation UML***

UML est un langage formel couvrant le cycle de développement de logiciel de la spécification à l'implémentation. De plus, il offre aux utilisateurs la possibilité de définir leurs besoins du système à développer : ce qui permet de mieux répondre à leurs exigences et à leurs attentes. Enfin, il décrit plusieurs types de modèles (générique, expressifs, flexibles...).

En effet, nous avons choisi UML pour les spécifications des utilisateurs à travers les diagrammes de cas d'utilisation et les interactions des acteurs avec notre système à travers les diagrammes de séquence.

Dans ce qui suit, nous allons détailler les notions de bases pour quelques diagrammes que nous allons appliquer durant la phase de conception de notre application.

### **3.2 *Conception générale de l'application***

#### **3.2.1 *Diagramme de séquence***

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. La figure 14 présente le diagramme de séquence globale de notre application.

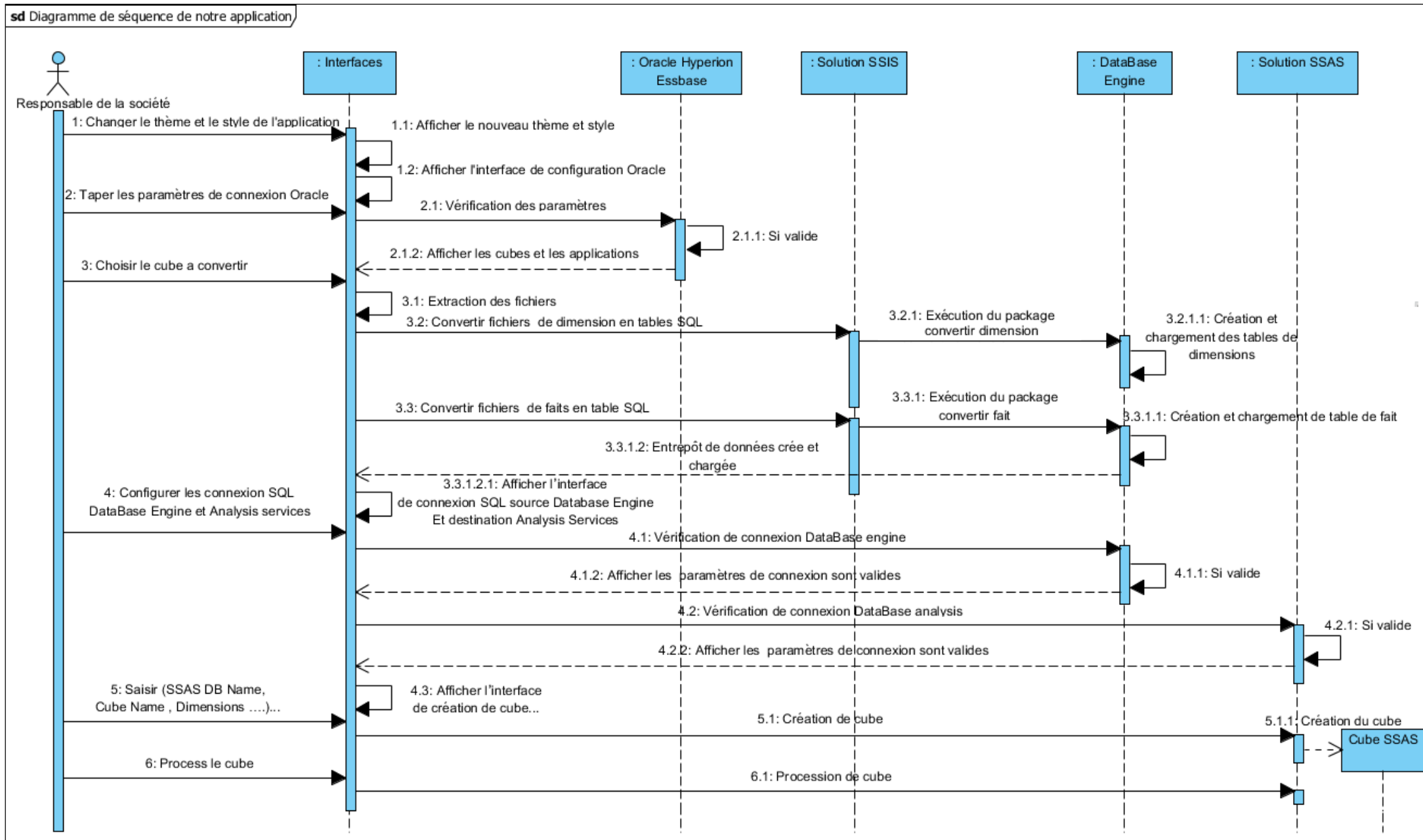
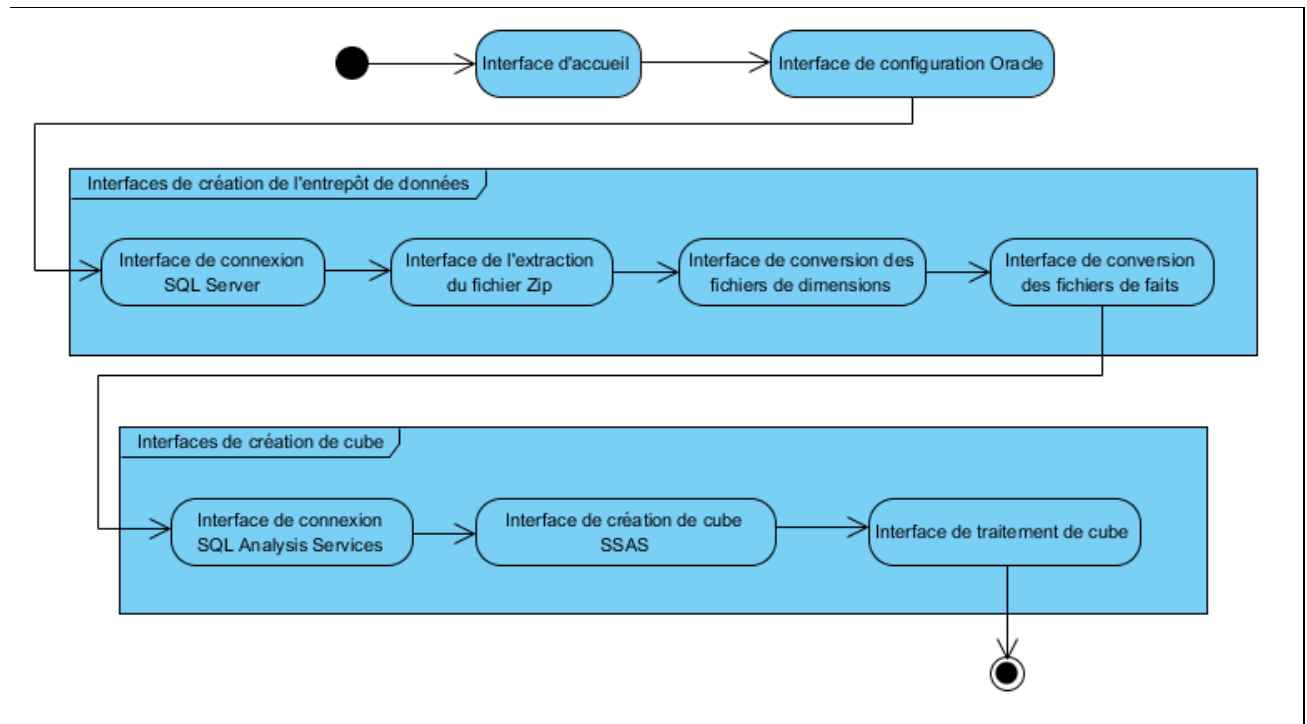


Figure 14. Diagramme de séquence

### 3.2.2 Diagramme de navigation

Le diagramme de navigation illustre la navigation au sein des éléments et des relations dans un projet. L'objectif du diagramme ci-dessous est de clarifier les étapes de migration et expliquer l'enchaînement de navigation entre les différentes interfaces de l'application en indiquant d'une façon structurée et précise les différentes actions mises à la disposition de l'utilisateur.



**Figure 15.** Diagramme de navigation

## 3.3 Conception de la phase d'intégration des données

Dans cette partie, nous présentons les diagrammes d'activité qui représentent les différents flux de création d'un entrepôt de données.

### 3.3.1 Diagramme d'activité

#### 3.3.1.1 Définition

Le diagramme d'activité est une représentation proche de l'organigramme. Ce diagramme est utilisé pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagrammes de classes ou de cas d'utilisation. La description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique.

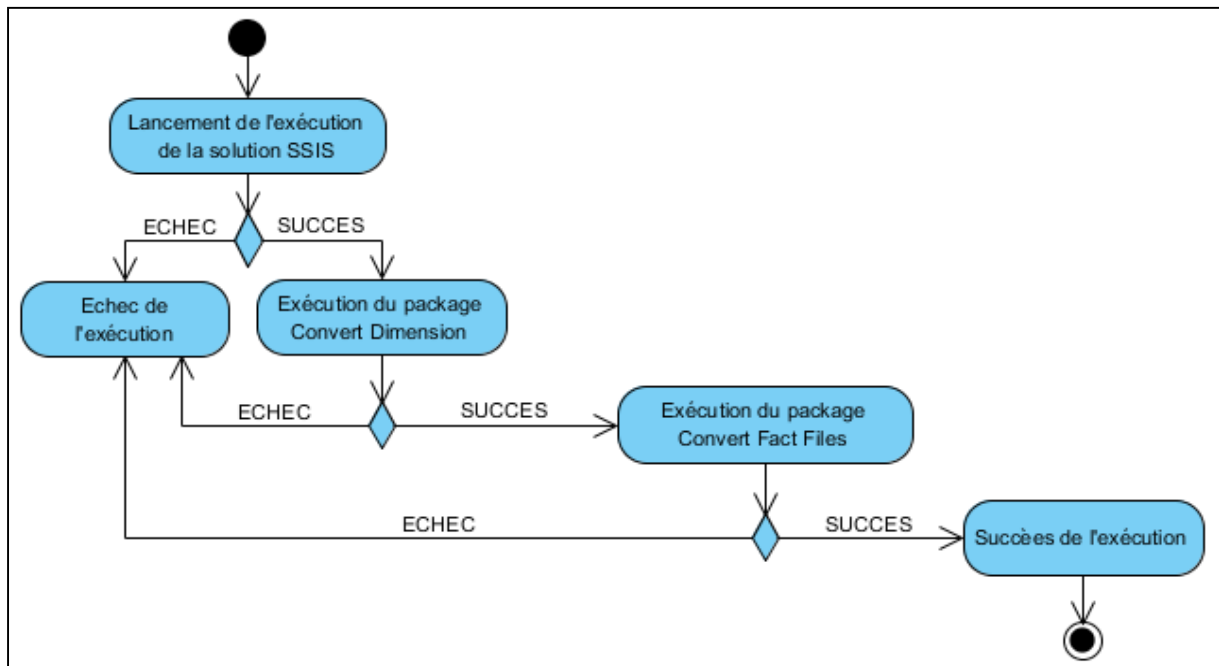
Le diagramme d'activité est composé de deux sortes d'états :

- Les états d'action ne contenant qu'une action en entrée,

- Les états d'activité ne contenant qu'une activité en leur sein.

### 3.3.1.2 Diagramme d'activité globale

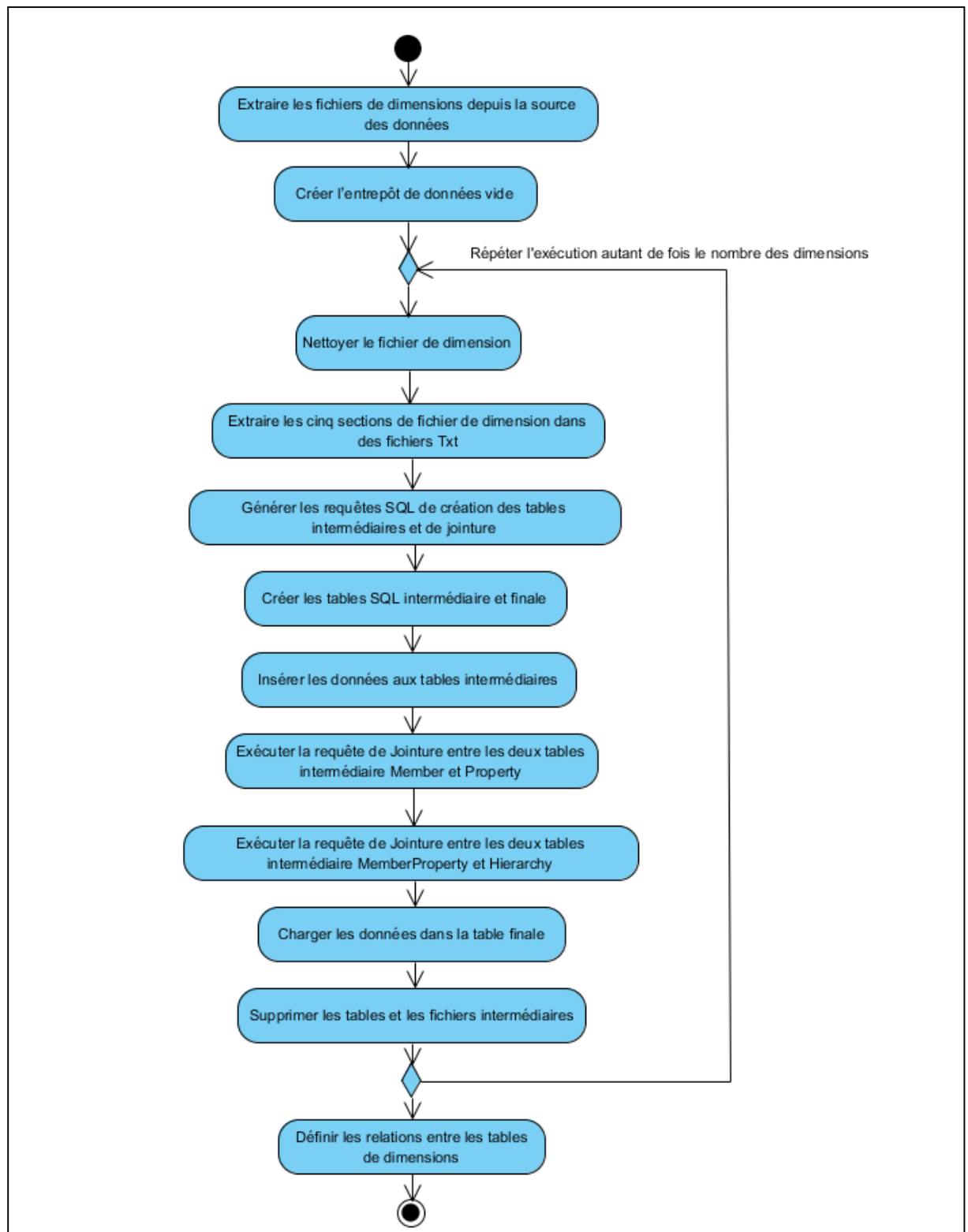
Le diagramme de la figure 16 décrit l'ordre d'exécution des packages correspond à la solution SSIS. Lors de l'exécution du chaque package, une erreur peut survenir pouvant ainsi mener à l'échec de l'exécution. Dans le cas contraire, l'exécution de ses deux flux de données donne le résultat attendu, un entrepôt de données.



**Figure 16.** Diagramme de flux de contrôle de l'entrepôt de données

### 3.3.1.3 Diagramme d'activité du package « Convert Dimension »

Le diagramme présenté ci-dessous montre les principales étapes d'exécution du package SSIS de création des tables des dimensions en passant par l'extraction des données jusqu'à leur chargement dans l'entrepôt de données.



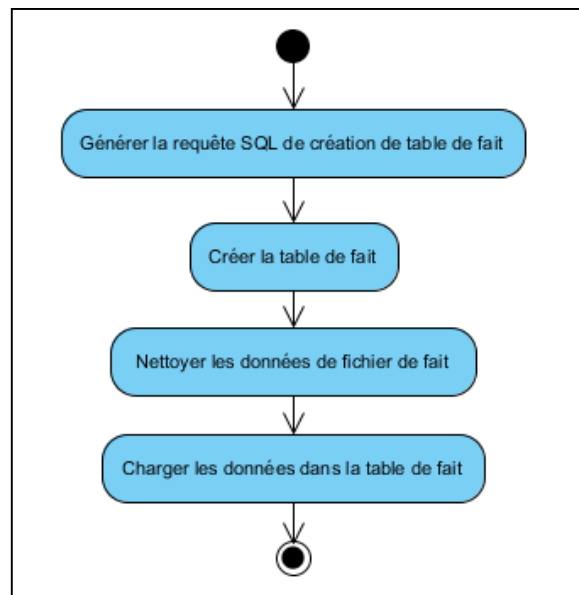
**Figure 17.** Diagramme d'activité de package de création des tables de dimensions

#### 3.3.1.4 Diagramme d'activité du package « Convert Fact Files »

La figure 18 présente le diagramme modélisant le flux SSIS de création de tables de faits. Les étapes d'exécution se résument dans la génération et l'exécution de la requête SQL de création



de la table de fait, le nettoyage et la transformation des données présentées dans le fichier source et l'insérer dans la table de fait.



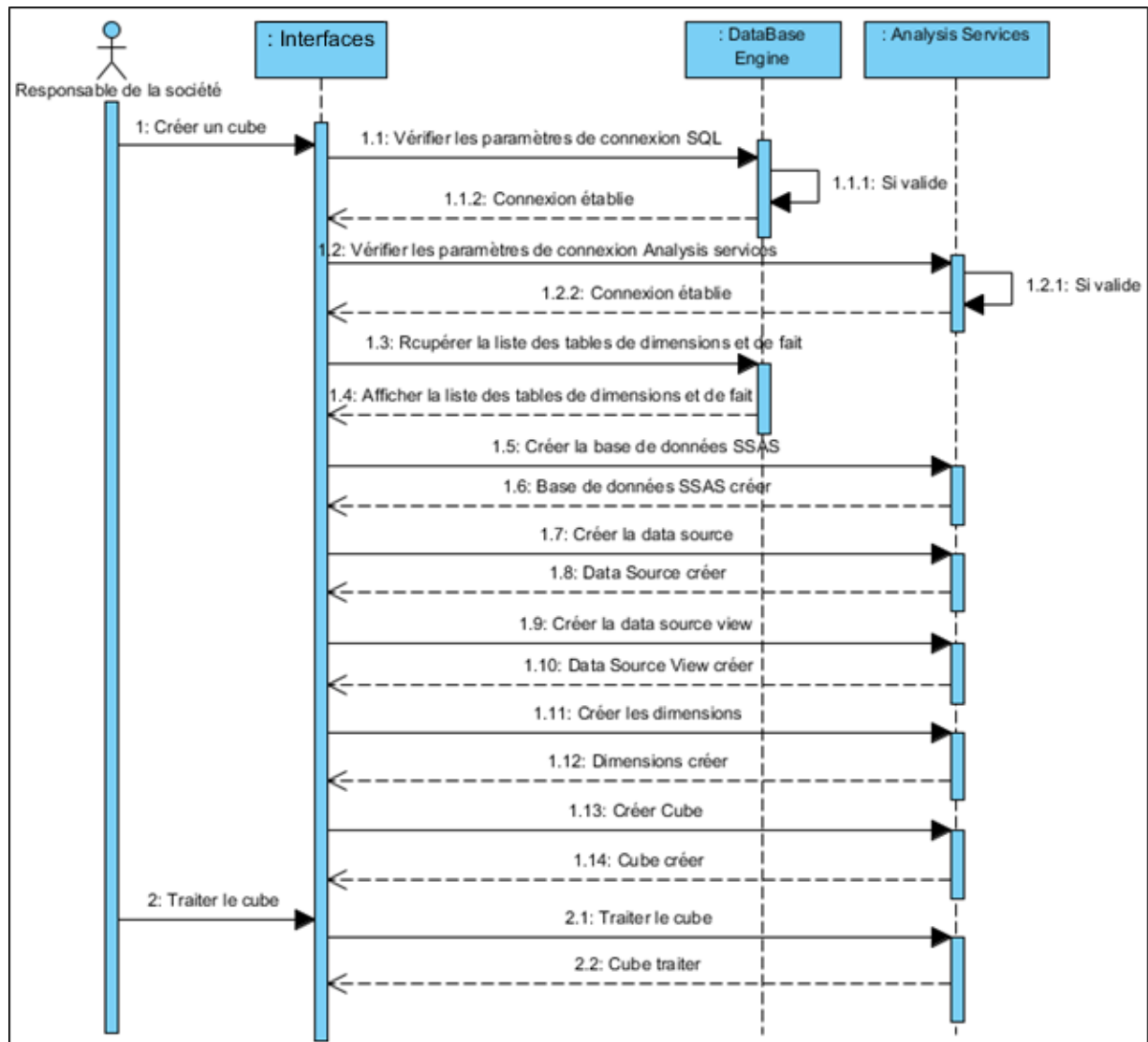
**Figure 18.** *Diagramme d'activité de package de création de table de fait*

### 3.4 Conception de la phase d'analyse des données

Dans cette partie, nous exposons le diagramme qui résume les étapes de création d'un cube SSAS multidimensionnel.

#### 3.4.1 Diagramme de séquence

La figure 19 présente le scénario de création d'un cube SSAS, commençant par la saisie des paramètres de connexion à l'entrepôt de données (où le système va récupérer sa source de données) et la saisie des paramètres de connexion de l'analysis services (où le système va enregistrer le cube à créer), arrivent à la création de la base de donnée SSAS et le cube avec ses différents composants.



**Figure 19.** Diagramme de séquence « création d'un cube SSAS »

#### 4 Conclusion

Dans ce chapitre, nous avons entamé la conception détaillée dans laquelle nous avons présenté le diagramme de cas d'utilisation, les diagrammes de séquences, les diagrammes d'activités et le diagramme de navigation de notre système.

Dans le chapitre suivant, nous allons présenter l'environnement de réalisation de notre solution et décrire d'une façon précise les étapes nécessaires pour réaliser la migration.

## Chapitre

## 4

# Réalisation

## 1 Introduction

Dans ce dernier chapitre, nous mettons l'accent sur l'étape de la réalisation du projet en se basant sur l'étude de la conception traitée dans le chapitre trois.

Nous commencerons par la présentation du choix de l'environnement de réalisation. Ensuite, nous présenterons le choix technologique, scénario d'exécution et les différentes interfaces de l'application à travers des captures d'écran.

## 2 Environnement de réalisation

Dans ce qui suit, nous présentons l'environnement matériel et logiciel ainsi que les différents langages de programmation que nous avons utilisé pour réaliser notre solution.

### 2.1 Environnement matériel

Tout au long des différentes étapes d'implémentation de notre application, à savoir l'étude théorique, la spécification des besoins, la conception, le développement et le test, deux postes de travail ont été utilisés pour le développement de l'application. Ces postes ont les configurations suivantes :

**Tableau 4.** Configuration Techniques des Ordinateurs

Caractéristiques	Processeur	Mémoire vive	Disque dur	Système d'exploitation
ASUS	Intel core i3	4 Go	500 Go	Windows 7
ASUS	Intel core i7	8 Go	1 To	Windows8 pro

### 2.2 Environnement logiciel

Durant la réalisation de l'application nous avons utilisé les outils suivants :

- **Windows 7**(machine locale) : système d'exploitation de la société Microsoft, sorti le 22 octobre 2009 et successeur de Windows Vista ;



- **Windows 8** (machine locale) : est la version du système d'exploitation Windows multiplateforme qui est commercialisée depuis le 26 octobre 2012. Version intermédiaire Windows 8.1 en 2013 ;



- **Windows Server 2008 R2** (machine distante) : est le système d'exploitation qui succède à Windows Server 2008. C'est la version serveur de Windows 7, dont il partage le noyau, Windows NT 6.1.



- **Oracle Hypérion ESSBASE v 11.1.2.3** (Extended SpreadSheet dataBASE) est un moteur de base de données multidimensionnelle de type MOLAP.



Mis sur le marché en 1992, par Arbor Software Corporation, Essbase est ensuite renommé Essbase OLAP au moment de la fusion d'Arbor avec Hyperion Software Corporation en 1998. [14]

- **SQL Server 2012** : Microsoft SQL Serveur 2012 est un système de gestion de bases de données relationnelles (SGBDR) conçues pour l'environnement de l'entreprise. Comme ses prédécesseurs, SQL Serveur 2012 comprend un ensemble d'extensions de programmation pour améliorer le langage des requêtes structurées (SQL), un langage de programmation standard et interactif pour obtenir et mettre à jour les informations d'une base de données. [15]



Il inclut deux outils de développement :

- Pour les bases transactionnelles, par le biais du **SQL Server Management Studio** (SSMS) ;
- Pour les bases décisionnelles, le forage des données, le reporting et l'ETL, par le biais d'une surcouche de Visual Studio appelé BIDS (Business Intelligence Développement Studio) puis **SQL Server Data Tools** depuis la version 2012.
- **Visual paradigm v 13.1**: est un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant



d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier. [16]

- **Visual Studio 2012** : est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. [17]



### 2.3 *Langages de programmation*

- **C#** : est un langage orienté objet de type sécurisé et élégant qui permet aux développeurs de générer diverses applications sécurisées et fiables qui s'exécutent sur le .NET Framework. Ce langage est utilisé pour créer entre autres des applications clientes Windows, des services Web XML, des composants distribués, des applications client-serveur et des applications de base de données. [18]



- **SQL** : (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. [19]



- **T-SQL** : Transact SQL, est une extension du dialecte SQL de SQL Server et constitue un langage procédural (par opposition à SQL qui est un langage déclaratif) qui permet de programmer des algorithmes de traitement des données, des procédures stockées et des triggers au sein des SGBDR Sybase Adaptive Server et Microsoft SQL Server. [20]



### 2.4 *Techniques de développement*

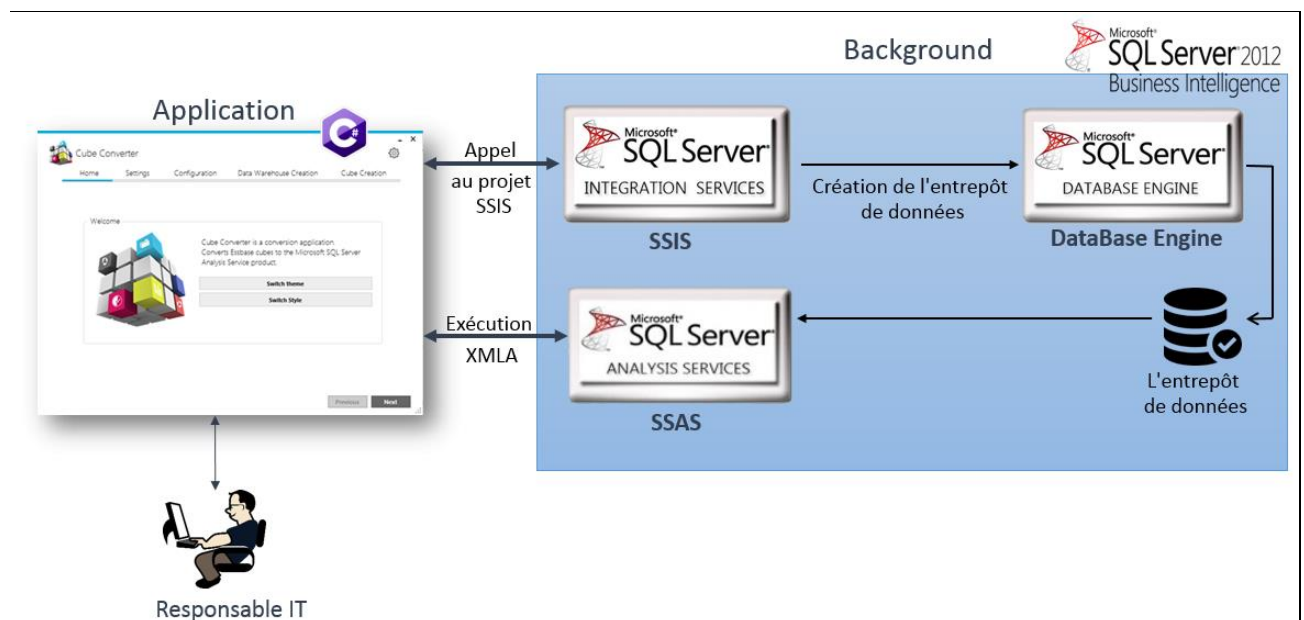
- **Metro Framework** : est un framework implémenté sous Visual studio. Ce Framework est développé pour ajouter le thème du Windows 8 aux interfaces classique de l'application WinForms ainsi que pour changer le comportement des classes d'affichage des composants de contrôle.

- **XMLA** : (XML for Analysis) est un protocole XML basé sur SOAP (Simple Object Access Protocol) conçu spécifiquement pour offrir un accès universel à n'importe quelle source de données multidimensionnelle standard accessible via une connexion HTTP. Analysis Services utilise XMLA comme seul protocole pour gérer toutes les communications entre les applications clientes. [21]



### 3 Vue architecturale du système

La figure 20 présente une vue globale de l'architecture de notre solution. L'utilisateur s'interroge avec notre application C# afin de créer l'entrepôt de données et le cube SSAS.



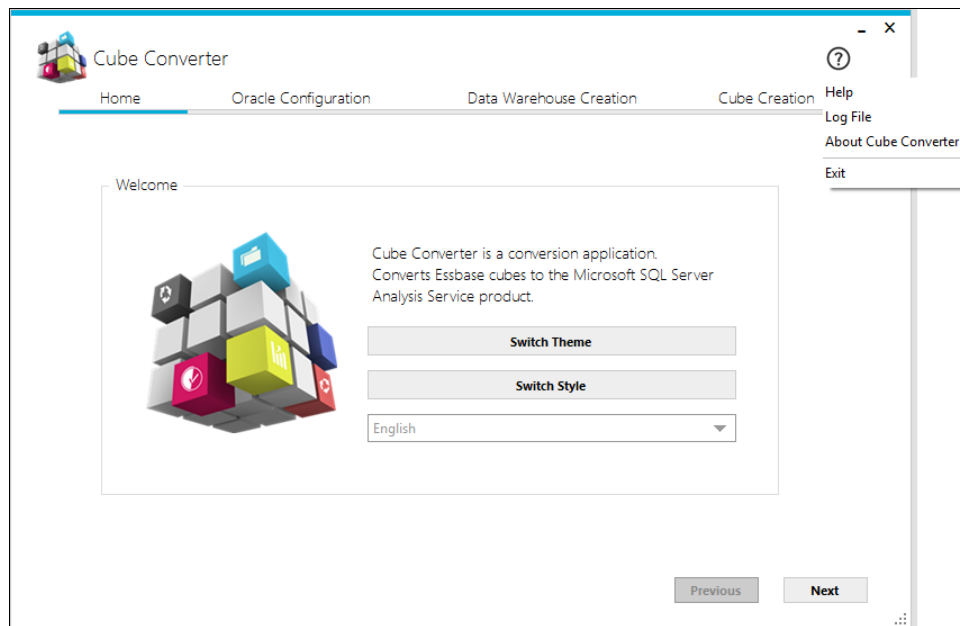
**Figure 20.** Architecture globale du système

### 4 Présentation des interfaces graphiques de l'application

Dans cette partie, nous allons présenter les captures d'écran de notre application.

#### 4.1 Interface d'accueil

Avant de commencer les opérations de configuration et des conversions, l'utilisateur peut changer le thème, le style et la langue de l'application selon son choix, via cette interface présentée dans la figure 21. Il a le droit aussi d'accéder à l'historique de ses conversions et de visualiser les pages d'aides à n'importe quelle étape.

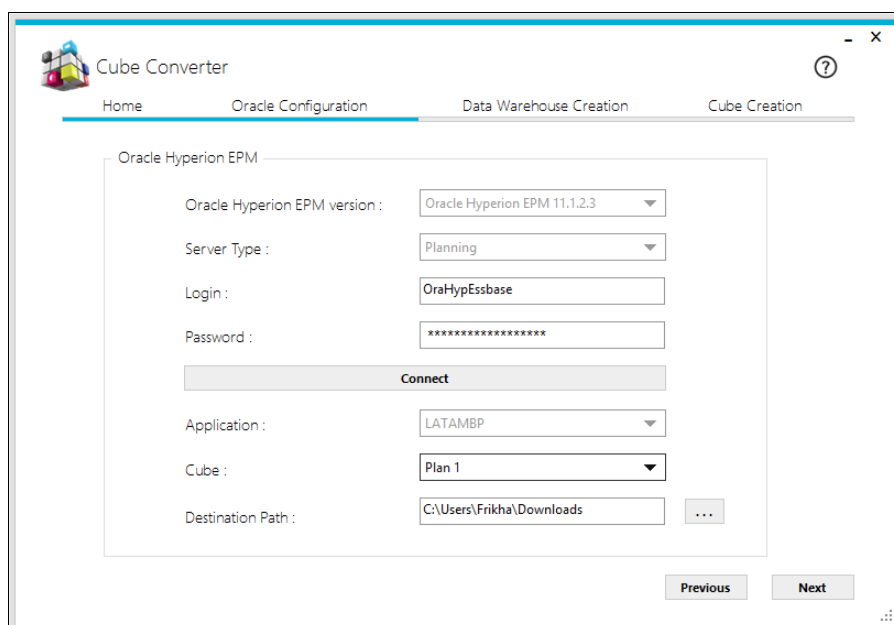


**Figure 21.** *L'interface d'accueil*

#### 4.2 Interface de connexion à Oracle Hypérion

La première étape de migration est la configuration de la version d'Oracle Hypérion actuel installé sur la machine et le type de serveur afin d'identifier la source de l'input de l'application.

Le nom de l'application Essbase, le cube à convertir et la destination de l'export doit être sélectionné par l'utilisateur après une vérification du login et mot de passe, établie par le système.



**Figure 22.** *Configuration de la connexion à Oracle Hypérion EPM*

### 4.3 Création de l'entrepôt de données

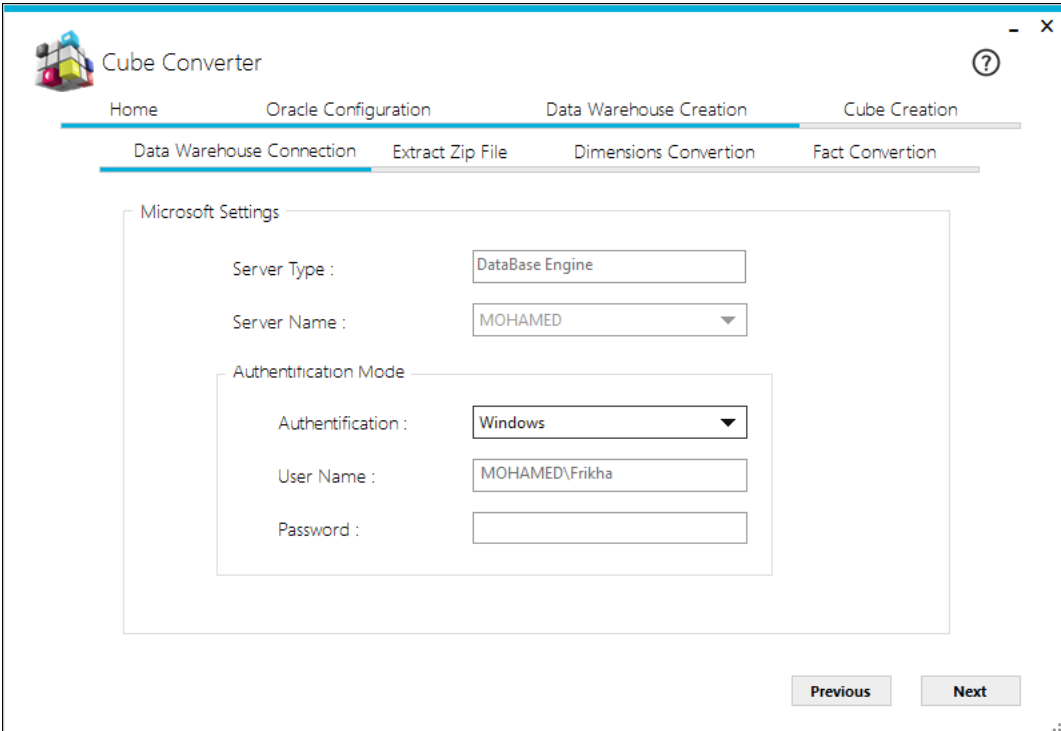
La deuxième étape de conversion est la construction de l'entrepôt de données, cette étape comporte quatre phases :

- Une phase de connexion au serveur moteur de la base de données SQL,
- Une phase pour la création de l'entrepôt de données,
- Une phase de conversion des fichiers de dimensions en tables SQL,
- Une phase de conversion des fichiers de faits en une table SQL.

#### 4.3.1 Interface de connexion au serveur moteur de base de données

À cette phase, l'utilisateur doit se connecter à un serveur de moteur de base de données, dans lequel va être enregistré l'entrepôt de données. Pour ce faire, l'usager de cette application doit sélectionner le nom du serveur et choisir son propre mode d'authentification :

- Authentification par le compte Windows,
- Authentification par son propre compte SQL.



**Figure 23.** Configuration de la connexion au serveur moteur de la base de données

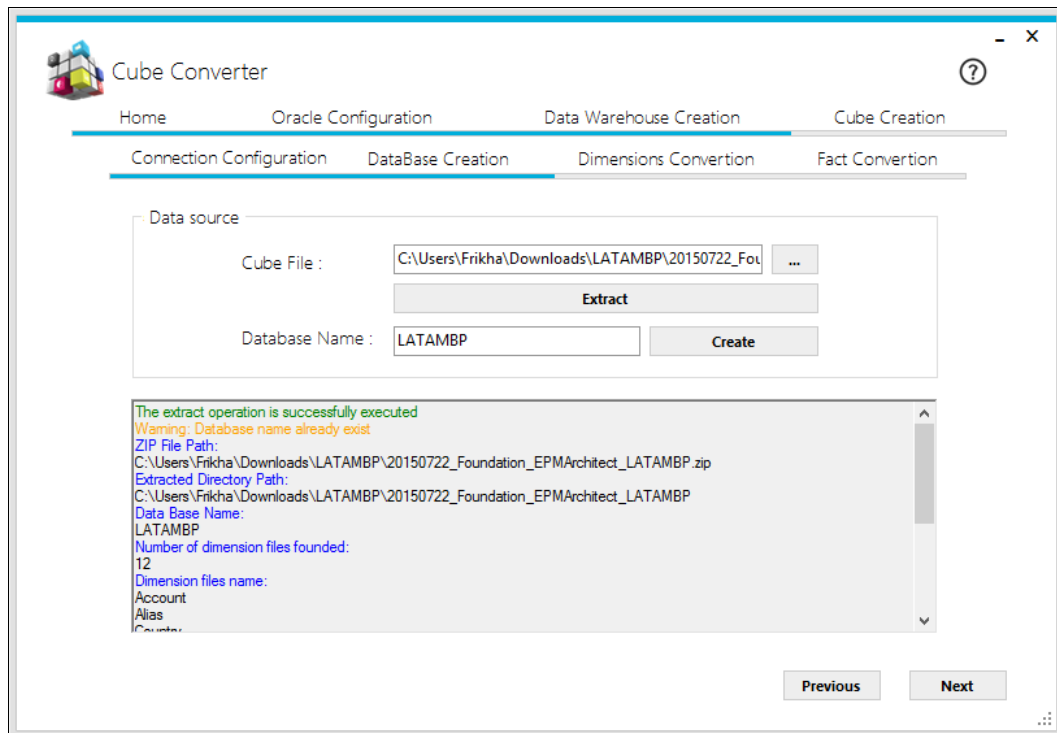
#### 4.3.2 Interface de création de l'entrepôt de données

Cette phase nous permet de définir notre source de données qui est présentée sous la forme d'un fichier d'extension Zip. Ce fichier est le résultat de l'exportation d'un cube Essbase à partir Oracle Hypérion EPM.



À cette phase, une vérification sur le contenu du fichier Zip est établie et une extraction des données effectuée lors du clic sur le bouton « extract ».

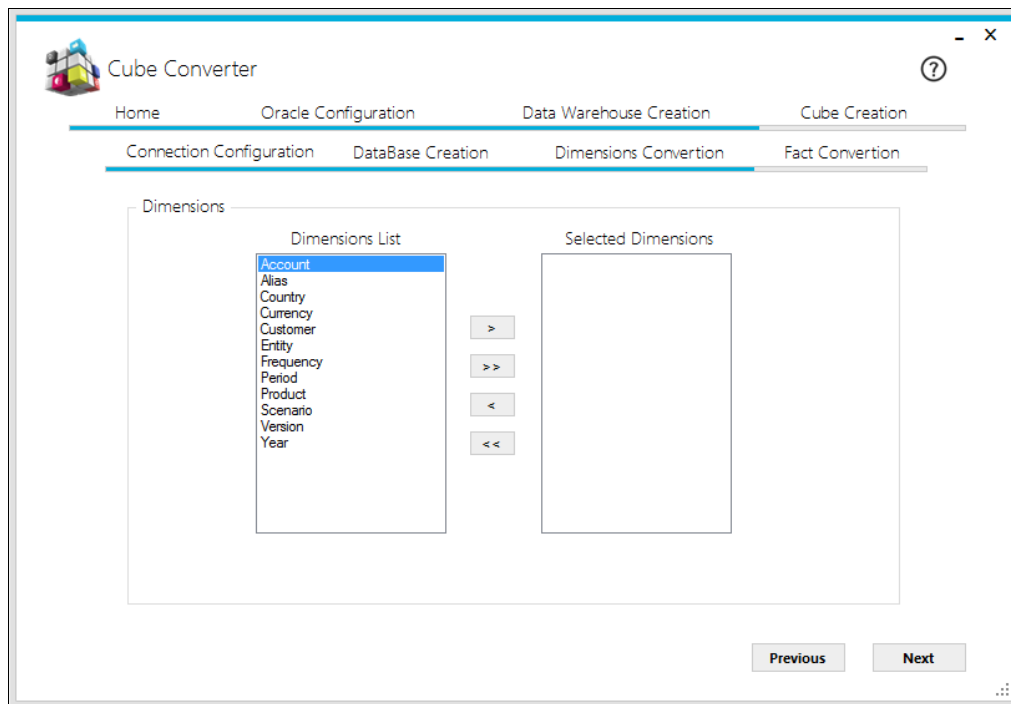
Le résultat de cette exécution affiche à l'utilisateur le nom du futur entrepôt de données, le chemin du dossier extrait, le nombre et la liste des noms des dimensions trouvées. Après cette opération l'utilisateur doit cliquer sur le bouton « create » pour lancer le processus de création de l'entrepôt de données.



**Figure 24.** Création de l'entrepôt de données

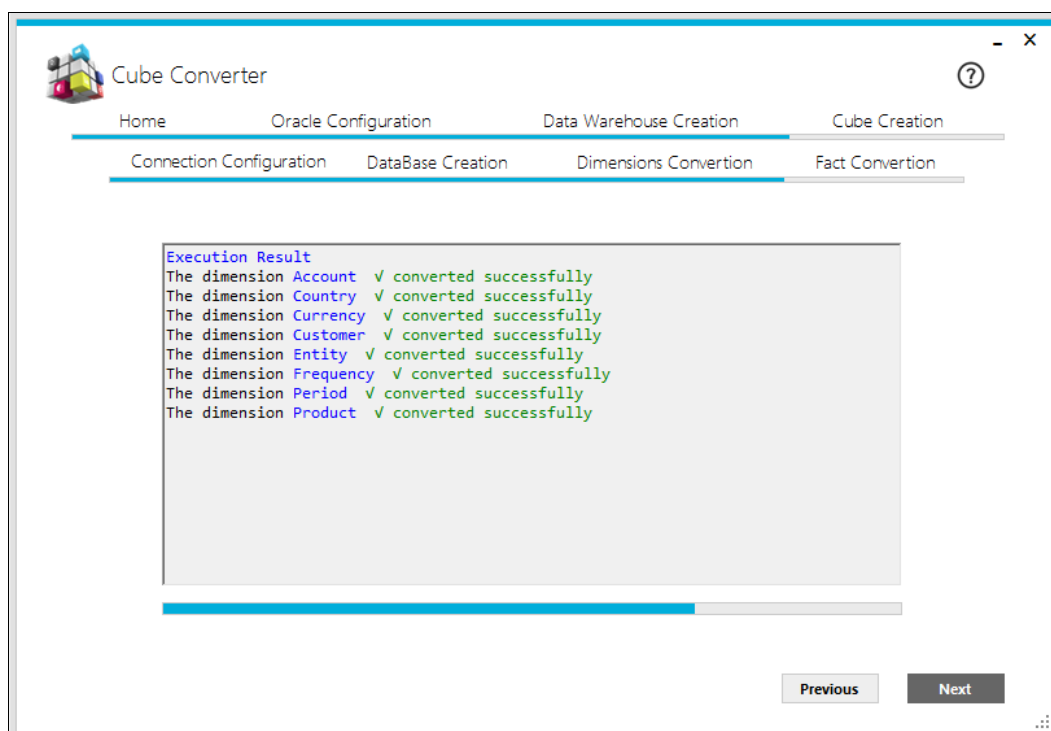
#### 4.3.3 Interface de conversion des dimensions

La troisième phase de construction d'un entrepôt de données est la conversion des fichiers de dimensions en tables SQL tout en respectant la structure, la hiérarchie et les données des fichiers. Au niveau de cette interface, un appel au flux SSIS développé « convert\_dimensions » se déclenche pour réaliser l'opération de conversion des dimensions.



**Figure 25.** *Conversion des dimensions*

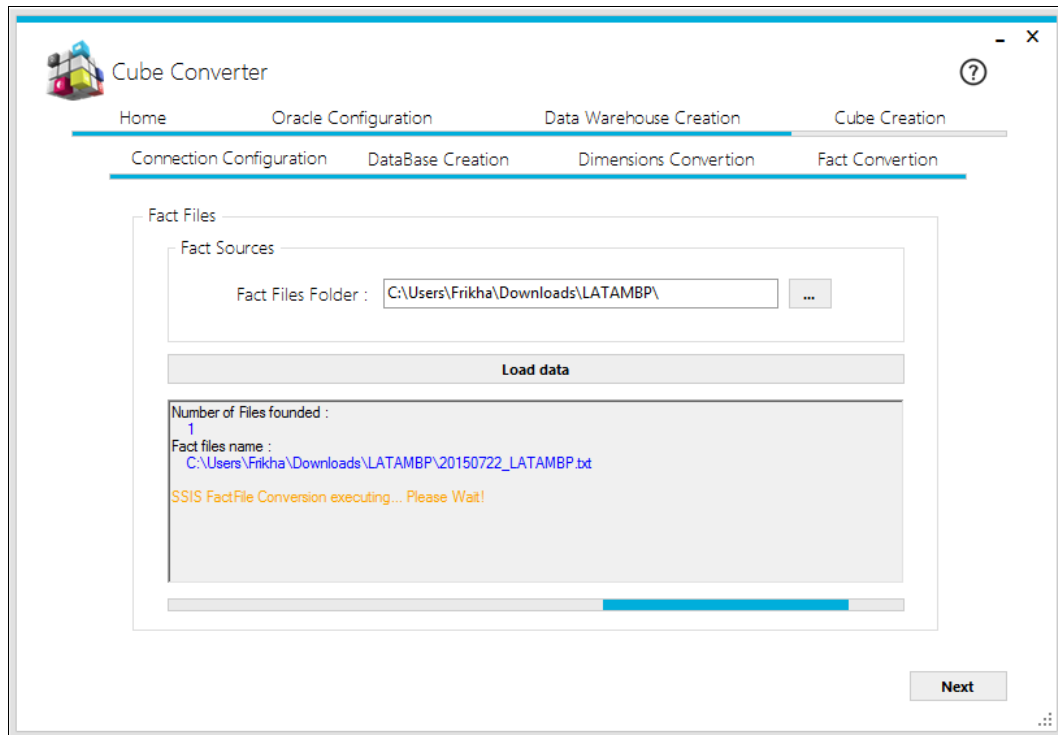
Lors du clic sur le bouton « Next », une nouvelle interface s’affiche pour mettre l’utilisateur en cours de l’état de progression des opérations effectuées.



**Figure 26.** *L’état de progression de l’exécution du flux SSIS*

#### 4.3.4 Interface de conversion des fichiers de faits

La quatrième phase de construction d'un entrepôt de données est la conversion des fichiers de faits en une tables SQL tout en respectent les dimensions relié et ses mesures.



**Figure 27.** Conversion de(s) fichier(s) de fait

### 4.4 Création du cube SSAS

La troisième étape de conversion est la construction du cube SSAS, cette étape comporte deux phases :

- Une phase de configuration de connexion au moteur de la base de données (source de données) et Analysis services (emplacement de destination du cube),

Cette étape est accessible de deux façons :

- Accès direct par le clic sur le champ de texte « Cube Création ». Ce cas d'utilisation exige que l'entrepôt de données soit déjà créé auparavant.
- Accès séquentielle après la réalisation de toutes les étapes mentionnées précédemment. Dans ce cas le formulaire à gauche sera rempli automatiquement car l'utilisateur a déjà introduit ses paramètres de connexion et le nom de la base de données dans les interfaces précédentes.
- Une phase de création du cube SSAS.

#### 4.4.1 Interface de configuration de connexion

L'interface de la figure 28 présente la première phase de construction d'un cube où l'utilisateur doit définir la configuration de connexion aux deux serveurs. Le premier est de type moteur de base de données qui contient l'entrepôt de données et le deuxième est de type analyse services où nous avons enregistré le cube SSAS.

The screenshot shows the 'Cube Converter' application window. The 'Connection Configuration' tab is active, displaying two configuration panels. The left panel, 'DataBase Engine Configuration', includes fields for 'Server Type' (DataBase Engine), 'Server Name' (MOHAMED\MSSQLSERVER2012), 'Authentication Mode' (Authentication: Windows), 'User Name' (MOHAMED\Frikha), 'Password' (empty), and 'Database Name' (LATAMPB). The right panel, 'Analysis Services Configuration', includes fields for 'Server Type' (Analysis Services), 'Server Name' (MOHAMED\MSSQLSERVER2012), 'Authentication Mode' (Authentication: Windows), 'User Name' (MOHAMED\Frikha), and 'Password' (empty). A 'Next' button is located at the bottom right of the window.

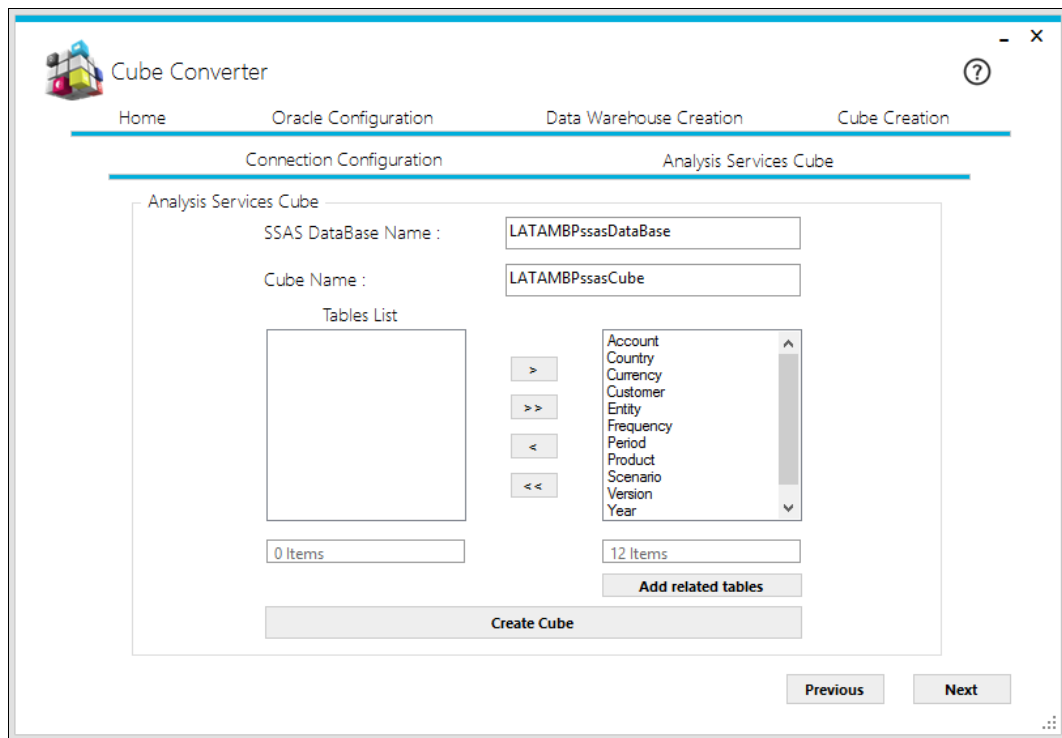
**Figure 28.** Interface de configuration de connexion

#### 4.4.2 Interface de création du cube

Cette interface présente la deuxième phase de construction d'un cube où l'utilisateur définit le nom de la base de données SSAS et le nom du cube. Il doit aussi sélectionner les tables de dimensions et de faits.

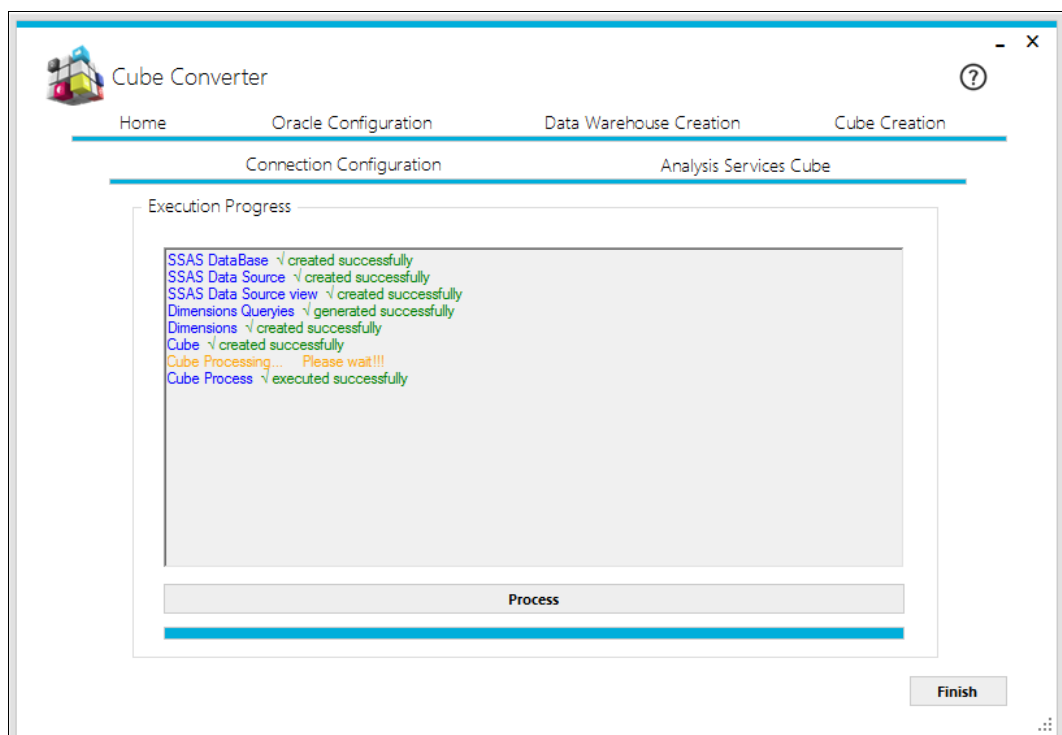
Le clic sur le bouton « Create Cube » lance la génération et l'exécution des fichiers XMLA qui nous permet de créer :

- Une Base de données SSAS ;
- La source de données ;
- Les vues des sources de données ;
- Les dimensions ;
- Le cube.



**Figure 29.** Interface de création du cube

#### 4.4.3 Interface traitement du cube



**Figure 30.** Interface de traitement du cube

L'interface de la figure 30 présente à l'utilisateur l'état de progression de création des différents composants d'un cube indiqué précédemment. Après la création du cube, l'utilisateur doit cliquer sur le bouton « Process » pour lancer l'opération de traitement.

#### 4.5 Résultat attendu de notre application

Une fois le traitement de migration est terminé, l'application lance le logiciel SQL server management Studio afin de tester et analyser les données de cube SSAS.

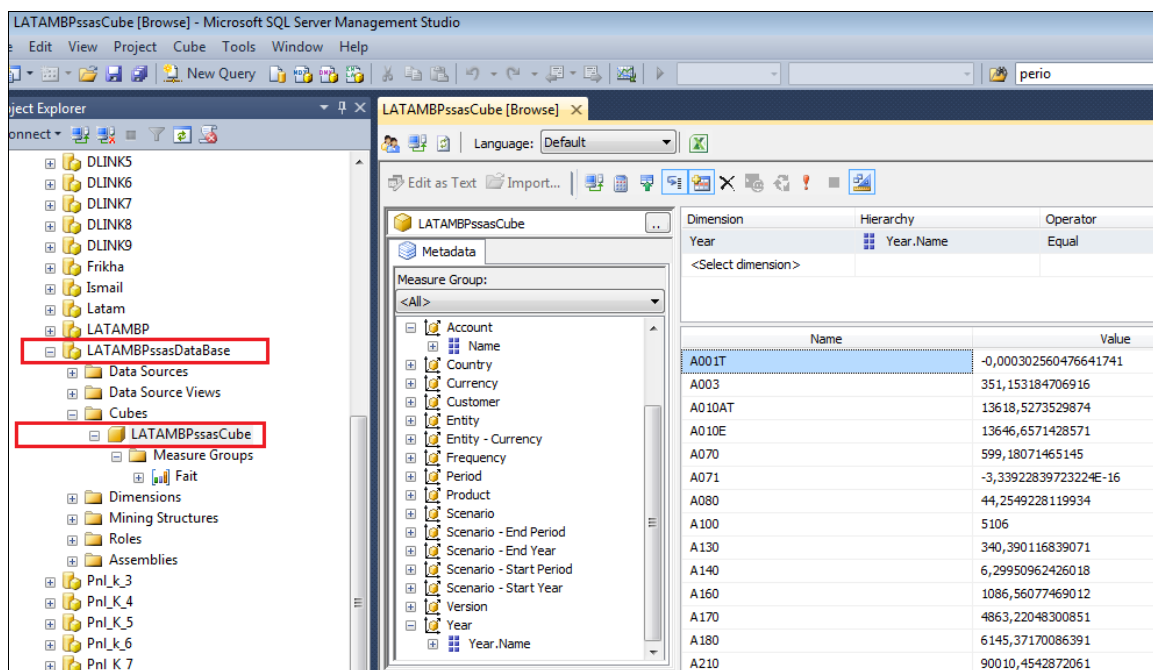


Figure 31. Cube Microsoft finale

## 5 Conclusion

Ce chapitre nous a donné une idée sur l'ensemble des logiciels et matériels qui ont été utilisés dans le développement d'une application qui réalise une migration entre deux technologies. Nous avons clôturé ce chapitre par la présentation des différentes interfaces et fonctionnalités du notre application.

# Conclusion Générale

La variété des outils d'informatique décisionnel est un facteur majeur aujourd'hui de migration des sociétés vers d'autres technologies BI afin d'obtenir des résultats de décision plus fiable, pour des raisons des couts de licences ou bien pour des raisons de maitrise de technologie et des compétences humain.

Dans ce contexte, notre projet de fin d'étude s'est déroulé au sein de la société TODEOS NEARSHORE afin de mettre en place une solution d'automatisation de la migration d'un cube Oracle en cube Microsoft.

Dans le but de mener le projet à bien, notre travail est réparti en trois grandes parties, à savoir : une première partie dédiée à la construction de l'entrepôt de données élément majeur de l'application. La deuxième partie se résume en la génération du Cube Microsoft, la création des relations entre les faits et les dimensions et le déploiement du cube à l'aide de l'outil SQL Server Analysis Services. La dernière partie est consacré au développement une application lourde qui fait l'appel à ces deux modules mentionnées précédemment.

A travers la réalisation de notre application, nous avons eu la chance d'appliquer nos connaissances théoriques acquises dans notre cursus universitaire pour réaliser une architecture logiciel et développer une application qui répond aux besoins. Ainsi, ce projet été une occasion pour maitriser les outils SSIS qui nous a permet de réaliser la partie ETL du projet et la création de l'entrepôt de données, et maitriser les outils SSAS pour créer le cube multidimensionnel Microsoft.

Notre application est bien appréciée en termes de réduction des coûts budgétaires et de diminution du temps de migration manuelle. Néanmoins, elle présente quelques insuffisances, nous pouvons alors les proposer comme perspectives:

- Améliorer l'application pour être plus générique afin qu'elle traite tous les schémas de modélisation d'un entrepôt de données possible et non seulement le modèle en étoile,
- Réduction du temps d'exécution de la conversion des fichiers de faits qui sont vraiment massive,
- Réalisation d'une connexion distante à l'Oracle Hypérion installé sur des serveurs distants,
- Extraction automatique des fichiers de dimensions à partir l'Oracle Hypérion Workspace,
- Conversion des scripts claque Oracle en des membres calculé sous SSAS Microsoft.

## Bibliographie

[2] Ivar Jacobson, Grady Booch, James Rumbaugh, «*Le processus unifié de développement logiciel*», Eyrolles, 2000 - 488 pages

## Webographie

[1] <http://www.todeos.com/> : Présentation de la société TODEOS NEARSHORE : Avril 2016

[3] <https://bellouti.wordpress.com/2007/09/24/le-developpement-iteratif/>: les avantages attendus par le développement itératif: Avril 2016

[4] <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/informatique-informatique-decisionnelle-15057/>: Définition de l'informatique décisionnelle: Avril 2016

[5] <http://blog.netapsys.fr/les-etapes-et-notions-dun-projet-bi-2/>: Présentation des 4 phases de la chaîne décisionnelle: Avril 2016

[6] <http://www-igm.univ-mlv.fr/~dr/XPOSE2010/chainedecisionnelle/modelisation.html>: Introduction à la chaîne décisionnelle : Juin 2016

[7] [http://broom02.revolvy.com/main/index.php?s=OLAP%20cube&item\\_type=topic](http://broom02.revolvy.com/main/index.php?s=OLAP%20cube&item_type=topic): Définition de hiérarchie: Avril 2016

[8] [http://www.mcnext.com/pole\\_decisionnel/business\\_intelligence/Pages/Default.aspx](http://www.mcnext.com/pole_decisionnel/business_intelligence/Pages/Default.aspx) : Business Intelligence Microsoft : Mai 2016

[9] <https://msdn.microsoft.com/fr-fr/library/mt204009.aspx> : Définition de l'SQL Server Data Tools : Mai 2016

[10] <http://www.piloter.org/business-intelligence/outils-business-intelligence-oracle-hyperion.htm>: Les principaux produits de la solution Oracle Hypérion: Avril 2016

[11] [http://www-igm.univ-mlv.fr/~dr/XPOSE2006/DELTIL\\_PEREIRA/pentaho.html](http://www-igm.univ-mlv.fr/~dr/XPOSE2006/DELTIL_PEREIRA/pentaho.html): Juin 2016

[12] [http://www.dimo-analytics.fr/solutions/logiciels\\_cognos/ibm-cognos-10](http://www.dimo-analytics.fr/solutions/logiciels_cognos/ibm-cognos-10)

[13] <http://www.exologic.com/products.htm> : Site Officiel du « Cube Port » : Mai 2016



**[14]** <http://sroux.developpez.com/tutoriels/essbase/> : Présentation d'Oracle Hypérion Essbase : Mai 2016

**[15]** <http://whatis.techtarget.com/definition/SQL-Server-2012> : Définition du SQL Server 2012 : Mai 2016

**[16]** <http://www.clubic.com/telecharger-fiche384046-visual-paradigm-for-uml.html>: Présentation du visual paradigm : Mai 2016

**[17]** [https://msdn.microsoft.com/fr-fr/library/fx6bk1f4\(v=vs.90\).aspx](https://msdn.microsoft.com/fr-fr/library/fx6bk1f4(v=vs.90).aspx): Présentation du visual studio 2012 : Mai 2016

**[18]** <https://msdn.microsoft.com/fr-fr/library/z1zx9t92.aspx> : Introduction au langage C# et Plateforme.NET : Mai 2016

**[19]** <http://www.lesdeveloppeurs.net/BD-SGBD-definition-de-sql-274> : Définition de SQL : Mai 2016

**[20]** <http://sqlpro.developpez.com/cours/sqlserver/transactsql/> : Petit guide de Transact SQL : Mai 2016

**[21]** [https://msdn.microsoft.com/fr-fr/library/ms186654\(v=sql.120\).aspx](https://msdn.microsoft.com/fr-fr/library/ms186654(v=sql.120).aspx) : Développement avec XMLA dans Analysis Services : Mai 2016

## Annexe : Solution SSIS

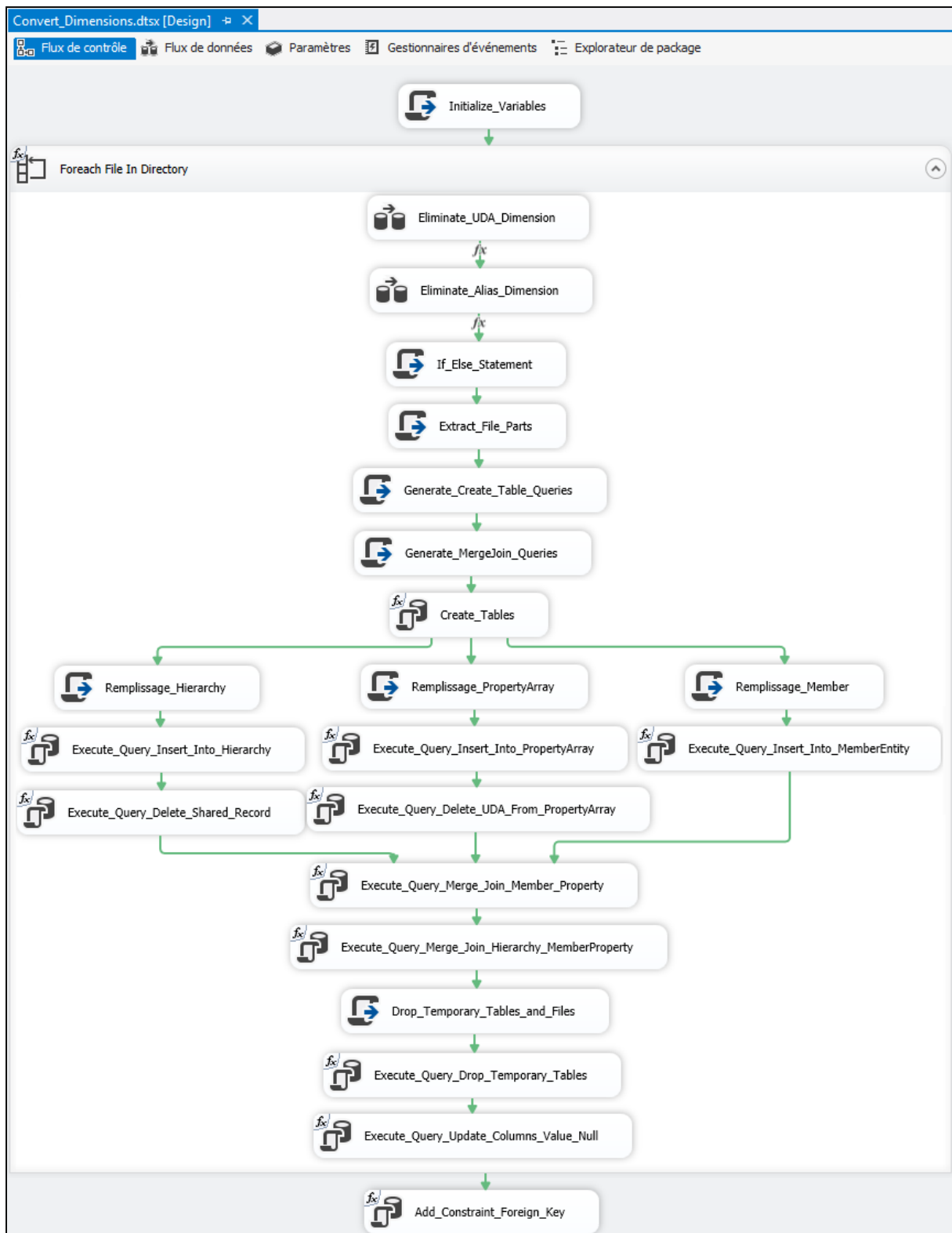
Microsoft Integration Services est une plateforme qui permet de créer des solutions de transformation de données et d'intégration de données au niveau de l'entreprise. Integration Services nous permet de résoudre des problèmes professionnels complexes en copiant ou en téléchargeant des fichiers, en envoyant des messages électroniques en réponse à des événements, en mettant à jour des entrepôts de données, en nettoyant et en explorant des données et en gérant des données et des objets SQL Server. Integration Services peut extraire et transformer des données à partir d'un éventail de sources, par exemple des fichiers de données XML, des fichiers plats et des sources de données relationnelles, puis charger les données dans une ou plusieurs destinations.

Integration Services inclut un ensemble riche de tâches et de transformations intégrées, des outils pour construire des packages, et le service Integration Services permettant d'exécuter et de gérer des packages. Les tâches d'intégration sont rassemblées dans un package ou lot. Le flux de contrôle établit l'enchaînement des tâches du package. Certaines tâches ont pour vocation d'assurer la transformation de données. On appelle tâches de flux de données. Les flux de données traitent essentiellement des fonctions de transformation de données. Elles sont minimum composées d'une source de données, d'une transformation et d'une destination.

Nous présentons dans ce qui suit notre solution SSIS qui nous permet de créer l'entrepôt de données. Elle est constitué principalement par deux packages. Le premier intitulé « Convert\_Dimension » permet la création des tables de dimensions et le deuxième intitulé « Convert\_Fact\_File » responsable à la création de table de faits.

### ❖ Package SSIS « Convert Dimensions »

La figure ci-dessous présente l'implémentation du package « Convert\_Dimensions » qui nous permet de convertir les fichiers des dimensions de n'importe quel type et de n'importe quel cube, en des tables SQL. Ce flux est constitué principalement par une tâche de script permet l'initialisation des variables utilisé dans se package, un conteneur de boucle foreach permet le parcours des fichiers de dimensions et les convertir en des tables SQL et une tâche d'exécution de requête SQL permet la définition des relations entre les tables créer.



Dans la partie suivante, nous allons présenter en détail les composants qui constituent ce package.

- *Initialisation des variables*

Le premier composant tâche de script nous permet d'initialiser les variables et les paramètres de notre package SSIS et de vérifier l'existence de l'entrepôt de données. S'il n'existe pas il le crée.

- *Parcours des fichiers de dimensions*

Ce composant est utilisé pour parcourir les fichiers de dimensions d'un répertoire. Chaque fichier sera traité par une suite d'étapes qui sont présentées dans cette boucle.

- *Eliminate UDA, Alias Dimension*

Ces deux composants sont utilisés pour éliminer les deux fichiers de dimension UDA et Alias car ils sont vides pour tous les cubes traités. La dimension UDA présente les attributs définis par l'utilisateur et la dimension Alias présente les alias et les descriptions qui nous aident à définir les dimensions.

- *If\_Else\_Statement*

Cette tâche de script nous permet de rendre une requête Script Calque présenté en plusieurs lignes sur une seule ligne.

- *Extract\_File\_Parts*

Tous les fichiers des dimensions extraites à partir le gestionnaire des bases de données Oracle Hypérion EPM sont structurés d'une façon bien spécifique. Ces fichiers de dimension sont organisés en cinq sections : Dimensions, DimensionsAssociations, Members, Hierarchies et PropertyArray. Le rôle de ce composant est d'extraire ces cinq parties et de les enregistrer dans des fichiers intermédiaires d'extension Txt. Chaque fichier présente une section.

- *Generate\_Create\_Tables\_Queries*

Ce composant nous permet de générer les requêtes SQL de création des tables intermédiaires et finale d'une dimension traitée ainsi que les requêtes de jointure entre les tables intermédiaires.

- *Generate\_MergeJoin\_Constraints*

Dans certains cas des dimensions traitées, nous avons trouvé des dimensions qui sont associées à d'autres dimensions. Ce composant nous permet de définir et de générer les requêtes de jointure entre ces dimensions.

- Create\_Tables

Cette tâche d'exécution d'une requête SQL nous permet d'exécuter les requêtes de création des tables temporaires et finale déjà générées auparavant.

- Remplissage des tables temporaires

Les composants présentés en parallèle sont utilisés pour extraire les données à partir des fichiers intermédiaires et de les insérer dans des tables temporaires.

- Execute\_Query\_Delete\_Shared\_Record

Ce composant nous permet de supprimer les membres partagés. Les membres partagés sont des membres de dimension qui apparaissent dans plusieurs emplacements d'une hiérarchie.

- Execute\_Query\_Delete\_UDA\_From\_PropertyArray

Comme nous n'avons pas traité la dimension UDA précédemment, elle sera aussi éliminée dans la partie PropertyArray.

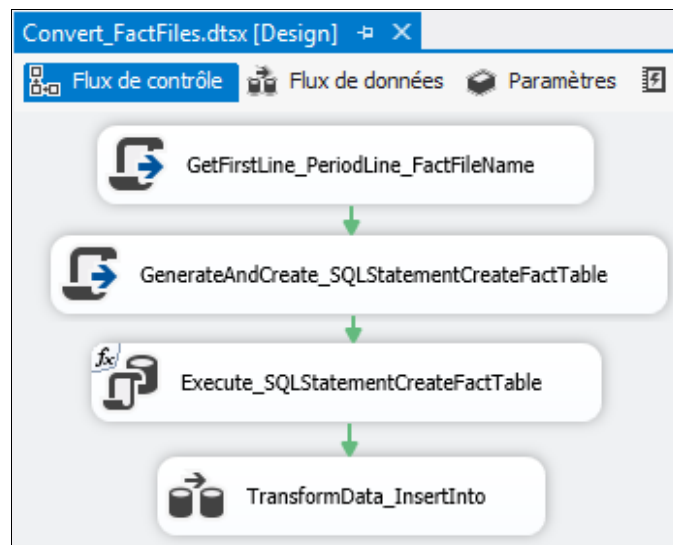
- Jointure des tables intermédiaire et remplissage de la table finale

Après le remplissage des tables intermédiaire, nous avons exécuté la requête de jointure entre la table « Member » et « PropertyArray ». Le résultat de cette jointure sera enregistré dans la table « MemberPropertyArray ». La deuxième requête de jointure entre la table « MemberPropertyArray » et « Hierarchy » sera exécutée et le résultat sera enregistré dans la table finale. Après le remplissage de la table finale, les tables intermédiaires et les fichiers temporaires seront supprimés d'une façon automatique.

- Définition des relations entre les tables de dimensions

Après le traitement de tous les fichiers des dimensions, nous avons exécuté les requêtes qui définissent les liaisons entre les dimensions.

### ❖ Package SSIS « Convert Fact File »



La figure ci-dessus présente l'implémentation du package « Convert Fact Files » qui nous permet de convertir le(s) fichier(s) de faits en une table SQL. Ce flux est constitué par deux tâches de script. La première tâche de script intitulé « GetFirstLine\_PeriodLine\_FactFileName » et la deuxième intitulée « GenerateAndCreate\_SQLStatementCreateFactTable ». Ce package contient aussi une tâche d'exécution de requêtes SQL et une tâche de flux de données qui est composé par une source du fichier plat et un composant script. Dans ce qui suit nous présentons le rôle de chaque composant :

- GetFirstLine\_PeriodLine\_FactFileName

Ce composant nous permet d'extraire le nom du fichier de fait à partir le chemin introduit au flux, la première et la deuxième ligne du fichier qui représentent successivement l'axe période et le premier enregistrement qu'il sera enregistré dans la table de fait.

- GenerateAndCreate\_SQLStatementCreateFactFile

Le deuxième composant tâche de script nous permet de générer automatiquement la requête SQL de création de la table de fait toutes en définissons ses colonnes, leurs types et leurs tailles d'une façon optimale.

- Execute\_SQLStatmentCreateFactTable

Ce composant nous permet d'exécuter la requête SQL de création de la table de fait qui a était générer par le composant « GenerateAndCreate\_SQLStatementCreateFactFile » et les requêtes de jointure qui définissent la liaison entre la table de fait et les tables des dimensions.

- TransformData\_InsertInto

Cette tâche de flux de données nous permet de transformer le contenu de chaque ligne à fin de définir un séparateur unique pour tous les lignes. Après avoir réalisé cette transformation nous avons enregistré tous ces lignes dans la table que nous avons déjà créée précédemment.

Le résultat d'exécution de ces deux packages donne un entrepôt de données, nous présentons dans ce qui suit le schéma en étoile d'un cas réel d'entrepôt de données du cube «LATAMBP» qui comporte une table de faits et des différentes dimensions. Nous expliquerons en détail quelques tables.





Nous présentons dans le tableau suivant la liste de quelque table et leurs types.

Nom de la table	Type de la table	Cardinalité	Description
<b>Fait</b>	Table de faits	* - 0..1	Cette table contient une mesure intitulée value et dix références vers les tables de dimension.
<b>Currency</b>	Table de dimension	* - 0..1	Représente les devises utilisées dans l'entrepôt de données : Euro, USD,..
<b>Frequency</b>	Table de dimension	* - 0..1	Représente la fréquence des opérations de calcul : Mois, du début de l'année en cours jusqu'au jour présent,...
<b>Account</b>	Table de dimension	* - 0..1	Représente les informations du compte tel que : le type, Les ventes nettes, revenue,..
<b>Period</b>	Table de dimension	* - 0..1	C'est une dimension temporelle qui est la base de chaque entrepôt de données utile pour l'historisation des données et le suivi de l'évolution de l'information.
<b>Scenario</b>	Table de dimension	* - 0..1	Représente les scénarios de calcul utilisés comme par exemple actuel, budget, forecast.
<b>Country</b>	Table de dimension	* - 0..1	Représente les informations des pays tel que le nom et le code sur deux caractères.
<b>Product</b>	Table de dimension	* - 0..1	Elle comprend les informations concernant les produits comme le code du produit, sa description.
<b>Customer</b>	Table de dimension	* - 0..1	Elle comprend les informations concernant les clients.

## DEPARTEMENT DE GENIE INFORMATIQUE

# MISE EN PLACE D'UNE SOLUTION D'AUTOMATISATION DE LA MIGRATION D'UN CUBE ORACLE EN CUBE MICROSOFT

**MOHAMED FRIKHA – ISMAIL ABIDA**

### RESUME :

Dans le cadre de ce projet de fin d'étude, nous avons conçu et réalisé une solution d'automatisation de la migration d'un cube Oracle en cube Microsoft. L'objectif de cette solution est d'automatiser et de faciliter l'opération de migration des cubes aux responsables des sociétés, que ce soit novice ou expert au domaine de l'informatique décisionnelle.

L'étude et la conception sont réalisées conformément avec le langage de conception des systèmes d'information « UML », l'implémentation des entrepôts de données et les bases de données multidimensionnelles sont faites sous « SQL Server » et les interfaces de l'application sont réalisées avec le langage de programmation « C# ».

### ABSTRACT :

As part of this final project study, we designed and implemented an automation solution for migration from an Oracle cube to Microsoft cube. In order to automate and facilitate the cubes migration operations for the IT Managers whether a BI novice or an expert.

The study and the conception step are achieved with « UML » language, the data warehouse, the multidimensional databases are stored in « SQL Server », and the application interfaces are produced with « c # » language.

**MOTS CLES :** informatique décisionnelle, SSIS, SSAS, Entrepôt de données, Cube OLAP.

**KEYWORDS:** Business Intelligence, SSIS, SSAS, Data Warehouse, Cube OLAP.