

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITÉ
Abdelhamid Ibn Badis
MOSTAGANEM

UNIVERSITÉ ABDELHAMID IBN BADIS DE MOSTAGANEM

Faculté des Sciences Exactes & de l'Informatique

Département de Mathématiques et d'Informatique

Filière Informatique

Projet de fin d'étude

Pour l'obtention de diplôme de

Master en Informatique

Option : Ingénierie des systèmes d'information

Thème

**Architecture basée sur les modèles des entrepôts de données
pour l'analyse multidimensionnelle des communications
téléphoniques**

Présenté par :

- Beladjel Abdellah
- Mertad Abdelkadir

Encadré par :

- Abdallah Bensalloua Charef

Co-encadré par :

- Benameur Abdelkader

Année Universitaire :2018/2019.

RESUME

Les technologies DW (Data Warehouse) et OLAP (On-Line Analytical Processing) sont utilisées pour stocker les observations, faciliter la visualisation, l'analyse, la comparaison et la prise de décision.

Ces technologies permettent aux utilisateurs de produire facilement des rapports et des graphiques. Dans ce travail, dans un premier temps, nous avons réalisé une étude de l'état de l'art sur les différents outils de l'informatique décisionnelle. Dans la deuxième étape nous avons opté pour une modélisation basée sur UML de l'entrepôt de données du domaine des communications téléphoniques. Ensuite nous avons développé une application décisionnelle (OLAP) qui permet d'exploiter cet entrepôt de données pour l'analyse multidimensionnelle et en ligne des données relatives au cas d'étude.

Mots clés : Data Warehouse, OLAP, UML, Analyse Multidimensionnelle, Gestion des communications téléphoniques.

Remerciement

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nos remerciements avant tout d'abord au corps professoral et administratif de la Faculté des Faculté des Sciences Exactes de l'université Abdelhamid ibn badis Mostaganem, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous tenons à remercier sincèrement Monsieur Charef Abdallah Bensalloua qui, en tant que Directeurs de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il a bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Nous tenons à remercier sincèrement Monsieur Benameur Abdelkader, en tant que co-encadreur de mémoire, qui nous a aidé pour la réalisation de ce travail.

Nous tenons aussi à exprimer nos remerciements aux membres de jury qui ont accepté d'examiner notre travail. Nous n'oublions pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire.

Liste des abréviations

API : Application programmation interface

ATM : Algérie Telecom Mobile

BI : Business Intelligence

BIDS : Business Intelligence Développement Studio

DWH : Data Warehouse

ED : Entrepôt de Données

ETL : Extract Transformation Loading

GSM : Global System for Mobile Communication

IIS : internet information service

LTE : Long Term Evolution Technology

MDX : Multi Dimensional eXpression

OLAP : OnLine Analysis Processing

OLTP : OnLine Transactional Processing

OMG : Object Management Group

OTA : Orascom Telecom Algérie

PIB : Produit Intérieur Brut

SID : Système Informatique Décisionnelle

SIM : Subscriber Identity Module

SQL : Structured Query Language

SSDT : SQL Server Data Tools

SSMC : SQL Server Configuration Manager

SSMS : SQL Server Management Studio

UML : Unified Modeling Language

XMLA : eXtended Markup Language for Analysis

WTA : Watanya Telecom Algérie

Liste des tables

Tableau 1.1 : un tableau comparatif entre les systèmes opérationnels et décisionnels	3
Tableau 2.1 : le nombre d'abonnés selon les opérateurs	18
Tableau 2.2 : la distribution des abonnés selon les technologies des opérateurs	18

Liste des figures

Figure 1.1 : Architecture typique d'un système OLAP.....	4
Figure 1.2 : la représentation schématique de l'approche Top-Down.....	7
Figure 1.3 : la représentation schématique de l'approche Bottom-up.....	7
Figure 1.4 : table de faits	10
Figure 1.5 : table de dimensions.	10
Figure 1.6 : la représentation schématique d'un modèle en étoile.	11
Figure 1.7 : la représentation schématique d'un modèle de neige en flocon.	11
Figure 1.8 : la représentation schématique d'un modèle en constellation.....	12
Figure 2.1 : comparaison entre le nombre d'abonnés des opérateurs.	19
Figure 2.2 : l'évolution du parc de la téléphonie mobile (2001-2017)	19
Figure 2.3: l'évolution du parc de la téléphonie mobile selon les technologies	20
Figure 3.1 : le profile UML	24
Figure 3.2 : Le diagramme de classe.	27
Figure 4.1 : Organigramme de travail	29
Figure 4.2 : connexion au moteur de bases de données.....	31
Figure 4.3: Création d'une nouvelle BDD.....	31
Figure 4.4: Création d'un nouveau tableau.....	31
Figure 4.5 : remplissage de données.....	32
Figure 4.6 : Insertion des données.	32
Figure 4.7 : Diagramme flocon de neige	33
Figure 4.8 : Création d'un nouveau projet	34
Figure 4.9 : Création d'un projet d'informatique décisionelle	35

Figure 4.10 : Explorateur des solutions pour le projet multidimensionnelle	35
Figure 4.11 : Sélectionner les mesures	36
Figure 4.12 : Sélectionner Une hiérarchie	36
Figure 4.13 : la déploiement de cube de données	37
Figure 4.14 : Connexion de serveur SSAS.....	37
Figure 4.15 : Architecture de IIS.....	38
Figure 4.16 : Les fichiers ISAPI de notre cube	40
Figure 4.17 : Connexion de SSAS à travers l'URL	41
Figure 4.18 : Créer un nouveau projet Java.....	43
Figure 4.19 : Créer un nouveau application java.....	43
Figure 4.20 : configure le build path	44
Figure 4.21 : ajouter des bibliothèques JAR.	44
Figure 4.22 : Le requête MDX pour le nombre des appels non null selon les villes.	45
Figure 4.23 : Exploitation par Power BI pour le nombre des appels.....	46
Figure 4.24 : diagramme de funnel pour l'exploitation par power BI.....	46
Figure 4.25 : Exploitation par Eclipse IDE	47

Introduction Générale

Les entrepôts de données associés à des outils d'analyse On-Line Analytical Processing (OLAP), représentent une solution effective pour l'informatique décisionnelle. Celle-ci désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données, et permet à un décideur d'avoir une vue d'ensemble de l'activité traitée. L'entrepôt de données permet de stocker l'ensemble de ces données dans un modèle multidimensionnel. C'est donc un stockage de données en masse, avec différentes dimensions : spatiales, temporelles, ...etc.

La modélisation de la solution d'aide à la décision à savoir OLAP, doit permettre l'optimisation de la gestion des structures de données ainsi que les différents concepts multidimensionnels à savoir les dimensions, les tables de faits, les hiérarchies, les niveaux d'agrégation, ...etc.

Le présent mémoire se propose pour comprendre la dynamique des entreprises de communication téléphonique en Algérie, et ce, par l'analyse des informations relatives à ce domaine comme : l'âge des abonnés, leur sexe, leur niveau intellectuel...etc. L'utilisation des résultats de cette analyse peut-être d'un apport considérable dans la prévision des futurs investissements. Pour cela, les gestionnaires de ces entreprises ont besoin d'outils leur permettant de calculer les indicateurs selon lesquels cette analyse peut être menée.

Dans ce travail nous avons effectué des recherches sur les opérateurs des télécommunications en Algérie. Notre contribution peut se résumer au développement d'un système d'aide à la décision pour ce domaine basée sur la technologie de l'entrepôt de données et l'analyse en ligne.

Ce mémoire est devisé en quatre chapitres. Le premier résume les concepts de bases des entrepôts de données et de l'analyse multidimensionnelle. Le deuxième chapitre donne un aperçu sur les entreprises de télécommunications en Algérie, en présentant quelques statistiques données par l'ARPT (Autorité de Régulation de la Poste et des Télécommunications). Dans le troisième chapitre nous avons présenté la modélisation du cas d'étude notamment les communications téléphoniques en Algérie, en se basant sur les diagrammes UML. Le quatrième chapitre résume les différentes étapes pour l'implémentation de l'entrepôt de données et la solution OLAP ; avant de tester notre application en utilisant un jeu de données relatives aux communication téléphoniques.

Enfin, nous allons terminer avec une conclusion générale et perspectives de recherche.

1. Chapitre1 : Concepts de base en informatique décisionnelle.

1.1.Introduction

Aujourd'hui l'informatique s'est généralisée dans tous les secteurs d'activité, ce qui a conduit les entreprises à conserver des sources d'information volumineuses et complexes, d'où la nécessité d'analyse et de consolidation de ces dernières pour une meilleure prise de décision.

Face à ce réel besoin, les entreprises ont recours à des systèmes d'aide à la décision spécifiques, basés sur l'approche des entrepôts de données qui ont fait leurs apparitions à la fin des années 1980. Dans ce chapitre, nous allons présenter les concepts et notions de base des entrepôts de données.

1.2.Informatique décisionnelle

1.2.1. Définition

L'informatique décisionnelle, aussi appelée *Business Intelligence (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 collectés *via* un service en ligne, etc. Le tout est traité par des outils d'extraction, de transfert et de consolidation (en anglais *Extrant Transform Load, ETL*) mis en place pour normaliser ces sources et établir une cohérence entre elles.

Les données sont stockées et traitées dans des *Entrepôts De Données* ou *ED* que l'on appelle aussi *Data Warehouse* ou *DWH*. Il existe également des bases plus spécifiques, les *Magasins De Données* ou *Datamart*, qui ciblent des activités précises de l'entreprise : marketing, planification budgétaire, logistique, ressources humaines... [1]

1.2.2. Le Système Opérationnel Vs Le Système Décisionnel

Le système opérationnel représente les tâches quotidiennes, répétitives et atomiques (Insertion, modification, suppression) qui sont effectuées par les employés de l'entreprise pour permettre à cette dernière d'avoir une activité et donc de survivre.

Les systèmes d'information opérationnels *OLTP (OnLine Transactional Processing)* sont des systèmes de gestion ou de production faits pour assister les opérations d'une entreprise.

Avec l'accroissement de volume des données, les systèmes opérationnels se trouvent limités devant les besoins des dirigeants qui veulent des informations synthétisées pour l'analyse et la prise de décisions d'où la naissance des systèmes d'information de type *OLAP (On-Line Analytical Processing)*. [2]

Le tableau (table 1.1) représente une comparaison entre les deux systèmes.

	OLTP	ED - OLAP
Vocation (objectif)	gestion des activités opérationnelles quotidiennes	évaluation de l'activité et aide à la décision
Utilisateurs	nombreux (des milliers), agents opérationnels	moins nombreux (des centaines), analystes et décideurs
Pattern d'utilisation	régulier, prédictible, et fréquent	irrégulier, non prédictible, et moins fréquent
Modèle de données	normalisé (3FN) et optimisé pour les exigences de performance des traitements transactionnels	dénormalisé et optimisé pour les performances des traitements analytiques
Données	opérationnelles orientées transaction, détaillées, courantes (un horizon temporel moyen de 60 à 90 jours), non redondantes et moins volumineuses	orientées analyse, moins détaillées, historiques (un horizon temporel moyen de 5 à 10 ans), redondantes et volumineuses
Mode d'accès	lecture, rajout, modification et suppression	lecture et rajout
Type de traitements	transactionnel : accès à des centaines d'enregistrements	analytique : agrégation et accès à des millions d'enregistrements
Type de technologies	optimisé pour le traitement transactionnel (temps de réponse et gestion d'accès concurrents)	optimisé pour le traitement analytique (historisation et analyse en ligne)

Tableau 1.1 : un tableau comparatif entre les systèmes opérationnels et décisionnels.[2]

1.2.3. Architecture du Système d'Information Décisionnelle (SID)

L'architecture typique des systèmes d'Information Décisionnelle est constituée de quatre couches logicielles. En amont de cette architecture se trouve les sources de données qui alimentent le système **OLAP** en données décisionnelles et qui peuvent présenter différentes hétérogénéités (e.g. plateforme technologique, encodage, conventions de nommage, unités de mesure, etc.)

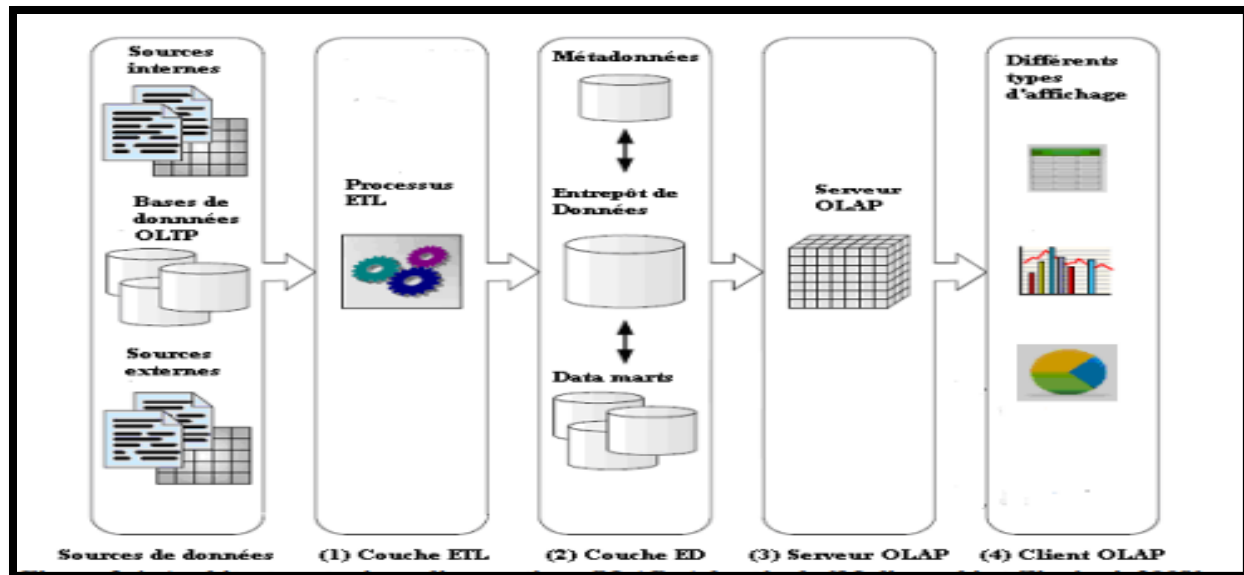


Figure 1.1 : Architecture typique d'un système OLAP. Adaptée de [2].

- **Couche d'intégration (ETL) :**

Les données décisionnelles extraites des sources subissent un ensemble de transformations pour les rendre sous un schéma unifié et homogène, qui est celui de l'*ED*. Cette importante phase d'intégration, qui peut être plus au moins complexe à réaliser dépendant du degré d'hétérogénéité des sources, incombe le plus souvent aux outils *ETL*.

De façon générale, les outils *ETL* sont des outils qui permettent la transformation d'une ou plusieurs bases de données sources vers une ou plusieurs bases de données cibles. Dans les systèmes d'*ED*, ils sont généralement utilisés pour automatiser l'intégration et la réconciliation des sources de données au sein de l'*ED*.

Dans ce cadre d'utilisation, ces outils exécutent de façon régulière un processus de 4 phases :

- ✓ **Extraction** : sélection et extraction régulière (par rapport à l'occurrence d'un événement temporel (e.g. fin du mois) ou intemporel (e.g. changement de données sources)) des données sources utiles à l'analyse décisionnelle.
- ✓ **Nettoyage** : le but de cette phase est d'améliorer la qualité des données qui peut être mauvaise dans les sources en supprimant/rectifiant certaines erreurs et inconsistances (e.g. doublons, données manquantes, données incorrectes (e.g. 30/02/2011), données inconsistantes (e.g. Alg et Algérie)).
- ✓ **Transformation** : permet la conversion des données nettoyées vers le schéma de l'*ED*.
- ✓ **Chargement** : permet le chargement des données intégrées dans l'*ED*.

- **Couche de stockage (ED) :**

Cette couche comprend l'*ED*, le référentiel de métadonnées et éventuellement un ou plusieurs magasins de données. L'*ED*, appelé également *ED primaire*, définit un lieu d'intégration et d'historisation pour toutes les données de l'organisation.

Cet **ED** peut être lui aussi la source directe à la création de petits **ED** focalisés sur des besoins analytiques particuliers (par exemple, relatifs à un groupe d'utilisateurs ou répondant à certaines exigences d'optimisation), appelées magasins de données (Datamarts). Le référentiel de métadonnées stocke des métadonnées (données sur les données) pour décrire toutes les phases du processus d'entreposage et d'exploration des données (sources de données, processus **ETL**, schémas de données, profils utilisateur, unités de mesure, etc.). Ces métadonnées peuvent être utilisées pour meilleure administration de ces processus ou servir pour l'amélioration de la compréhension des données multidimensionnelles par les décideurs. Ce référentiel doit être accessible par toutes les autres couches de l'architecture.

- **Serveur OLAP :**

Le serveur OLAP permet d'effectuer une analyse de données conforme au paradigme multidimensionnel, avec des temps de réponse optimisés [Codd et al., 1993]. Un serveur OLAP fournit aux utilisateurs une représentation multidimensionnelle des données sous forme d'un ensemble d'hypercubes et implémente un ensemble d'opérateurs **OLAP** (**Roll-up**, **Drill-down**, etc.) qui permettent d'explorer ces hypercubes.

- **Client OLAP :**

Cette couche définit une série d'interfaces utilisateurs intuitives pour l'exploration interactive et multidimensionnelle des données. Ces interfaces permettent de déclencher les opérateurs **OLAP** et présentent l'information en utilisant différents types d'affichages interactifs : tableaux croisés dynamiques, histogrammes et diagrammes statistiques. [2]

1.3. Entrepôt de données (Data Warehouse)

1.3.1. Définition

De manière plus concrète, nous pouvons le définir comme une structuré pour l'organisation des systèmes d'information. Il s'agit un process d'aide à la prise de décision et la gestion de la connaissance tant pour l'usage quotidien que pour l'élaboration de stratégies à long terme.

Il représente une mégabase, thématique le plus souvent, constituée afin d'analyser de gros volumes de données très détaillées, durables, en principe datées, et qui ont été stockées et organisées (**data sourcing**) sur un puissant système informatique.

L'objectif est de les synthétiser de manière à en extraire l'information essentielle la plus pertinente et ainsi favoriser la prise de décision. [3]

1.3.2. Les avantages d'un Entrepôt de données

L'entrepôt de données a une vision centralisée et complète de toutes les informations de l'entreprise.

C'est une structure qui a pour but de regrouper les données de l'entreprise pour des fins analytiques et pour aider à la prise de décisions stratégiques. La **décision stratégique** est une action mise en place par les décideurs de l'entreprise dans l'objectif d'améliorer, quantitativement ou qualitativement, la performance de l'entreprise dans un ensemble d'informations organisées, historisées et provenant de différentes sources de données.

Lorsqu'on réalise un **schéma de Base de Données** pour un système d'information, on parle de « *tables* » et de « *relations* ».

Une **table** est une représentation d'une entité. Une **relation** est une technique pour lier ces entités entre elles. En terme **BI**, on parle de « *Dimension* » et de « *Faits* ». [3]

1.3.3. Les caractéristiques d'Entrepôt de données

L'entrepôt de données n'est pas une simple copie des données de production, il est organisé et structuré.

- **Orienté sujet** : Au cœur du **ED**, les données sont organisées par thème et ils propres à un thème, les ventes par exemple, seront rapatriées des différentes bases **OLTP** de production et regroupées.
- **Intégré** : Les données proviennent de sources hétérogènes utilisant chacune un type de format. Elles sont intégrées avant d'être proposées à utilisation.
- **Non volatile** : Les données ne disparaissent pas et ne changent pas au fil des traitements, au fil du temps (**Read-Only**).
- **Historisé** : Les données non volatiles sont aussi horodatées. On peut ainsi visualiser l'évolution dans le temps d'une valeur donnée. 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. [3]

1.3.4. Structure d'un data Warehouse

Le datawarehouse se structure en trois classes de données, organisées selon un axe historique.

- **Les données détaillées** :

Elles reflètent des évènements les plus récents. Les données provenant des systèmes de production sont intégrées à ce niveau.

- **Les données agrégées** :

Elles correspondent à des éléments d'analyse représentatifs des besoins des utilisateurs. Ce sont donc des données déjà traitées par le système et représentant un premier résultat d'analyse et de synthèse des données contenues dans les systèmes de production. Elles doivent être facilement accessibles et compréhensibles.

- **Les méta-données** :

Il s'agit « de données sur les données ». Elles décrivent les règles ou processus attachés aux données du système. Les méta-données permettront notamment de connaître :

- ✓ Quelles sont les données entreposées, leur format, leur signification, leurs degrés d'exactitude.
- ✓ Les processus de récupération/extraction dans les bases sources.
- ✓ La date du dernier chargement du datawarehouse.
- ✓ L'historique des données sources et de celles du datawarehouse. [4]

1.3.5. Les 2 approches pour construire un Data Warehouse

Nous allons ici entreprendre une analyse comparative des deux approches, tout en mettant en avant les critères significatifs de choix.

- **L'approche d'Inmon (approche Top-Down) :**

Dans son approche, le Data Warehouse est un référentiel centralisé d'entreprise (ou *CIF*) stockant l'information au niveau le plus détaillé. Des Datamarts modélisés sous forme de *schéma en étoile* sont ensuite créés à partir de ce Data Warehouse.

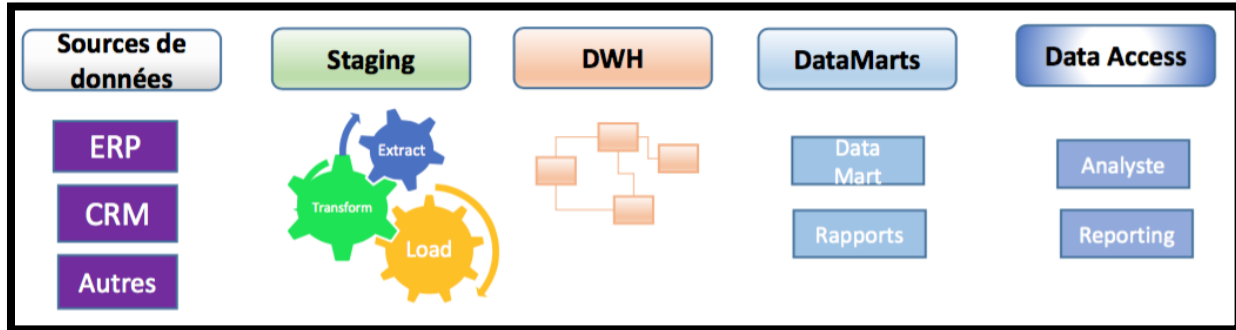


Figure 1.2 : la représentation schématique de l'approche Top-Down. [3]

- **L'approche de Kimball (ou approche Bottom-Up) :**

Son approche s'oppose à celle de *William H. Inmon*, affirmant qu'un *ED* doit être rapide et compréhensible.

Le *ED* peut être vu, selon l'auteur, comme l'union des datamarts cohérents entre eux grâce aux dimensions conformes (*Data Warehouse Bus*). *La Modélisation Dimensionnelle* permet un requêtage facile et performant pour les utilisateurs.

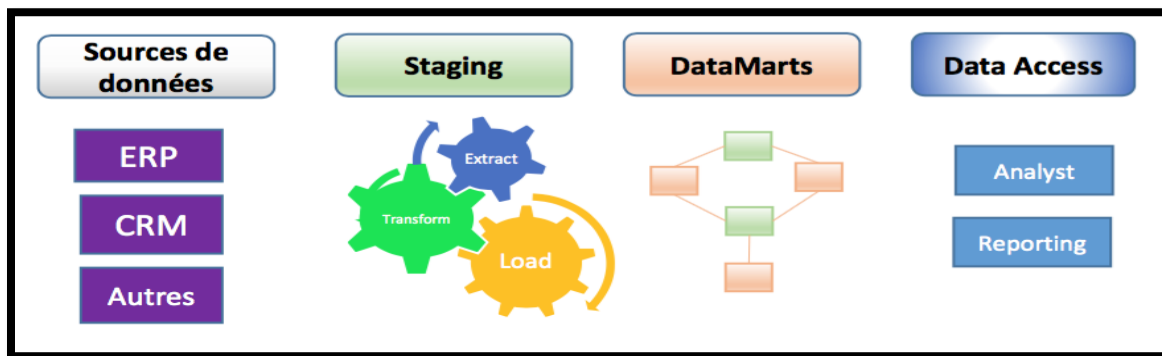


Figure 1.3 : la représentation schématique de l'approche Bottom-up. Adaptée de [5]

1.4. OnLine Analysis Processing (OLAP)

1.4.1. Définition

OLAP permet aux utilisateurs d'analyser les informations de base de données à partir des systèmes de bases de données multiples à un moment donné. Bien que les bases de données relationnelles soient considérées comme deux dimensions, les données *OLAP* est multidimensionnelle, ce qui signifie l'information peut être comparée de différentes façons.

1.4.2. Les 12 règles OLAP

Les 12 règles telles que définies par E.F Codd sont les suivantes [6] :

1. **Multidimensional Conceptual View** : Permet d'avoir une vision multidimensionnelle des données (ce qui n'est pas le cas avec une table – unidimensionnel)
2. **Transparency** : L'utilisateur ne doit pas se rendre compte de la provenance des données si celles-ci proviennent de sources hétérogènes (système homogène à l'analyste) ; ces sources peuvent être un fichier Excel, une base de données de production ou même un fichier texte.
3. **Accessibility** : OLAP est décrit comme un middleware qui se place entre les sources de données hétérogènes et un **front-end** (sous la forme d'un datawarehouse).
4. **Uniform Reporting Performance** : Les performances ne doivent pas être diminuées lors de l'augmentation du nombre de dimension ou de la taille de la base de données, mais proportionnelles à la taille des réponses retournées.
5. **Client-Server Architecture** : l'essentiel que le produit soit Client-Serveur mais aussi que les composants serveurs d'un produit **OLAP** intègrent facilement ses différents clients.
6. **Generic Dimensionality** : Chaque dimension doit être équivalente par rapport à sa structure et ses capacités opérationnelles.
7. **Automatic Adjustment of Physical Level** : le système **OLAP** ajuste automatiquement son schéma physique pour s'adapter au type du modèle et au volume des données (plus on dispose de place plus on peut agréger).
8. **Multi-User support** : les outils **OLAP** doivent fournir des accès concurrents, l'intégrité et la sécurité.
9. **Unrestricted Cross-dimensional Operations** : Les calculs doivent être possibles à travers toutes les dimensions (les agrégats doivent être faits dans toutes les dimensions).
10. **Intuitive Data Manipulation** : La manipulation des données se fait directement à travers les cellules d'une feuille de calcul, sans recourir aux menus ou aux actions multiples. Il doit permettre l'analyse intuitive dans plusieurs dimensions au final.
11. **Flexible Reporting** : Lors de la création de rapports, les dimensions peuvent être présentées de n'importe quelle manière.
12. **Unlimited Dimensions & Aggregation Levels** : Dimensions et niveaux d'agrégations illimités.

1.4.3. Cube OLAP

1.4.3.1. Définition

Un cube **OLAP** est une base de données à plusieurs dimensions, optimisée pour les applications d'entrepôt de données et de traitement analytique en ligne (**OLAP**, Online Analytical Processing).

Un cube **OLAP** est une méthode de stockage de données sous forme multidimensionnelle, généralement à des fins de génération de rapports.

Dans les cubes **OLAP**, les données (ou mesures) sont classées par dimensions. Les cubes **OLAP** sont souvent pré-synthétisés entre les dimensions, ceci afin d'accélérer considérablement l'interrogation par rapport aux bases de données relationnelles.

Le langage de requête utilisé pour interagir avec les cubes *OLAP* est *MDX (MultiDimensional Expressions)*. Il repose sur des expressions à dimensions multiples. Développé initialement à la fin des années 90 par **Microsoft**, le langage *MDX* a été adopté par de nombreux autres éditeurs de bases de données multidimensionnelles.

Si un cube *OLAP* peut stocker des informations comme le ferait une base de données traditionnelle, sa structure est radicalement différente. Historiquement, les bases de données sont conçues selon les exigences des systèmes informatiques qui les utilisent. Les cubes *OLAP*, en revanche, sont exploités par des utilisateurs métier dans une optique d'analytique avancé.

Aussi sont-ils conçus à l'aide d'une intelligence et d'une logique de l'activité. Ils sont optimisés pour des objectifs analytiques, et peuvent ainsi générer des rapports sur des millions d'enregistrements simultanés. Les utilisateurs métier peuvent interroger des cubes *OLAP* en utilisant le simple langage humain. [6]

1.4.3.2. Opérations sur le cube

- **Roll-up** : permet d'agréger les valeurs de mesure en montant dans une hiérarchie de dimension. Par exemple, le Roll-up du niveau "Mois" vers "Année", permet de grouper et d'agréger les ventes mensuelles par année en utilisant par exemple la fonction Sum.
- **Drill-down** : c'est l'opération inverse à Roll-up, elle augmente le niveau de détail de la mesure en descendant dans la hiérarchie de dimension.
- **Slice** : effectue une sélection des cellules de l'hypercube en utilisant une condition (prédicat) définie sur les membres d'une dimension ; il retourne un sous hypercube de l'hypercube initial.
- **Dice** : effectue une sélection en utilisant une condition définie sur deux dimensions ou plus.
- **Projection** : permet de sélectionner un sous-ensemble de mesures de l'hypercube.
- **Rotate** : effectue un pivotage des axes de l'hypercube pour obtenir différentes alternatives de présentation des données décisionnelles.

1.4.4. Les différentes variantes de l'OLAP

Il existe différentes variantes de l'*OLAP*. Vous avez la possibilité de spécifier la façon dont vous souhaitez que votre cube stocke les données. Chaque variante a un impact sur la performance du cube.

- **MOLAP** : le multidimensionnel *OLAP* stocke aussi bien les données que les agrégats dans une structure multidimensionnelle. En conséquence, les requêtes sont très performantes et les temps de réponse sont significativement réduits. L'inconvénient c'est que le traitement de la partition est consommateur de ressources.
- **ROLAP** : le relationnel *OLAP* stocke aussi bien les données que les agrégats dans la base de données relationnelle source. Il est très lent pour répondre aux requêtes car un rapport *ROLAP* est une requête *SQL*. Les requêtes sont reconverties en langage *SQL*. Le traitement est léger car moins de consommation de ressources.
- **HOLAP** : l'hybride *OLAP* stocke les données dans la base relationnelle source et les agrégats dans une structure multidimensionnelle. Il présente un compromis entre *MOLAP* et *ROLAP*. Il combine les avantages des de *MOLAP* et *ROLAP*. Les temps de

réponse dépendent donc des requêtes et des données à récupérer. Par défaut, c'est la variante *MOLAP* qui est utilisée pour un cube *SSAS (SQL Server Analysis Services)*. [6]

1.5. Modélisation multidimensionnelle

1.5.1. Définition

La construction d'un entrepôt de données est réalisée à partir de la modélisation multidimensionnelle. Ces derniers sont organisés de manière à mettre en évidence le sujet de l'analyse (*la table des faits*) et les différentes perspectives de l'analyse (*les tables des dimensions*) qui symbolisent les différentes valeurs de l'activité analysée.

- ✓ **Les faits** : les faits représentent des sujets d'analyse.

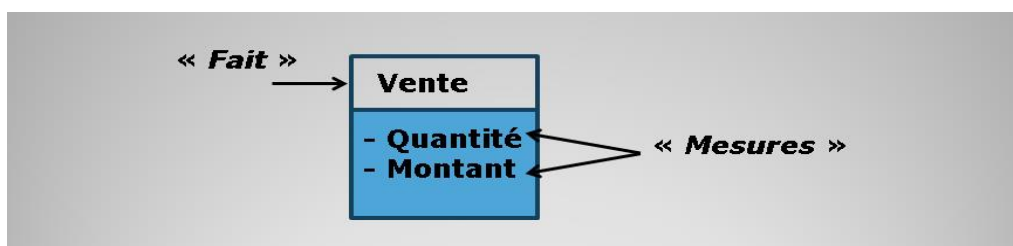


Figure 1.4 : table de faits

- ✓ **Les dimensions** : les dimensions sont les critères selon lesquels on souhaite évaluer, quantifier les faits.

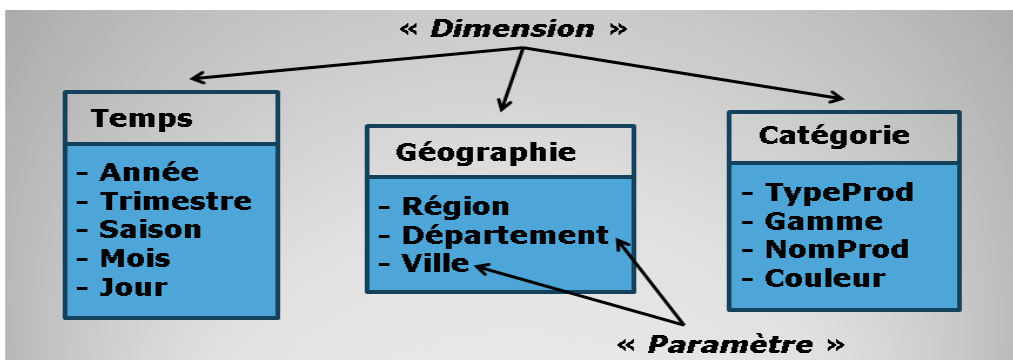


Figure 1.5 : table de dimensions.

1.5.2. Les types de schémas

- **Schéma en étoile** :

Le modèle en étoile centre une table des faits et la relie à chaque table de dimension ou axe d'analyse.

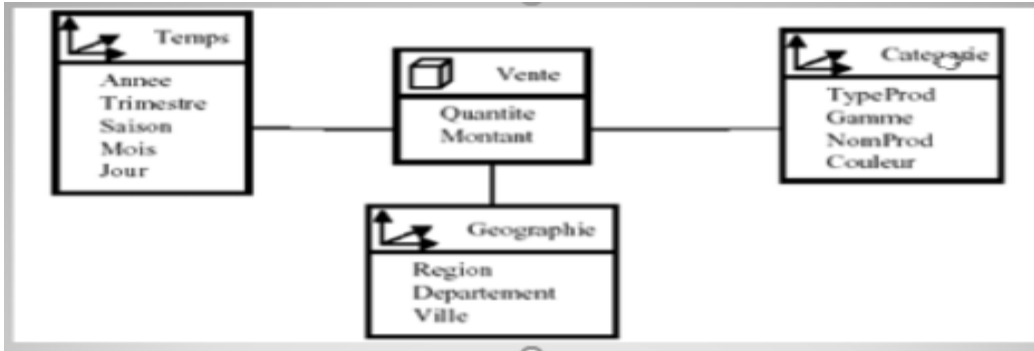


Figure 1.6 : la représentation schématique d'un modèle en étoile. Adaptée de [4]

- **Schéma en flocon de neige :**

Le schéma en flocon est dérivé du schéma en étoile, il se compose d'une relation centrale entourée des différentes tables de dimension qui sont normalisées. Avec ce schéma, les dimensions sont éclatées ou décomposées en sous hiérarchies. L'expression « hiérarchies alternatives » est utilisée pour désigner les hiérarchies différentes et parallèles d'une même dimension.

Ce modèle minimise le volume des tables à joindre lors de l'exécution des requêtes par rapport au schéma en étoile mais le modèle de données est plus complexe que le schéma en étoile car comportant un nombre plus élevé de tables.

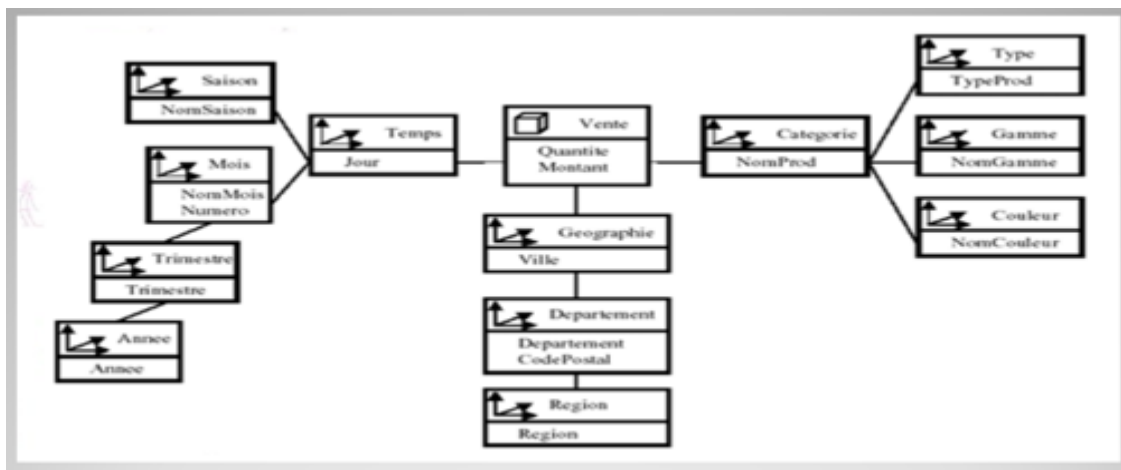


Figure 1.7 : la représentation schématique d'un modèle de neige en flocon. Adaptée de [4]

- **Le modèle en constellation :**

Le modèle en constellation rassemble plusieurs tables des faits qui utilisent les mêmes dimensions.

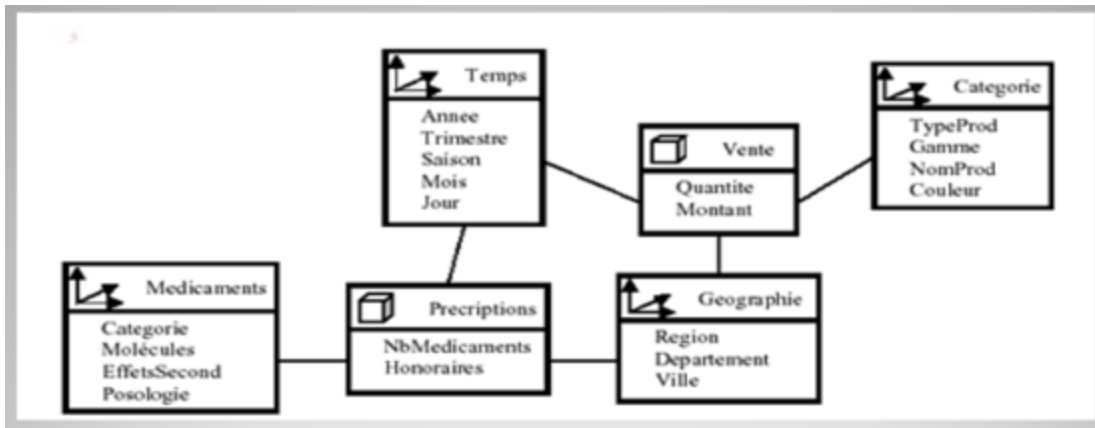


Figure 1.8 : la représentation schématique d'un modèle en constellation. Adaptée de [4]

1.6. Les solutions disponibles

1.6.1. Open Source

Voir quelques outils open source d'informatique décisionnelle :

- ✓ **Mondrian** : Mondrian est un serveur **OLAP** écrit en langage **Java**. Mondrian utilise le langage d'interrogation **MDX**. Mondrian, précurseur du décisionnel Open source, est désormais intégré au projet **Pentaho**.
- ✓ **Palo** : Commercialisée par l'éditeur **Jedox**, Palo OLAP Server est un serveur de bases de données multidimensionnelles de type **MOLAP** dédié à la gestion décisionnelle pour le planning, l'analyse, le reporting et la consolidation des données.
- ✓ **Spago BI** : est une solution de Business Intelligence open Source italienne. Il permet de faire des analyses multi-dimensionnelle comme Mondrain.

1.6.2. Propriétaire

Voir quelques outils propriétaires d'informatique décisionnelle :

- ✓ **IBM Cognos** : est un logiciel intégré pour l'élaboration des prévisions, des plans et des budgets, permettant de piloter la performance de l'entreprise.
- ✓ **Oracle** : Oracle propose, depuis la version 9i, un moteur **OLAP** directement intégré à sa base de données relationnelle.
- ✓ **SQL Server (Analysis Services)** : Microsoft SQL Server 2005 Analysis Services (**SSAS**) fournit des fonctions **OLAP** d'exploration de données pour les applications décisionnelles. Analysis Services prend en charge **OLAP** en permettant de concevoir, de créer et de gérer des structures multidimensionnelles qui contiennent des données agrégées provenant d'autres sources de données, telles que des bases de données relationnelles.

1.7.Conclusion

Dans ce chapitre, des notions et des concepts importants sur les entrepôts de données ont été présentés, décrivant la méthode de modélisation d'un entrepôt de donnée et les différentes stratégies d'implémentation ainsi que les différents modèles multidimensionnels pour sa construction, montrant ainsi l'impact de ces derniers sur la prise de décision au sein des entreprises.

Dans le prochain chapitre, nous introduisons le domaine des communications téléphoniques.

2. Chapitre 2 : La communication téléphonique en Algérie

2.1.Introduction.

Le téléphone est un outil très important dans la vie quotidienne des personnes et des entreprises. C'est la principale source de contact entre une entreprise et ses clients. Dans ce chapitre, nous allons présenter quelques notions de base sur les communications téléphoniques, ainsi que les statistiques concernant les opérateurs prestataires des services téléphoniques en Algérie.

2.2.La communication téléphonique

2.2.1. Définition

La communication, est l'action de communiquer, de transmettre des informations ou des connaissances d'une personne à autre (ex : le dialogue). On utilise une chaîne de communication qu'est constituée de l'émetteur, du récepteur, du message, du code (ex : la langue), le canal de transmission (ex : de vive voix), le contexte. Du latin *communicare*, mettre en commun, faire part de, partage, dérivé de *communis*, commun.

La téléphonie signifie transmettre la voix humaine entre deux lieux distants l'un de l'autre. Du deux mots grecs : *têle* et *phônê*, littéralement (voix à distance).

La communication téléphonique, est un échange oral entre deux ou plusieurs personnes via le téléphone, ou un autre moyen de télécommunication.

2.2.2. Les moyens de communication téléphonique

Avec l'évolution des moyens de communication, on est convaincu de démontrer certains nouveaux outils ou médias de communication externe pour l'entreprise, et surtout de service : on peut citer certains moyens comme le téléphone fixe, téléphone sans fil, le portable, le smartphone, tablette, ordinateurs, wearables...etc.

2.3.Les opérateurs téléphoniques en Algérie

L'Algérie compte trois principales opérateurs téléphoniques, cette liste récapitule les différents tarifs pratiqués chez ces derniers et les différentes offres qui y sont disponibles.

2.3.1. Définition d'opérateur

Opérateur, désigne le titulaire licence d'établissement et/ou d'exploitation d'un réseau public de télécommunications de différents générations et/ou d'exploitation de services téléphoniques et/ou de données en Algérie. [7]

2.3.2. Les Dénominations commerciales des opérateurs

2.3.2.1. Djezzy

Djezzy est un opérateur de télécommunications algérien a été créé en juillet 2001. Avec plus de 16,5 millions d'abonnés au mois de décembre 2016. Djezzy est la dénomination commerciale retenue pour représenter le réseau *GSM* d'*Optimum Telecom Algérie* (orascom télécom Algérie anciennement).

Optimum Telecom Algérie est une filiale d'*OTH (Orascom Telecom Holding)* qui a décroché en *juillet 2001*, la deuxième licence de téléphonie mobile en Algérie pour un montant de *737 millions* de dollars. Le 04 Octobre 2010, *OTH* et *VimpelCom* (opérateur *Russo-norvégien*) ont, annoncé leur fusion, *OTH* est désormais partie intégrante de VimpelCom. Avec plus de *2.5 milliards USD* d'investissement depuis 2001 à ce jour, fort d'un capital humain de plus de 4000 employés, plus de 17 millions d'abonnés et plusieurs offres et solutions.

L'entreprise est dirigée par **Vincenzo Nesci** Président Exécutif et **Matthieu Galvani**, Directeur Général. [8]

2.3.2.2. Ooredoo

Premier opérateur multimédia de téléphonie mobile en Algérie, *Nedjma*, devenue *Ooredoo* le *21 nov 2013*, est la filiale algérienne du *Groupe Ooredoo* (anciennement *Qatar télécom (Qtel)*).

Présent en *Algérie* depuis le *23 décembre 2003*, date d'obtention de la licence de fourniture des services de téléphonie mobile en Algérie, la marque *Nedjma* a été commercialement lancée le *24 août 2004*, en offrant aux Algériens, qu'ils soient clients particuliers ou entreprises, une gamme d'offres et de services novateurs, en respect avec les standards internationaux.

Groupe Ooredoo est une compagnie internationale leader des télécommunications qui fournit les services de téléphonie mobile, fixe et l'Internet haut débit et les services Entreprise adaptés aux besoins des particuliers et des entreprises à travers les marchés du Moyen Orient, d'Afrique du Nord et du Sud-Est asiatique. Etant une compagnie orientée vers les populations, dont les actions sont cotées à la Bourse du Qatar « *Qatar Exchange* » et à la Bourse d'Abou Dhabi, « *Abu Dhabi Securities Exchange* ».

Sheikh *Abdullah Bin Mohammed Bin Saud Al-Thani* est président du Conseil d'Administration de *Qatar Telecom (Qtel)*. Après l'acquisition de *Qtel Koweït - Wataniya*, qui était considérée à l'époque comme le plus important contrat de télécommunications dans le monde arabe, Sheikh Abdullah est devenu président de *Wataniya*. [9]

2.3.2.3. Algérie télécom

Algérie Télécom est le leader sur le marché Algérien des télécommunications qui connaît une forte croissance. Offrant une gamme complète de services de voix et de données aux clients résidentiels

et professionnels. *Algérie Télécom*, est une société par actions à capitaux publics opérant sur le marché des réseaux et services de communications électroniques.

Sa naissance a été consacrée par la loi dans le **05 août 2000**, fixant les règles générales relatives à la poste et aux télécommunications ainsi que les résolutions du conseil national aux participations de l'Etat (*CNPE*) du **1^{er} Mars 2001** portant sur la création d'une Entreprise Publique Economique dénommée « *Algérie Télécom* ».

Algérie Télécom est donc régie par cette loi qui lui confère le statut d'une entreprise publique économique sous la forme juridique d'une société par actions *SPA* au capital social de **50.000.000.000 Dinars** et inscrite au centre du registre de commerce le **11 mai 2002**. mis en œuvre un plan de filialisation des activités liées au mobile et satellite qui s'est traduit par sa transformation en groupe auquel sont rattachées trois filiales :

- *Algérie Télécom Mobile ATM (Mobilis)* : société par actions au capital social de 100.000.000 DA, pour la téléphonie mobile.
- *Algérie Télécom Satellite ATS (RevSat)* : société par actions au capital social de 100.000.000 DA, pour le réseau satellitaire.
- *Algérie Télécom Internet ATI (Djaweb)* : pour le réseau internet. Algérie Télécom a comme activité principale la téléphonie fixe et pour laquelle elle est actuellement en position de monopole. Elle intervient aussi dans le mobile avec sa filiale *Mobilis*, via les satellites : *VSAT, INMARSAT, GMPCS Thuraya*. [9]

2.4.Indicatif téléphonique en Algérie

Pour téléphoner en *Algérie* depuis l'étranger, deux informations vous seront utiles. Il vous faudra d'abord composer l'indicatif national : **+213**. Une fois cet indicatif composé un autre indicatif correspondant à la région (pour téléphone fixe) ou correspondant à l'opérateur (pour téléphone portable) vous sera nécessaire.

Par exemple, pour téléphoner à *Mostaganem*, il vous faudra composer le 00 213 45 xx xx xx (où xx représente le numéro de votre contact qui se compose de 6 chiffres). On a donc dans cet indicatif '213' correspondant à l'indicatif national suivi de '45' qui lui fait référence à l'indicatif régional.

- ✓ Pour téléphoner à Mobilis, il vous faudra composer le 00 213 6 xx xx xx xx
- ✓ Pour téléphoner à Ooredoo, il vous faudra composer le 00 213 5 xx xx xx xx
- ✓ Pour téléphoner à Djezzy, il vous faudra composer le 00 213 7 xx xx xx xx

2.5. Marché de téléphonie en Algérie

2.5.1. Un aperçu sur le marché

Avec ses **41,2 M** d'habitants, l'Algérie est le marché le plus vaste d'*Afrique du Nord*. C'est aussi le plus grand pays d'Afrique de par sa superficie de **2.381.741 km²**. 4^{ème} **PIB (Produit Intérieur Brut)** en Afrique, la dette extérieure de l'Algérie est inférieure que 2% du **PIB**. Le marché des télécommunications présente un très fort potentiel de développement.

Algérie Télécom, opérateur historique, détient le monopole de la téléphonie fixe. Les 3 opérateurs détiennent une licence de téléphonie Mobile : **Mobilis** (Public), **DJEZZY** (51% public et 49 % **VEON**), **OOREDOO** (privé). [7]

2.5.2. L'Autorité de Régulation de la Poste et des Télécommunications (ARPT)

L'ARPT, l'autorité de régulation des télécoms Algérien vient d'entamer un changement dans son mode de fonctionnement à commencer par le changement radical de son nom qui devient désormais « Autorité de Régulation de la Poste et des Communications Electroniques (**ARPCE**) ».

L'Autorité de Régulation de la Poste et des Télécommunications (**ARPT**) a été créée dans le cadre de la libéralisation du marché postal et des télécommunications. Leur ouverture à la concurrence et à la promotion de la participation de l'investissement privé dans ces marchés ont été consacrés par la loi n°2000-03 du 5 août 2000 fixant les règles générales relatives à la poste et aux télécommunications.

Selon certains spécialistes, de nombreux pays africains ont entamé des changements de ce genre cette année Et enfin le mot « télécommunications » ne reflète plus la réalité du terrain puisqu'il s'agit désormais de « communications électroniques ». [7]

2.5.3. Statistiques & graphes d'ARPT

On va donner quelques statistiques et graphes qui publié en l'année 2018, par le site officiel d'Autorité de Régulation de la Poste et des Télécommunications d'Algérie (<https://www.arpce.dz>) :

Le tableau (Table 2.1) donne la distribution des abonnés selon les trois opérateurs d'Algérie

Opérateurs mobiles	2015	2016	2017
ATM	14 087 440	16 885 490	18 365 148
OTA	17 005 165	16 360 904	14 947 870
WTA	12 298 360	12 571 452	12 532 647
Total abonnés	43 390 965	45 817 846	45 845 665
Pénétration aux réseaux mobiles	107,40%	111%	109%

Table 2.1 : le nombre d'abonnés selon les opérateurs. [7]

Le tableau (Table 2.2) donne la distribution des abonnés selon les technologies des trois opérateurs d'Algérie

Année	2017		
Opérateur	ATM	OTA	WTA
GSM	32,36%	41,89%	25,75%
3G	47,26%	25,30%	27,44%
4G	35,53%	35,06%	29,41%
Part de marché globale	40,06%	32,60%	27,34%

Table 2.2 : la distribution des abonnés selon les technologies des opérateurs. [7]

La figure (Figure 2.1) représente un graphique qui donne une comparaison entre le nombre d'abonnés des opérateurs d'Algérie

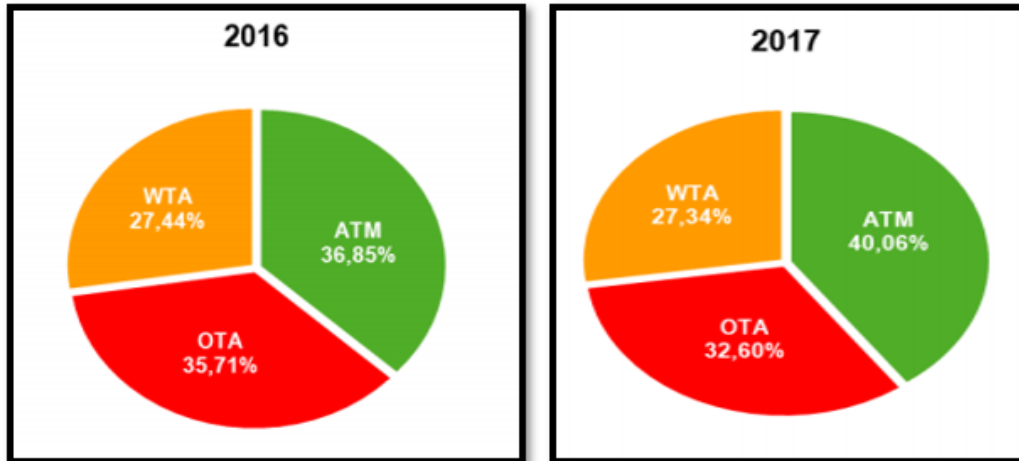


Figure 2.1 : comparaison entre le nombre d'abonnés des opérateurs. [7]

La figure (Figure 2.2) représente l'évolution du parc de la téléphonie mobile depuis 2001 jusqu'à 2017



Figure 2.2 : l'évolution du parc de la téléphonie mobile (2001-2017)

La figure (Figure 2.3) représente l'évolution du parc de la téléphonie mobile selon les technologies depuis 2001 jusqu'à 2017.

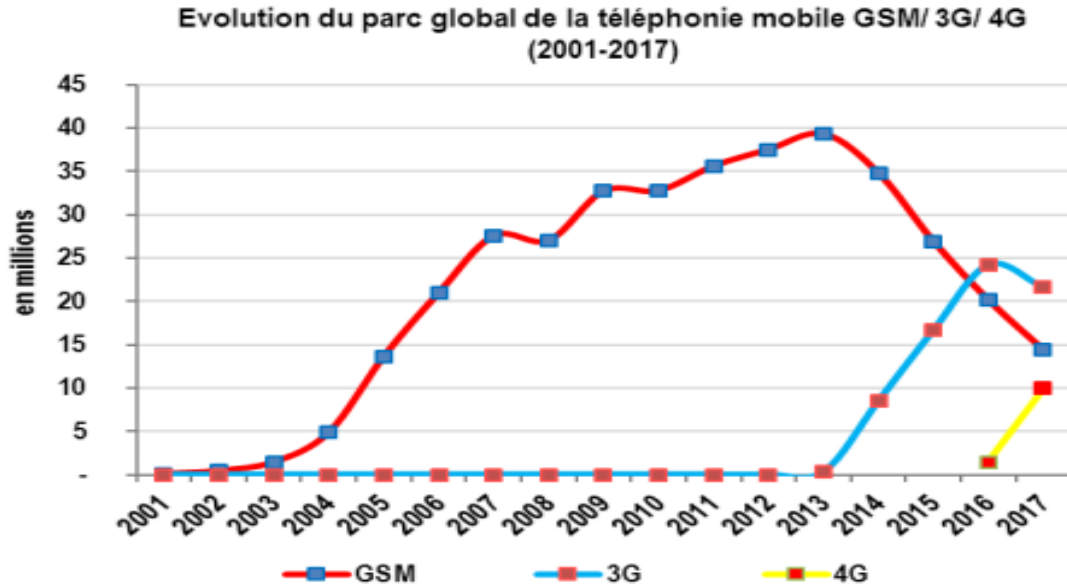


Figure 2.3: l'évolution du parc de la téléphonie mobile selon les technologies (2001-2017)

2.6. Conclusion

Dans ce chapitre nous avons présenté des notions sur les télécommunications et les statistiques sur les opérateurs téléphoniques en Algérie.

Dans le prochaine chapitre nous allons entamer à la modélisation de la solution de système d'aide à la décision

3. Chapitre3 : Modélisation

3.1.Introduction

Dans ce chapitre nous allons présenter la modélisation UML du cas des communications téléphoniques. Nous allons donc présenter le profile UML de l'entrepôt de données puis le diagramme de classes du modée focon de neige de l'application OLAP .

3.2.Quelques définitions

- **Unified Modeling Language (UML)** : Est un langage graphique de modélisation pour visualiser, spécifier, construire et documenter les artefacts d'un système logiciel, *UML* intègre et étend les concepts et la notation des méthodologies de développement orienté objet.
- **Les diagrammes d'UML** : Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. Les différents types de diagrammes d'*UML* sont présentés dans l'extrait du métamodèle .
- **Papyrus** : est un outil d'édition graphique pour *UML* développé principalement par le content enterprise management (*CEA*) ,Papyrus est composé de plusieurs éditeurs principalement graphiques, complétés par d'autres types d'éditeurs tels que les éditeurs textuels ou arborés. L'ensemble de ces éditeurs permettent la visualisation simultanée de plusieurs diagrammes où chacun pouvant exprimer un point de vue différent.
- **OCL (Object Constraint Language)** : Est un langage formel pour l'expression de contraintes et de requêtes sur des diagrammes UML (diagrammes de classes, états-transitions, etc.) . Ce standard adopté par l'**OMG** présente de nombreux avantages .

OCL a été conçu par *IBM* autour de **1995** comme un langage de modélisation métier (Business modeling language) à la portée des concepteurs de systèmes d'information; la syntaxe et la sémantique de ses expressions ont été formalisées afin d'éviter les ambiguïtés des langages naturels.

- **Le diagramme de classes** : Est le point central dans le développement orienté objet, il représente la structure interne du système sous forme de classes et les relations entre eux. Les classes constituent la base pour la génération de code et la génération des schémas des bases de données.
- **Le méta-modèle** : Selon *MOF (Meta Object Facility)*, un *méta-modèle* définit la structure que doit avoir tout modèle conforme à ce Méta-modèle. Autrement dit, tout modèle doit respecter la structure définie par son méta-modèle. Par exemple, le méta-modèle UML définit que les modèles UML contiennent des packages, leurs packages des classes, leurs classes des attributs et des opérations, etc.

Les méta-modèles fournissent la définition des entités d'un modèle, ainsi que les propriétés de leurs connexions et de leurs règles de cohérence. *MOF* les représente sous forme de diagrammes de classes.

- **Les faits et les dimensions** : Dans un diagramme de classes, l'information est séparée en faits et dimensions, représentés respectivement par des classes de faits et des classes de dimensions. Les classes de faits se composent de deux types d'attributs :

Des *faits attributaires*, qui représentent des mesures (les opérations ou les valeurs en cours d'analyse).

Les *dimensions dégénérées* qui permettent au concepteur de l'entrepôt de données de représenter d'autres faits, en plus des mesures pour l'analyse.

3.3. Les approches de modélisation multidimensionnelle basés sur l'UML

Le modèle multidimensionnel utilisé dans ce travail est un modèle en flocon de neige qui est basé sur le *modèle en étoile* (le centre est la *table des faits*, et les branches en sont les *dimensions*). où les tables de dimension sont normalisées.

Pour présenter les propriétés principales structurelles de la modélisation multidimensionnelle, les auteurs de ce travail ont basé sur les *diagrammes de classe* d'une part, et les *stéréotypes* (faits, dimensions, descripteurs, etc.) et d'autres bases d'une autre part.

3.3.1. Extension de l'UML

L'*UML* est un langage général unifié applicable à différents types de systèmes, des *domaines*, des *méthodes* et des *processus*. Cependant, il peut y avoir des situations dans lesquelles l'utilisateur pourrait vouloir étendre le langage pour une utilisation souple et enrichie tout en ajoutant de nouveaux éléments de construction (*stéréotypes, valeurs, contraintes ...*).

3.3.2. Modèles UML

Le modèle est vu comme un étoile stéréotype « **StarPackage** » contient un ensemble de dimensions « **DimensionPackage** » et un paquetage de faits « **FactPackage** » qui à son tour contient un fait « **Fact** » ou plusieurs faits.

Le fait contient un ensemble de mesures ou d'attributs de fait « **FactAttribute** » et un ensemble de dimensions ; et chaque niveau d'agrégation peut contenir différents types d'attributs : un identifiant, un attribut particulier utilisé pour visualiser ses membres dans les outils OLAP (**Descriptor**).

Les niveaux d'agrégation sont reliés par des relations stéréotypées Rolls-up (spécialisations d'associations UML) pour former les hiérarchies de dimension.

Les différents stéréotypes sont formalisés comme des spécialisations de métaclasses UML (e.g. le stéréotype Base spécialise la métaclasse UML **class**) sur lesquelles sont définies des contraintes **OCL** pour éviter leur mauvaise utilisation par les concepteurs.

3.3.3. Le profil UML

Est constitué d'éléments qui étendent le métamodèle *UML* et qui permettent de spécialiser les notions de *UML*. En plus d'expliquer chaque étape de construction d'un profil.

À l'aide de profils, les métaclasses *UML* existantes (par exemple, une propriété, une classe et une association) peuvent être personnalisées à l'aide de trois mécanismes : stéréotypes, valeurs étiquetées et contraintes.

Un stéréotype (**Stereotype**) est une extension d'une métaclasse UML qui définit des valeurs balisées supplémentaires, contraintes et éventuellement une nouvelle notation graphique (icône). Au niveau du modèle, il peut être visualisé de deux manières : en utilisant son icône ou une chaîne représentant son nom entre une paire de guillemets « **Stereotype name** ».

Les valeurs marquées (**Tagged values**) représentent les propriétés des stéréotypes. Au niveau du modèle, chaque valeur étiquetée est attachée à un élément du modèle auquel un stéréotype est appliqué. Une valeur étiquetée est représentée par son nom et une valeur associée issue de son domaine de valeurs et est placée entre une paire d'accolades {tagged value name = value}.

Les contraintes (**Constraints**) affinent les définitions des stéréotypes et des valeurs étiquetées et empêchent leur utilisation arbitraire par les concepteurs lors de la définition des modèles. Ils peuvent être utilisés, par exemple, pour restreindre la valeur d'attribut domaines ou les associations de stéréotypes. Ces contraintes sont souvent formalisées avec **OCL**.

- **Description**

Dans notre approche, les aspects statiques et dynamiques des modèles *MD* sont modélisés de manière déclarative à l'aide d'éléments de diagramme de classes *UML* (par exemple, classe, package, propriété et opération).

En plus des exigences bien connues (telles que la modélisation des mesures et des attributs dimensionnels en tant que propriétés *UML*, des faits et des niveaux d'agrégation en tant que classes *UML* et de plusieurs hypercubes et dimensions en tant que packages), nous représentons explicitement les fonctions d'agrégation et les règles d'agrégation par des opérations, des indicateurs par classes et hiérarchies par packages. Nous utilisons des packages pour organiser les constructions en différents niveaux de détail. Cette organisation facilite la compréhension de modèles complexes.

Notre proposition est formellement définie comme un profil *UML* implémenté dans l'outil Papyrus. Notre profil définit un ensemble de stéréotypes et de valeurs étiquetées pour la modélisation conceptuelle des cubes de données. Dans le package de contraintes, un ensemble de contraintes **OCL** est défini pour empêcher l'utilisation arbitraire des stéréotypes et des valeurs balisées de notre profil. Nous présentons dans cette section des contraintes structurelles inhérentes aux profils représentatifs.

Nous représentons le métamodèle du package de modèle principal dans (la figure 3.1.)

Chaque model « **Model** » est considéré comme un ensemble fini d'hypercubes non vides « **Hypercube** ».

Les hypercubes représentent des sujets d'analyse. Formellement, chaque hypercube est constitué d'un (éventuellement vide) ensemble fini de mesures « **Measure** » regroupées dans une classe de faits « **Fact** » et un ensemble fini de dimensions « **Dimension** » non vide.

Les mesures sont représentées par « **Measure stereotype** », défini comme une extension de la **Metaclass** de la propriété **UML** (voir la figure 3.1). Les dimensions sont représentées par le stéréotype **Dimension**, qui étend **Metaclass** du package **UML**. Chaque dimension est définie par un ensemble fini de hiérarchies non vides (voir la figure 3.1).

Les hiérarchies permettent d'analyser des faits à différents niveaux de détail. Formellement, chaque hiérarchie « **Hierarchy** » consiste en un ou plusieurs niveaux d'agrégation connexes qui forment un graphe acyclique dirigé avec exactement un niveau inférieur et plusieurs niveaux supérieurs.

Toutes les hiérarchies d'une dimension ont le même *niveau inférieur*, ce qui devrait être *lié au fait* de l'hypercube.

Les niveaux d'agrégation « **Level** » définissent des niveaux de visualisation intéressants pour les mesures en fonction des exigences de l'analyse multidimensionnelle.

Chaque niveau d'agrégation (**Aggregation level**) peut contenir plusieurs attributs dimensionnels « **DimensionalAttribute** », qui peuvent être de types différents, verra ci-dessous.

Les attributs dimensionnels sont modélisés comme des attributs appartenant aux niveaux d'agrégation .

Nous distinguons quatre sous-types:

- des attributs d'identification « **IDAttribute** », utilisés à des fins de regroupement lors de l'exécution d'opérations **Roll-up**.
- Des attributs descriptifs Hypercube « **DescriptiveAttribute** », utilisé dans les rôles de prédicat des opérations **Slice and Dice**.
- les attributs « **LevelGeometry** », utilisés pour représenter des formes géométriques de dimension spatiale .
- Les attributs « **LevelTemporality** », utilisés pour modéliser les étendues temporelles des membres temporels.

Les niveaux d'agrégation en tant que classes de faits sont représentés par des extensions de la métaclasse de classe **UML** (voir Figure 3.1).

Pour former des hiérarchies, les niveaux d'agrégation doivent être liés les uns aux autres à l'aide de relations d'agrégation « **Relationship** » définissant des relations de confinement complètes ou partielles entre leurs membres.

Par exemple, la hiérarchie "**ville**" comprend trois niveaux d'agrégation liés. ("**Ville**", "**region**" et "**Pays**")

Chacun des niveaux d'agrégation est décrit par un certain nombre d'attributs. Par exemple, "**ville**" contient deux attributs "**id_ville**" et "**nom_ville**" (voir Figure 3.2).

Comme présenté dans la figure 3.1, nous spécialisons les dimensions, les hiérarchies et les niveaux d'agrégation en trois catégories (thématique, spatiale et temporelle).

- La dimension spatiale « **SpatialDimension** » représente l'emplacement des faits sous la forme d'un axe d'analyse et comprend au moins une hiérarchie spatiale.
- La dimension temporelle « **TemporalDimension** » enregistre des instants ou des périodes au cours desquels des faits se sont produits; il ne contient que des hiérarchies temporelles. À mettre en contraste avec les spatiales et temporelles.
- la dimension thématique « **ThematicDimension** » ne stocke que les membres qui ne sont ni visualisés sur les cartes ni définissent des cadres temporels de faits.

Pour éviter une utilisation arbitraire de notre profil, nous formalisons chaque stéréotype et valeur étiquetée avec un ensemble de contraintes **OCL**. Au niveau du modèle (c'est-à-dire lorsque le profil est utilisé par les concepteurs), ces contraintes sont vérifiées par papyrus au niveau de l'abstraction conceptuelle ; cela empêche modélisation incorrecte.

Par exemple, pour garantir que chaque hypercube spatial contient au moins une dimension spatiale ou au moins une mesure spatiale, nous définissons la contrainte **OCL** suivante dans le contexte du stéréotype Hypercube.

3.3.4. Le diagramme de classes

En se basant sur ce profile nous avons créé le diagramme de classe qui représente le modèle du *flocon de neige* de l'entrepôt de données **EDD** (voir le figure 3.2).

Quand l'échelle du système multidimensionnel est grande, le modèle peut être très complexe à cause des différentes interconnexions entre ses éléments. Dans notre approche, on va regrouper les classes en unités de *niveau supérieur* en créant différents niveaux d'abstraction puis on va simplifier notre *modèle* final.

Notre diagramme de classe est composé d'une table de fait (**appel**) et quatre hiérarchies de dimension (*temps, lieu, client, operateur*)

Le schéma suivant représente le diagramme de classes de l'entrepôt de donnée de notre solution décisionnelle. Ce diagramme est créé sous **papyrus** (Voir figure 3.2)

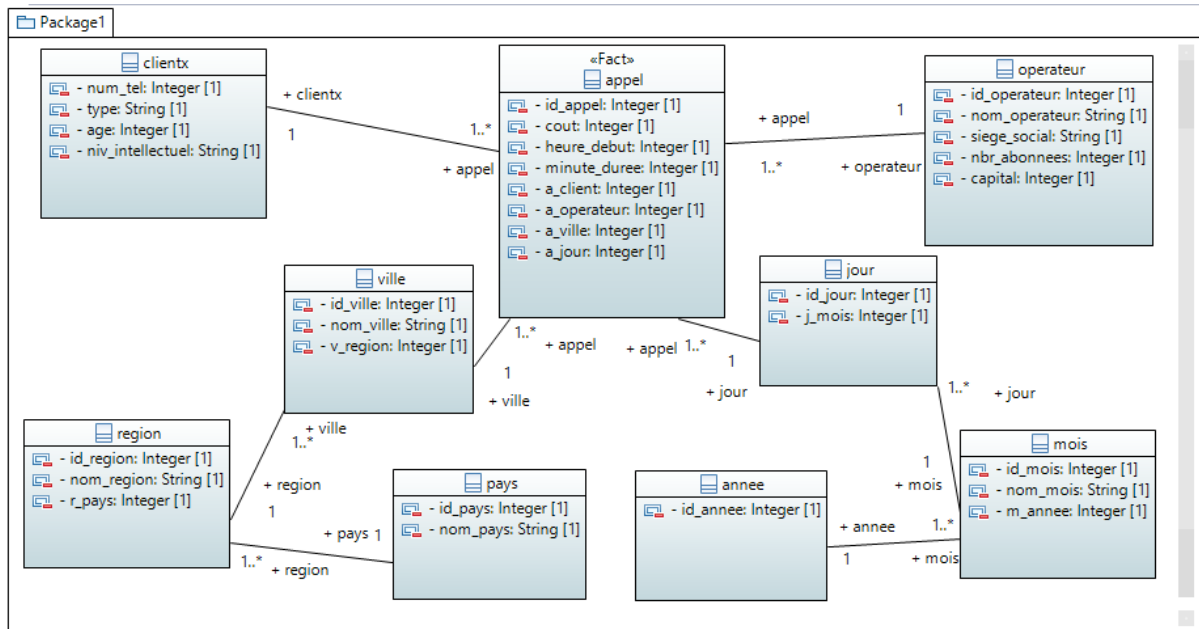


Figure 3.2 : Le diagramme de classe.

Dans ce diagramme, nous avons neuf classes : chaque classe a ses attributs, et entre chaque deux classes nous avons une relation de type association.

Nous avons 2 types de classes :

- **Tables de fait** : la table distribution contient les mesures :cout qui signifie le coût des appels, minute_duree qui signifie laduree des appels par minutes ,heure_debut qui signifie l'heure de début d'appel.
- **Niveau d'agrégation** : pour les dimensions suivants.
 - **Dimension temps** : c'est une dimension temporelle définie par 3 niveaux d'agrégation (jour, mois, année) définie par 3 classes « aggregation_level » stereotype .
Cette dernière dimension contient une hiérarchie:
 - ✓ la hiérarchie : jour -> mois -> annee
 - **Dimension ville** : c'est une dimension spatiale définie par les classes ville, region et pays.
Cette dernière dimension contient une hiérarchie:
 - ✓ la hiérarchie : ville -> region -> pays
 - **Dimension client** : c'est une dimension thématique définie par la classe client.
 - **Dimension operateur** : c'est une dimension thématique définie par la classe operateur.

3.3.5. Les contraintes d'intégrité

Dans nos travaux, nous utilisons essentiellement les contraintes OCL de type invariant pour exprimer des contraintes au niveau modèle sur le modèle multidimensionnel.

On va citer quelques exemples de contraintes d'intégrité

- **Exemple 1** : un hypercube doit avoir au moins deux dimensions.

context Hypercube inv:

```
self.ownedMember->select (m| moclIsKindOf(Dimension))->size()>=2
```

- **Exemple 2** : un hypercube doit avoir exactement une classe de fait.

context Hypercube inv:

```
self.ownedMember->select (m| moclIsTypeOf(Fact))->size()=1
```

- **Exemple 3** : une dimension doit avoir au moins une hiérarchie de dimension.

context Dimension inv:

```
self.ownedMember->select (m| moclIsKindOf(Herarchy))->size()>=1
```

- **Exemple 4** : Une hiérarchie de dimension doit avoir au moins un niveau d'agrégation.

context Herarchy inv:

```
self.ownedMember->select (m| moclIsKindOf(AggLevel))->size()>=1
```

- **Exemple 5** : Une classe de fait doit appartenir à un hypercube.

context Fact inv:

```
self.owner.ooclIsTypeOf(Hypercube)
```

3.4. Conclusion

Dans ce chapitre, nous avons cité les concepts de base de l'UML. Nous avons par la suite opté pour dresser le profil UML pour l'entrepôt de données. Ensuite nous avons tracé le diagramme de classes en se basant sur ce profil. Le diagramme de classes représente le modèle multidimensionnel en flocon de neige, pour le cas des appels téléphoniques.

Dans le chapitre suivant nous allons présenter les différentes étapes pour l'implémentation de l'Entrepôt de données et de l'application OLAP.

4. Chapitre 4 : Implémentation

4.1.Introduction

Dans ce chapitre nous voulons présenter les différentes étapes et les différentes technologies à utiliser pour implémenter l'entrepôt de données et pour réaliser notre application java pour OLAP d'aide à la décision pour les appels téléphoniques.

4.2.Les étapes de réalisation

- **Les outils de travail**

Nous avons développé notre application sur une machine avec les caractéristiques suivantes :

- Processeur **Intel Core (™) i3** avec une vitesse de **2,40 GHz**,
- **8 Go** de mémoire vive,
- Système d'exploitation **Microsoft Windows 10 (64bits)**.

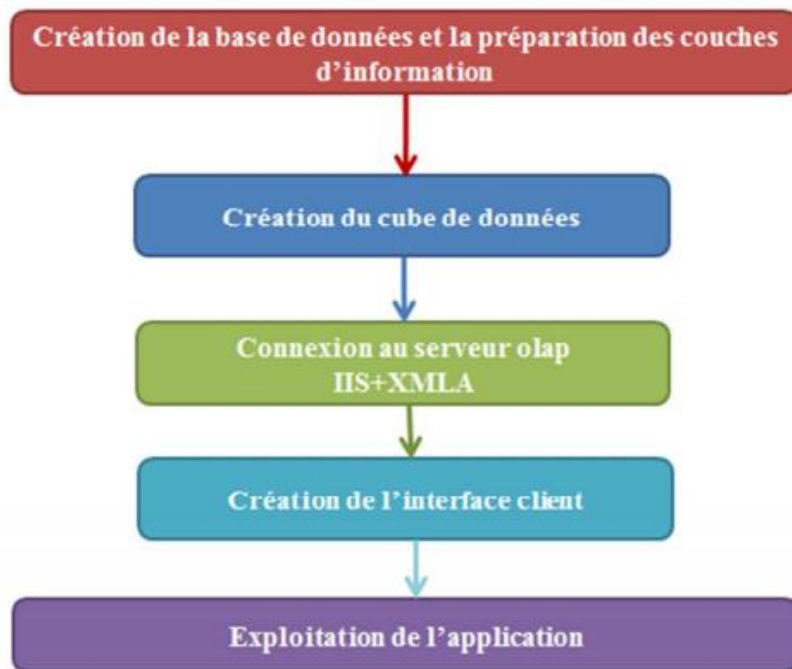


Figure 4.1 : Organigramme de travail

4.3.Création de la base de données

Nous avons créé notre *BDD* par l'utilisation le *SGBD* très connu de *Microsoft* qui s'appelle *SQL Server*.

4.3.1.Quelques définitions

- **Base de Données (BD ou BDD)** : est un ensemble structuré de données homogènes, stocké sur un support informatique, et permettant un accès simplifié et sécurisé à ces données via un logiciel (*SGBD*).
- **Système de Gestion de Bases de Données (SGBD)** : est un logiciel de haut niveau qui permet de manipuler les informations stockées dans une base de données à travers le langage de programmation *SQL*. Ex :*SQL server, MySQL, Oracle...etc.*
- **Structured Query Language (SQL)** : est un langage de programmation à usage spécial conçu pour traiter les données dans un système de gestion de base de données relationnelle. Un serveur de base de données est un programme informatique qui fournit des services de base de données à d'autres programmes ou d'ordinateurs, tels que définis par le modèle client-serveur. Par conséquent, un *serveur SQL* est un serveur de base de données qui implémente la *SQL*.
- **SQL Server 2014** : est un produit logiciel qui stocke principalement et récupère les données demandées par d'autres applications. Ces applications peuvent fonctionner sur le même ou un autre ordinateur. Il existe de nombreuses versions différentes de *Microsoft SQL Server*.
- **SQL Server Management Studio (SSMS)** : est un logiciel utilitaire qui a été lancé avec Microsoft SQL Server 2005, et est conçu pour aider les utilisateurs à configurer, gérer et administrer tous les composants de Microsoft SQL Server. Explorateur d'objets est l'une des principales caractéristiques de cette application et permettre à un utilisateur de naviguer, sélectionner et agir sur l'un des objets au sein du serveur.
- **SQL Server Configuration Manager (SSMC)** : est un outil pour gérer les services associés à *SQL Server*, pour configurer les protocoles réseau utilisés par *SQL Server* et pour gérer la configuration de la connectivité réseau à partir d'ordinateurs *clients SQL Server*.

4.3.2.Les étapes de création la Base De Données

Les étapes suivantes montrent comment créer une base de données et leurs tables dans *SQL Server 2014* à l'aide de *SSMS*, après le téléchargement et l'installation :Ouvrez le logiciel *SSMS* et développez le serveur sur lequel nous voulons créer notre *BDD* et choisissez le mode d'authentification.

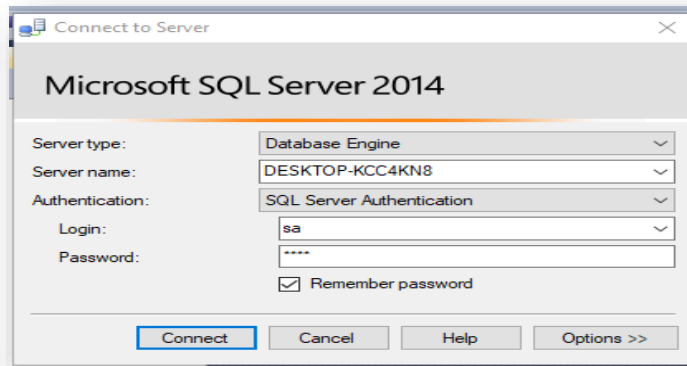


Figure 4.2 : connexion au moteur de bases de données.

- Dans la fenêtre **Object Explorer** qui apparaît dans le gauche de l'écran, cliquez droit sur le nœud **Databases**, puis cliquez sur **New Database**, puis tapez le nom de BDD dans la fenêtre qui va apparaître et cliquez sur **OK**.

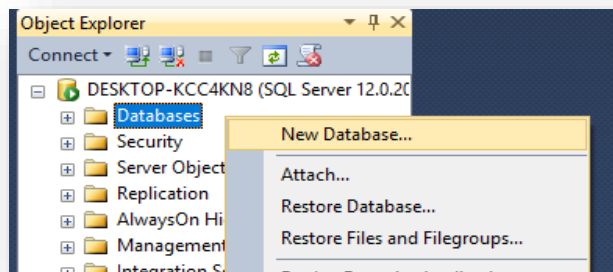


Figure 4.3: Création d'une nouvelle BDD

- Dans la fenêtre **Object Explorer**, cliquez sur **Databases**, puis cliquez sur notre nouvelle BDD (**dbtfn** dans notre cas), puis cliquez droite sur **Tables**, et cliquez sur **New**, et **Table**.

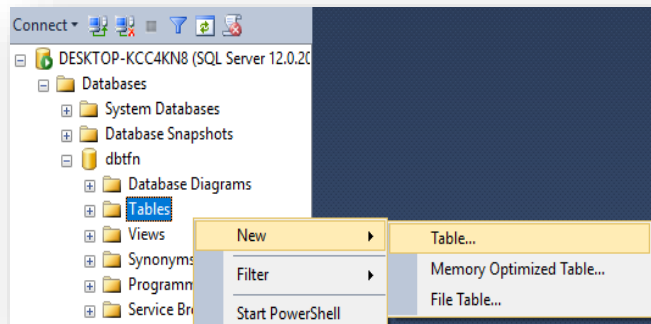


Figure 4.4: Création d'un nouveau tableau

- Dans la fenêtre qui apparaît à la droite, remplissez les colonnes de votre tableau par tapez les **noms de colonnes**, et le **type** leurs données, et cochez le carré dans la colonne **Allow Nulls** si les colonnes accepter **les valeurs nulles**, puis sélectionnez **les clés primaires** et tapez **les noms des tables** sur la fenêtre propriété. Et sauvegardez.

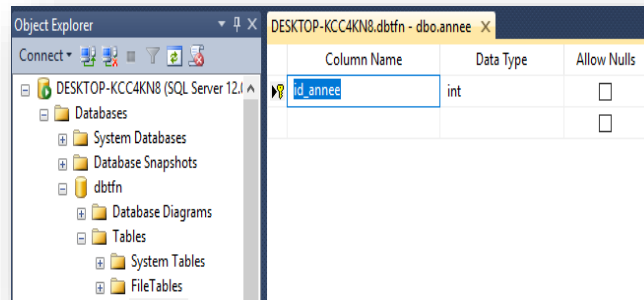


Figure 4.5 : remplissage de données

- Dans l'Explorateur d'objets, faites un clic droit sur la table que vous souhaitez ouvrir et sélectionnez **Edit Top 200 Rows**, pour insérer les données.
- **Note** : car il y a un manque des données dans ce domaine (les appels téléphoniques) puisqu'ils sont privés pour les sociétés des opérateurs et ne sont pas disponible pour tous les gens, nous avons essayé un échantillon de 200 appels virtuels dans différentes dates et différentes zones. Alors les résultats ne sont pas exacts, puisque ne sont pas réels.

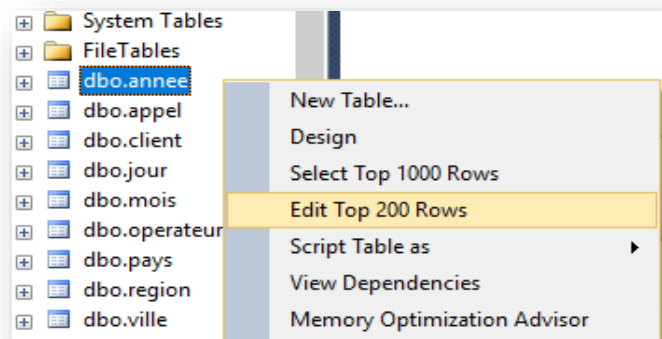


Figure 4.6 : Insertion des données.

- Dans la fenêtre *Object Explorer*, cliquez sur *Databases*, puis cliquez sur notre nouvelle *BDD*, puis cliquez droite sur *Database Diagrams*, et cliquez sur *New Database Diagram*, puis créez les relations entre les tableaux par les clés primaires et les clés étrangères.

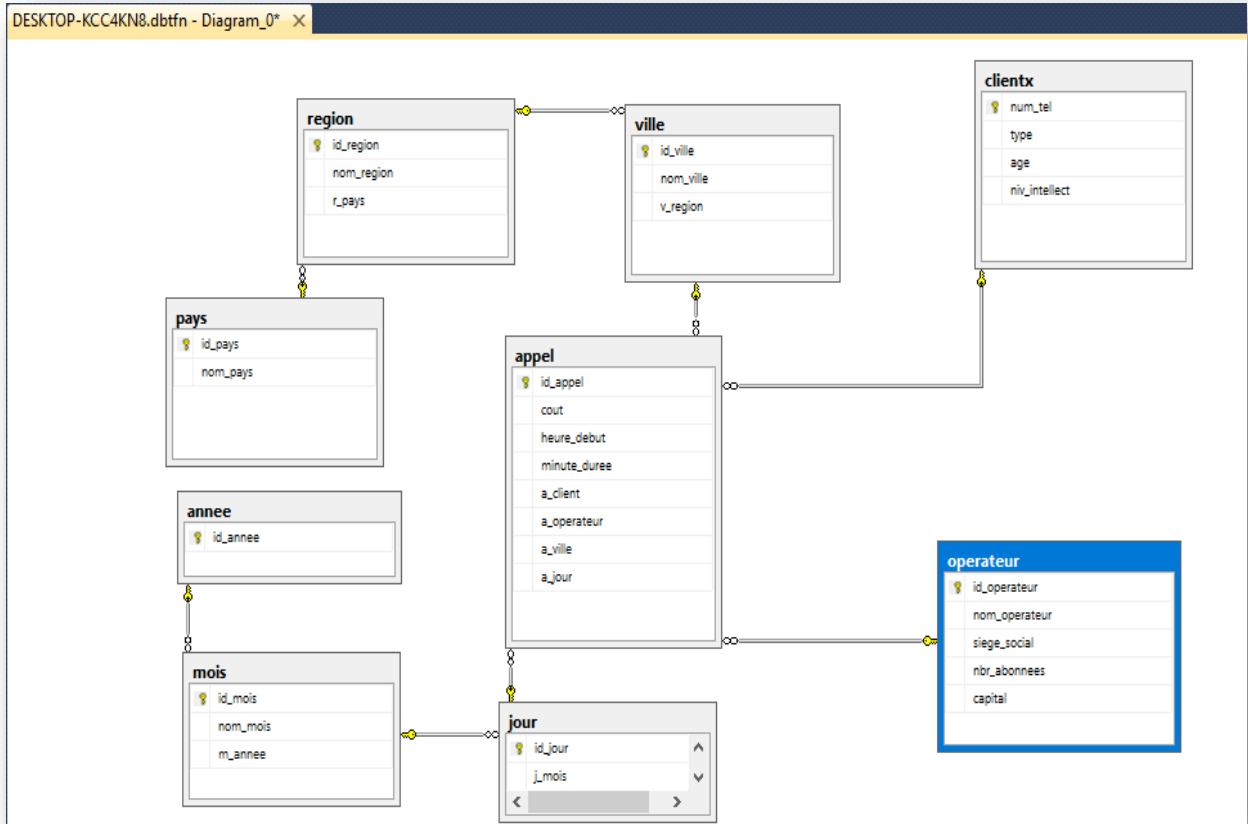


Figure 4.7 : Diagramme flocon de neige .

4.4. Création du cube OLAP

Pour la modélisation de notre *Cube OLAP*, nous avons utilisé le logiciel *Visual studio 2013* qui utilise un autre logiciel pour créer un projet *BIDS* qui s'appelle *SQL Server Data Tools*.

4.4.1. Quelques définitions

- **Visual Studio 2013** : Visual Studio 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. Ces outils permettent de créer notre application.
- **SQL Server Data Tools (SSDT)** : est un ensemble d'outils qui permet aux développeurs de bases de données professionnelles et d'application pour effectuer tous les travaux de conception de leur base de données pour *SQL Server* et *SQL Azure* dans *Visual Studio*.
SSDT offre une riche expérience de développement *SQL Server* avec les services de première langue de classe et l'intégration *VS*, ainsi que déclaratives, des outils basés sur des modèles qui peuvent être utilisés tant pour le développement en ligne et hors ligne.
- **MDX** : Le langage MDX remplace le langage SQL dans les bases de données relationnelles. Bien qu'il soit différent, il y a de fortes ressemblances avec le langage SQL.
- **BIDS** : Business Intelligence Développement Studio est un environnement intégré qui permet de développer des constructions Business Intelligence, telles que des cubes, des sources de données, des rapports et des packages Intégration Services. Il contient des modèles de projets qui fournissent un contexte pour développer des constructions spécifiques.

4.4.2. Les étapes de création le cube OLAP

- Ouvrez *VS_2013*, et dans la barre menu cliquez sur *File*, puis *New*, puis *Project*. Alors une fenêtre *New Project* va apparaître.

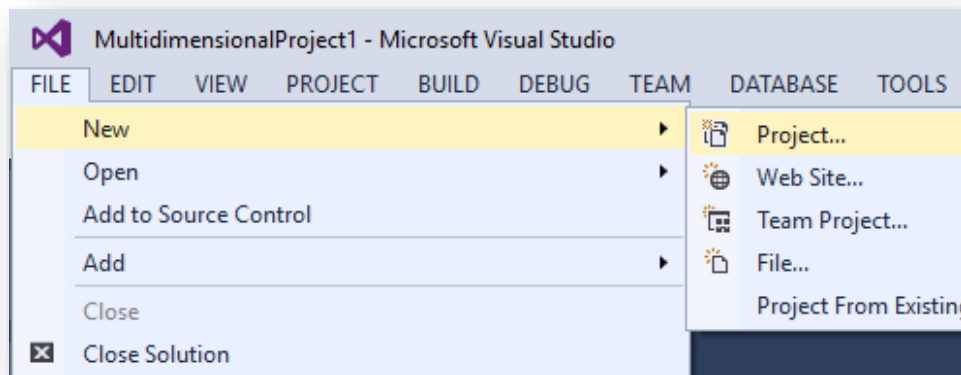


Figure 4.8 : Création d'un nouveau projet .

- Dans la fenêtre gauche cliquez sur **Installed**, puis **Templates**, puis **Business Intelligence** et **Analysis Services Multidimensional and Data Mining**, puis tapez le nom de votre nouveau projet, et cliquez sur **OK**.

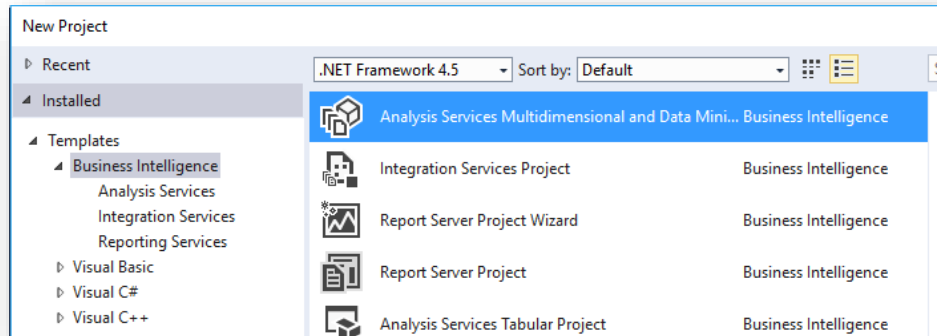


Figure 4.9 : Création d'un projet d'informatique décisionnelle

- Dans la fenêtre **Solutions Explorer** sur la droite d'écran, Ajoutez un nouveau **Data Source**, **Data Source View**, et un nouveau **cube**.

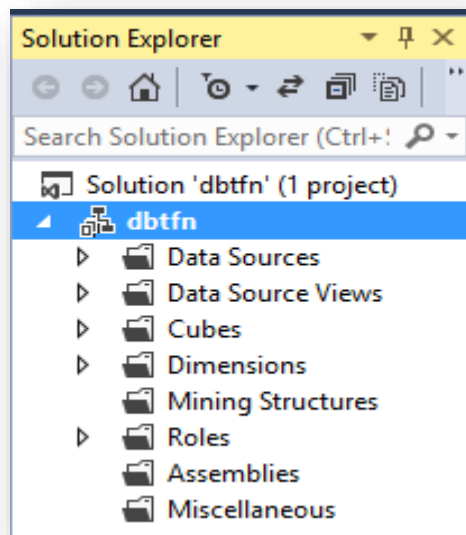


Figure 4.10 : Explorateur des solutions pour le projet multidimensionnelle

- Sélectionnez votre **Mésures**.

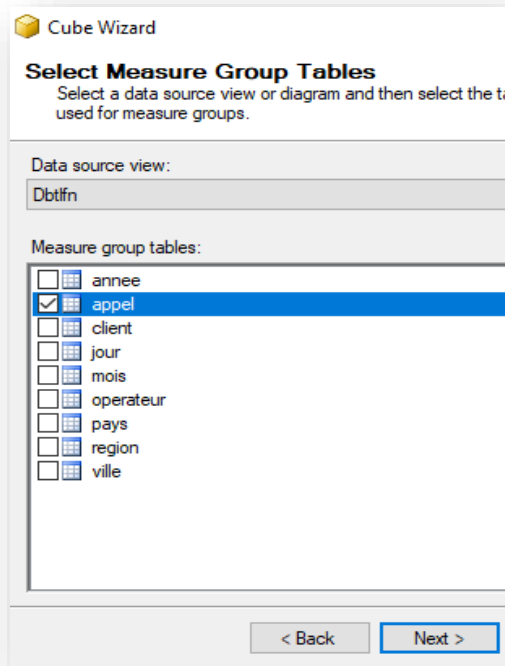


Figure 4.11 : Sélectionner les mesures

- Sélectionnez votre **Dimensions** et **Hiérarchies**.

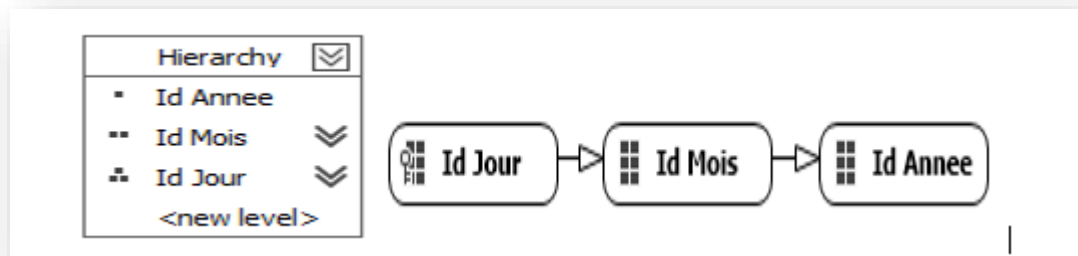


Figure 4.12 : Sélectionner Une hiérarchie

4.4.3. Tester la connexion entre SSMS & Visual Studio

- Dans la fenêtre *Solution Explorer* dans *VS_2013*, cliquez droite sur *votre projet*, puis cliquez sur *process*.

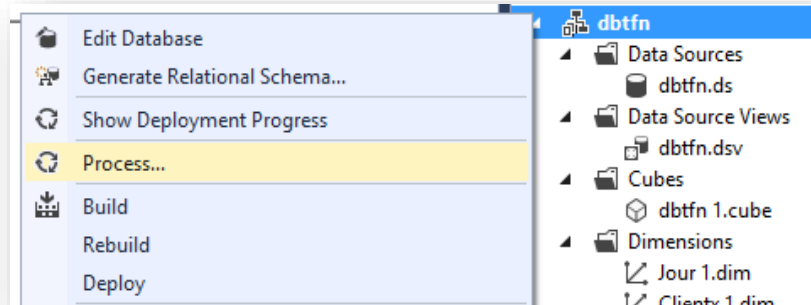


Figure 4.13 : la déploiement de cube de données

- Dans la fenêtre *Object Explorer* dans *SSMS_2014*, cliquez sur *Connect* et sélectionnez *Analysis Services*, et connecter avec votre server par l'utilisation de votre nom de serveur.

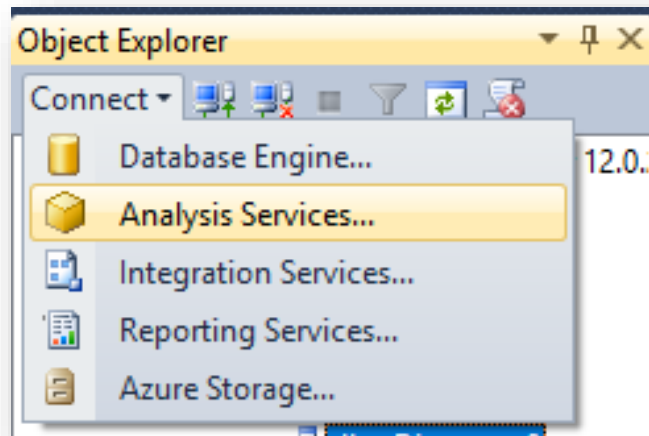


Figure 4.14 : Connexion de serveur SSAS

4.5. Connexion Server-IIS

Pour accéder à une instance Analysis Services vous pouvez activer l'accès HTTP en configurant MSMDPUMP.dll, une extension ISAPI qui s'exécute dans Internet Information Services (IIS) et qui pompe des données entre des applications clientes et un serveur Analysis Services SSAS.

Cette approche constitue une alternative à la connexion à Analysis Services lorsqu'une solution **OLAP** nécessite les capacités suivantes :

- Un accès client via une connexion Internet ou extranet.
- Une application cliente qui s'exécute dans un environnement réseau qui permet des connexions **HTTP** mais pas **TCP/IP**.
- Des méthodes d'authentification autres que la sécurité intégrée de **Windows** sont requises. **IIS** prend en charge les connexions anonymes et l'authentification de base.
- Des applications clientes qui ne peuvent pas utiliser les bibliothèques clientes Analysis Services (par exemple, une application **Java** dans notre cas).

Si on ne peut pas utiliser les bibliothèques clientes Analysis Services pour accéder aux données, on peut utiliser **SOAP** et **XML/A** sur une connexion **HTTP** directe à une instance Analysis Services.

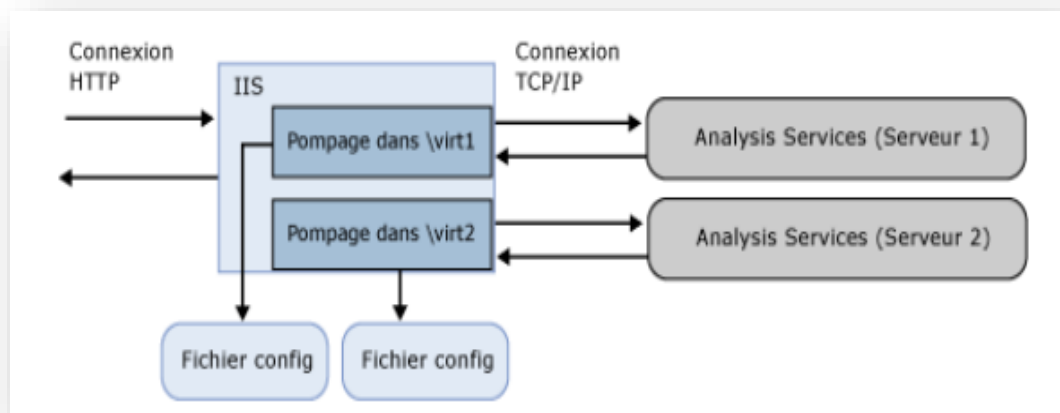


Figure 4.15 : Architecture de IIS

4.5.1. Quelques définitions

- **IIS10.0Express** : est une version allégée, autonome d'*IIS* optimisés pour les développeurs. *IIS express*, il est facile d'utiliser la version la plus récente de *IIS* pour développer et sites de test. Il a toutes les capacités de base de *IIS 7* et ci-dessus, ainsi que des fonctionnalités supplémentaires conçues pour faciliter le développement de site Web, notamment :
 - Il ne fonctionne pas comme un service ou d'exiger des droits d'utilisateur administrateur pour effectuer la plupart des tâches.
 - IIS Express fonctionne bien avec les applications *ASP.NET* et *PHP*.
 - Plusieurs utilisateurs d'*IIS Express* peuvent fonctionner indépendamment sur le même ordinateur.
- **HyperText Transfert Protocole (HTTP)** : est un protocole utilisé par le World Wide Web et ce protocole définit la façon dont les messages sont formatés et transmis, et quelles actions *les serveurs Web* et *les navigateurs* doivent prendre en réponse à diverses commandes.

Par exemple, lorsque vous entrez une *URL* dans votre navigateur, cela envoie en fait une commande *HTTP* au serveur Web dirigeant chercher et transmettre la demande page Web. L'autre principale norme qui contrôle le fonctionnement du *World Wide Web* est *HTML*, qui couvre la façon dont les pages Web sont formatées et affichées.
- **eXtended Markup Language for Analysis (XMLA)** : est un protocole basé sur *SOAP XML*, conçu spécialement pour l'accès aux données universel à toute source de données multidimensionnelle standard qui peut être accessible via une connexion *HTTP*. Analysis Services utilise *XMLA* comme seul protocole lors de la communication avec les applications clientes. Fondamentalement, toutes les bibliothèques client pris en charge par Analysis Services et réponses forment des demandes en *XMLA*.

En tant que développeur, vous pouvez utiliser *XMLA* pour intégrer une application client Analysis Services, sans aucune dépendance sur le *.NET Framework* ou des interfaces *COM*. Exigences de la demande qui incluent l'hébergement sur un large éventail de plates-formes peuvent être satisfaites en utilisant *XMLA* et une connexion *HTTP* à Analysis Services.
- **Simple Object Access Protocol (SOAP)** : est un protocole de messagerie basé sur XML pour l'échange d'informations entre les ordinateurs. *SOAP* est une application de la spécification XML.
 - SOAP est un protocole de communication destiné à communiquer via Internet.
 - SOAP peut étendre HTTP pour la messagerie XML.
 - SOAP assure le transport de données pour les services Web.

- SOAP peut échanger des documents complets ou appeler une procédure à distance.
- SOAP peut être utilisé pour diffuser un message.
- SOAP est plate-forme et indépendant de la langue.
- SOAP est le moyen XML de définir les informations envoyées et comment.
- SOAP permet aux applications clientes de se connecter facilement à des services à distance et invoquer les méthodes distantes.

Bien que **SOAP** peut être utilisé dans une variété de systèmes de messagerie et peut être livré via une variété de protocoles de transport, la mise au point initiale des appels de procédure **SOAP** est à distance transportés via **HTTP**.

4.5.2. Les étapes de connexion IIS

- Copier les fichiers de dossier **ISAPI** de votre projet **SSAS** dans autre nouveau dossier nommé **OLAP** dans « *C://inetpub/wwwroot/OLAP* ».

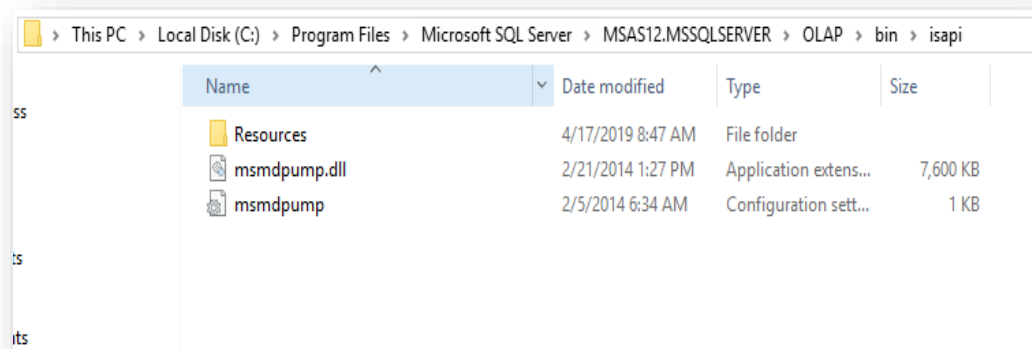


Figure 4.16 : Les fichiers ISAPI de notre cube

- Compléter la configuration dans **IIS 10.0 Express** ou directement sur le fichier xml de votre dossier **OLAP**.

4.5.3. Tester la connexion entre SQL Server 2014 & IIS

- Dans la fenêtre *Object Explorer* dans *SSMS_2014*, cliquez sur *Connect* et sélectionnez *Analysis Services*, et connectez-vous avec votre serveur par l'utilisation de la chaîne < <http://localhost/olap/msmdpump.dll> >.

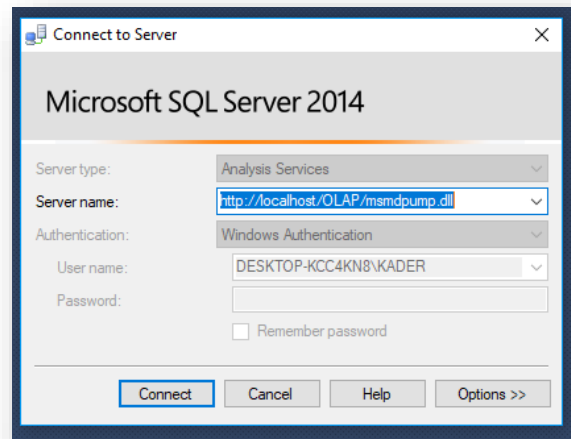


Figure 4.17 : Connexion de SSAS à travers l'URL

4.6. Création Application JAVA

Cette étape consiste à créer un projet sur l'environnement *java* en utilisant *Eclipse ide 03-2019*. Pour des raisons d'utilisation des outils spécifiques à l'affichage des outils de « Business Intelligence », on aura besoin d'ajouter quelques *JRE*.

4.6.1. Quelques définitions

- **Java** : est un langage de programmation qui produit des logiciels pour plusieurs plates-formes. Quand un programmeur écrit une application Java, le code compilé (connu sous le nom bytecode) fonctionne sur la plupart des systèmes d'exploitation (**OS**), y compris **Windows**, **Linux** et **MacOS**. Java tire une grande partie de sa syntaxe des langages de programmation **C & C++**. Java a été développé au milieu des années **1990** par **James A. Gosling**, un ancien informaticien avec **Sun Microsystems**.
- **JDK** : Le langage Java, dans sa version **JDK 8u101** qu'il est un langage bien connu et largement répandu. Il offre de nombreuses bibliothèques qui facilitent le développement d'applications, notamment des parsers pour **XML**. En plus les applications Java s'exécutent en utilisant une machine virtuelle, ce qui les rend indépendantes du système d'exploitation. Des machines virtuelles Java ont été développées pour la plupart des systèmes actuels, ce qui facilite la portabilité des applications Java. Nous avons opté pour ce langage.

- **Java Runtime Environment (JRE)** : est un ensemble d'outils logiciels pour le développement d'applications Java. Il combine la **machine virtuelle Java (JVM)**, les classes de base de la plate – forme et les bibliothèques de soutien. **JRE** fait partie du **Java Development Kit (JDK)**, mais peut être téléchargé séparément. JRE a été initialement développé par **Sun Microsystems Inc.**, une filiale en propriété exclusive d'**Oracle Corporation**. Aussi connu sous le nom d' **exécution Java**.
- **Eclipse** : est un **IDE** (environnement de développement intégré) écrit en **Java**, extensible par des greffons, multi-langages et multi-plates-formes, qui s'intègre particulièrement bien à **GNOME**. Toutes les fonctions qu'on peut attendre de ce genre de logiciel sont présentes ou existent sous forme de greffons (coloration syntaxique, complétion, debuggé, gestion de projets, intégration aux gestionnaires de versions, ...)
- **Olap4j** : Est un API qui est l'équivalent OLAP de **JDBC** pour les données relationnelles. Plus précisément, **olap4j** étend classes de base de **JDBC** spécifications afin d'apporter des sources de données **OLAP** pour la plate-forme **Java**. Les connexions peuvent être obtenues par le gestionnaire de connexion **JDBC**.

Il utilise des instructions, qui sont fournis par les connexions. Les requêtes, formulées en utilisant le langage **MDX**, peuvent être envoyés textuellement à la connexion.

- **Application programming interface (API)** : Est une collection de paquets pré écrits, des classes et des interfaces avec leurs méthodes respectives et les constructeurs. Semblable à une interface utilisateur, ce qui facilite l'interaction entre les humains et les ordinateurs, une **API** sert une interaction facilitante l'interface de logiciel.

En Java, la plupart des tâches de programmation de base sont effectuées par les classes de l'**API** et les paquets, qui sont utiles pour réduire le nombre de lignes écrites dans des morceaux de code.

- **Jfreechart** : est une bibliothèque Java libre de 100% graphique qui le rend facile pour les développeurs d'afficher des graphiques de qualité professionnelle dans leurs applications. Jeu de fonctionnalités étendu de JFreeChart comprend :
 - Une **API** cohérente et bien documenté.
 - Une conception flexible qui est facile à étendre, et les objectifs à la fois côté serveur et les applications côté client.
 - Support pour de nombreux types de sortie, y compris les composants **Swing** et le **Javax**, les fichiers d'images (y compris **PNG** et **JPEG**) et vectoriels formats de fichiers graphiques (y compris **PDF**, **EPS** et **SVG**).
 - **JFreeChart** est open source ou plus spécifiquement, des logiciels libres.
- **Java DataBase Connectivity (JDBC)** : est une **API** pour la connexion des programmes écrits en Java aux données populaire **BDD**. L'interface de programme d'application vous permet d'encoder des déclarations de demande d'accès au **SQL** qui sont ensuite transmises

au programme qui gère la base de données. Il renvoie les résultats à travers une interface similaire. *JDBC* est très similaire à la connectivité de base de données et un petit programme *java*.

- **Mondrian** : est un moteur *OLAP* écrit en Java. Il exécute des requêtes écrites en langage *MDX*, en lisant des données à partir d'une base de données relationnelle, et présente les résultats dans un format multidimensionnel via une *API* Java.

4.6.1. Les étapes de création l'application java

- Dans le barre de menu sur le haut, Cliquez sur **File, new** puis **JavaProject** et **Class** pour crée un nouveau project java et une nouvelle class java, avant nommer les deux.

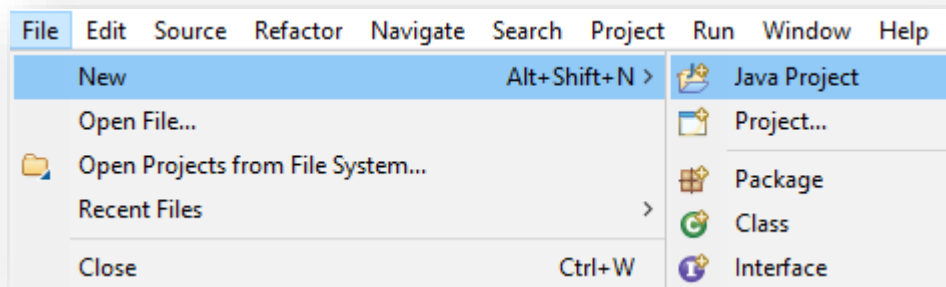


Figure 4.18 : Créer un nouveau projet Java

- Dans le barre de menu sur le haut, Cliquez sur **File, new** puis **Other, Windows Builder, Swing designer** et **Application java** pour crée un nouveau application java.

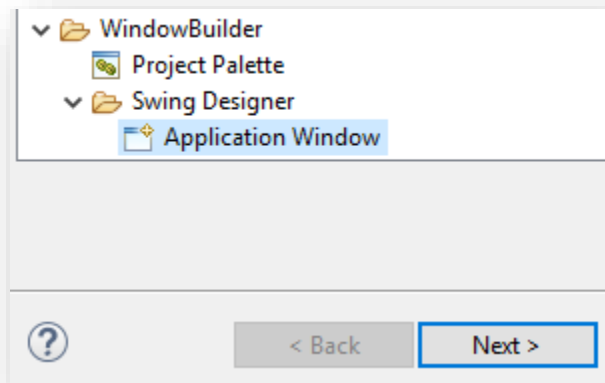


Figure 4.19 : Créer un nouveau application java

- Dans le *Package Explorer* sur le gauche, Cliquez sur **Buid path, Configure Build path, Libraries** puis **Add External JAR** pour ajouter les bibliothèques *jfreechart, JDBC*.

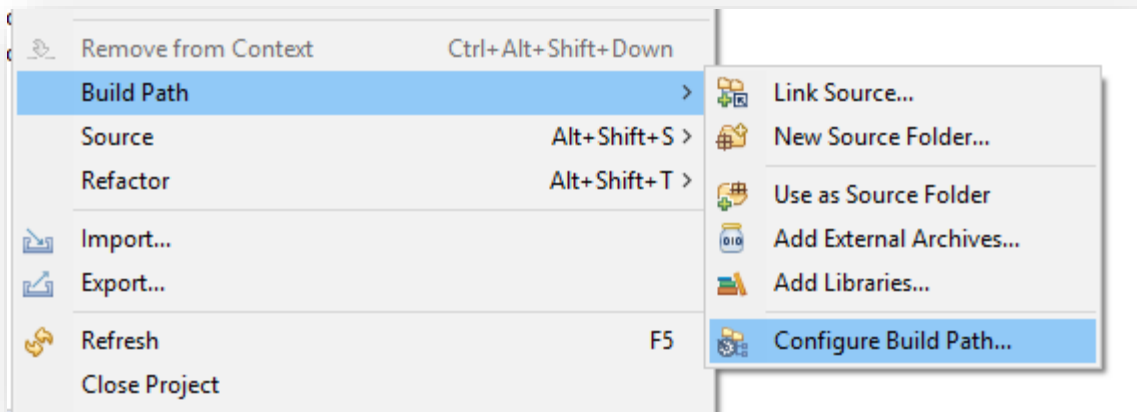


Figure 4.20 : configure le build path

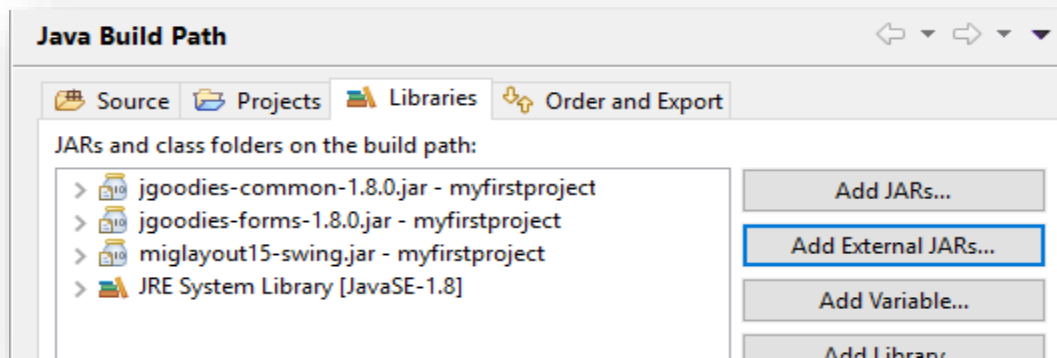


Figure 4.21 : ajouter des bibliothèques JAR.

4.7. Exploitation du cube de données

L'entrepôt de données peut être exploité à travers plusieurs manières , on mentionner quelques manières dans cette mémoire.

4.7.1. SQL Server management studio

L'entrepôt de données peut être exploité à travers l'interface *SQL server Management Studio SSMS*, à travers les requêtes *MDX* sur *SQLServer*.

- Exemple de requête

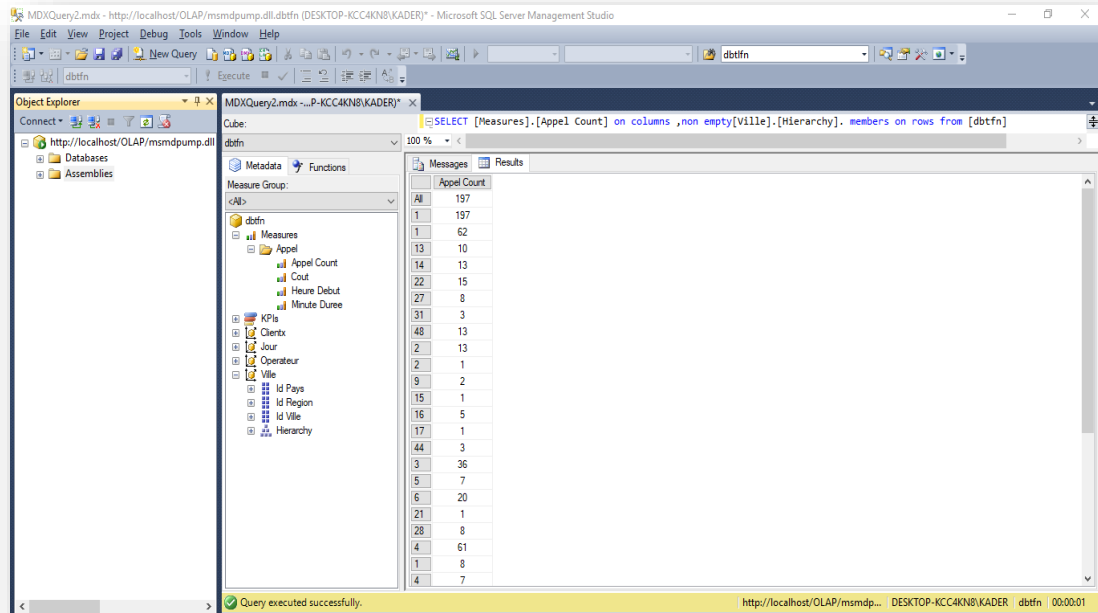


Figure 4.22 : La requête MDX pour le nombre des appels non null selon les villes.

4.7.2. Power BI Desktop

L'exploitation du cube de données peut se faire aussi grâce à d'autres clients OLAP tels que **Power BI**, qui nous avons utilisé dans notre projet.

- Définition de Power BI Desktop

Est une solution d'analyse d'affaires qui vous permet de visualiser vos données et partager des idées au sein de votre organisation, ou les intégrer dans votre application ou site web. Connectez-vous à des centaines de sources de données et mettez vos données à la vie avec des tableaux de bord en direct et des rapports.

- Exemples

Nous allons citer quelques exemples pour l'exploitation de cube de données par l'outil **Power BI**.

- Exemple 1

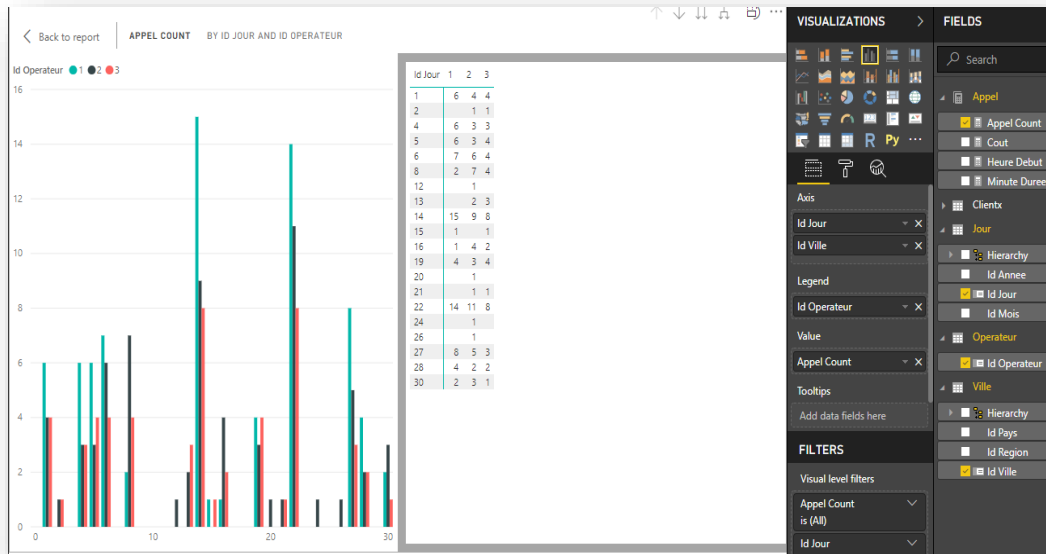


Figure 4.23 : Exploitation par Power BI pour le nombre des appels selon les villes et les jours de mois et l'opérateur mobile .

- Exemple 2

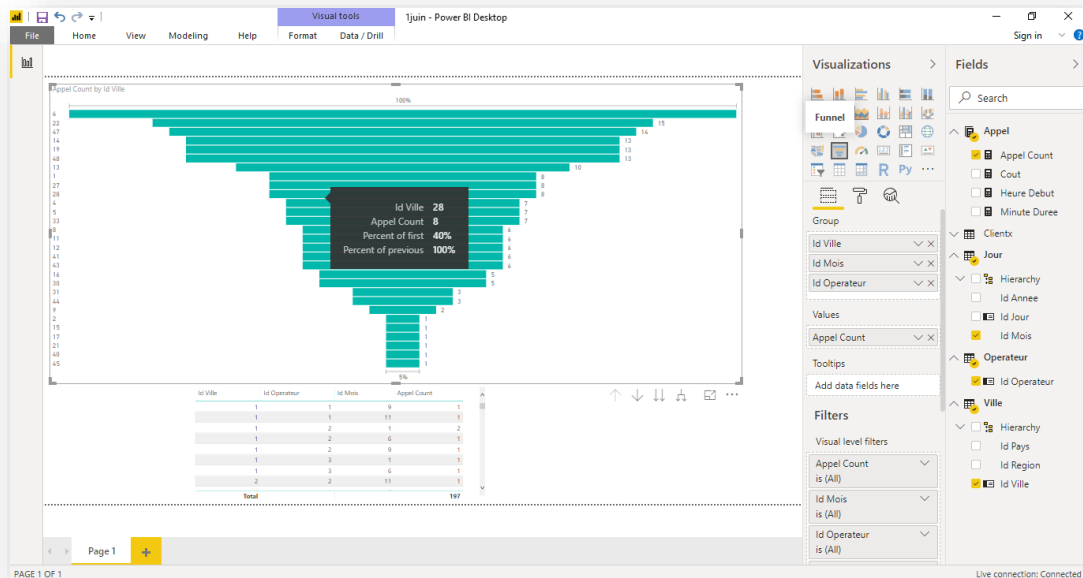


Figure 4.24 : diagramme de funnel pour l'exploitation par power BI

4.7.3. Application client OLAP

L'entrepôt de données peut être exploité aussi à travers l'interface *Eclipse ide*, par l'utilisation des requêtes *MDX* ou par l'importation de notre projet sur le logiciel.

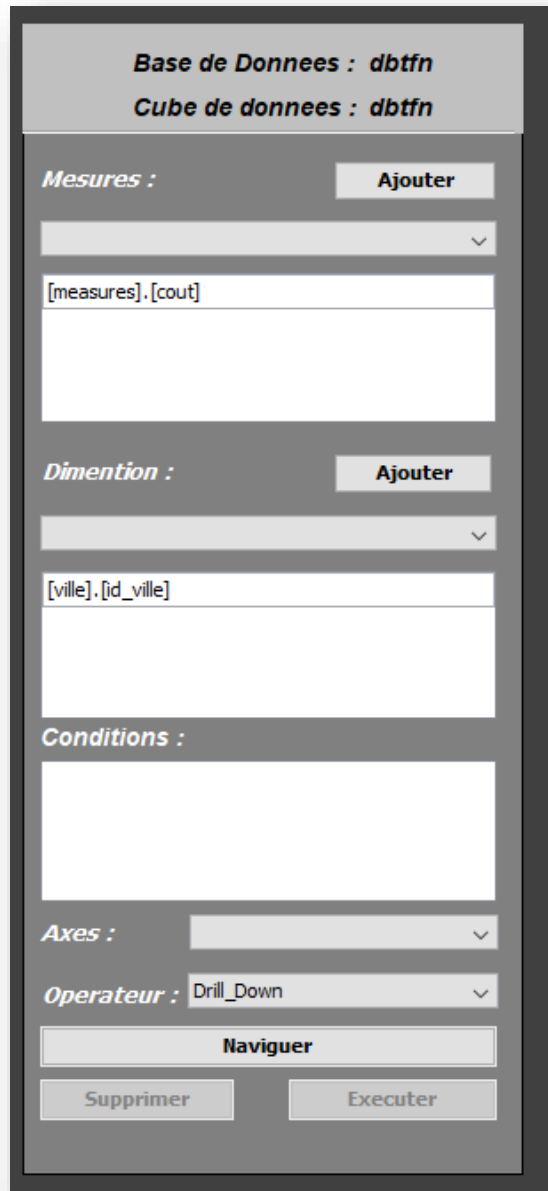


Figure 4.25 : Exploitation par Eclipse IDE

4.8.Conclusion

Dans ce chapitre, nous avons commencé par citer quelques définitions pour les concepts de base pour l'implémentation . Nous avons par la suite présenté les outils utilisés pour l'implémentation de notre application en exposant les différents composants et fonctionnalités permettant le développement d'une application décisionnelle. Ensuite nous avons cité les manières d'exploitation de notre travail.

Conclusion générale

L'informatique décisionnelle apporte des solutions nouvelles pour la modélisation, l'interrogation et la visualisation de données dans un objectif d'aide à la décision. Les modèles multidimensionnels ou modèles d'hyper-cube sont des modèles qui permettent de structurer les données pour l'analyse décisionnelle en explicitant la notion de dimension.

Nous avons vu à travers les différents chapitres entamés précédemment les différents outils et concepts pour réaliser notre travail.

Dans le premier chapitre, nous avons introduit les concepts de base concernant les systèmes de l'informatique décisionnels, l'Online analysis processing, et les entrepôts de données.

Dans le deuxième chapitre, nous avons parlé aussi sur les communications téléphoniques, puis nous avons présenté quelques statistiques sur le marché algérien de télécommunication.

Dans le troisième chapitre, nous avons présenté la modélisation UML du cas des communications téléphoniques par la présentation de profile UML de l'entrepôt de données et le diagramme de classes du modèle flocon de neige de l'application OLAP .

Dans le quatrième chapitre, nous avons présenté les différentes étapes et les différentes technologies à utiliser pour implémenter l'entrepôt de données et pour réaliser notre application java pour OLAP d'aide à la décision pour les appels téléphoniques.

A travers ce projet fin d'étude, nous avons pu effectuer les tâches suivantes :

1. Etude approfondue sur le marché de télécommunications.
2. Etude des caractéristiques de différents acteurs qui influencent sur le marché.
3. Etude des axes d'analyse (dimensions).
4. Etude des indicateurs de gestion (mesures de faits).
5. Réalisation de l'entrepôt de données pour les appels téléphoniques.
6. Développement de l'application OLAP pour l'analyse.
7. Interprétation des résultats.

Bibliographie

- [1] Informatique décisionnelle définition [en ligne] cité dans : futurascience.
<https://www.futura-sciences.com/tech/definitions>, consulté le 20 Février 2019
- [2] Boulil, K. (2012). *"Une approche automatisée basée sur des contraintes d'intégrité définies en UML et OCL pour la vérification de la cohérence logique dans les systèmes SOLAP : applications dans le domaine agri-environnemental"*, Thèse Doctorat : Informatique. France, Université Blaise Pascal – Clermont-Ferrand II.
- [3] Inmon, W. (2000). *"Building the Data Warehouse: The Third Edition"*. John Wiley & Sons.
- [4] Revuz, D. (2000). « *Entrepôt de données* » : cours [en ligne], <http://igm.univ-mlv.fr/~dr/XPOSE2010/chainedecisionnelle>. Consulté le 22 Février 2019
- [5] Kimball, R. (2002). *"The Data Warehouse Toolkit"*, Editions John Wiley & Sons.
- [6] Codd, E.F. (1993) *"Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate"*. Codd & Associates.
- [7] <https://www.arpce.dz>, consulté le 22 Février 2019
- [8] <https://www.djezzy.com>, consulté le 22 Février 2019
- [9] <https://www.ooredoo.com>, consulté le 22 Février 2019
- [10] <https://www.algeriatelecom.com>, consulté le 22 Février 2019.
- [11] Treiche, R. I. & Souci F. Z. (2016). *"Vérification des contraintes d'intégrité dans les entrepôts de données spatiales et SOLAP"*, Mémoire Master : Informatique. Mostaganem-Algérie, Université Abdelhamid Ibn Badis .
- [12] Benadda, A. & Boukredia Y. (2016). *"Modélisation d'un entrepôt de données spatiales à l'aide des profils UML. Application : Planification des travaux sylvicoles pour la conservation des forêts de Mostaganem"*, Mémoire Master : Informatique. Mostaganem-Algérie, Université Abdelhamid Ibn Badis .
- [13] [https:// www.microsoft.com](https://www.microsoft.com).
- [14] <https://www.iis.net>.

Sommaire

Introduction Générale	1
1. Chapitre1 : Concepts de base en informatique décisionnelle	2
1.1. Introduction.....	2
1.2. Informatique décisionnelle	2
1.2.1. Définition	2
1.2.2. Le Système Opérationnel Vs Le Système Décisionnel	2
1.2.3. Architecture du Système d'Information Décisionnelle (SID).....	3
1.3. Entrepôt de données (Data Warehouse)	5
1.3.1. Définition	5
1.3.2. Les avantages d'un Entrepôt de données.....	5
1.3.3. Les caractéristiques d'Entrepôt de données	6
1.3.4. Structure d'un data Warehouse	6
1.3.5. Les 2 approches pour construire un Data Warehouse.....	6
1.4. OnLine Analysis Processing (OLAP)	7
1.4.1. Définition	7
1.4.2. Les 12 règles OLAP.....	8
1.4.3. Cube OLAP	8
1.4.4. Les différentes variantes de l'OLAP	9
1.5. Modélisation multidimensionnelle.....	10
1.5.1. Définition	10
1.5.2. Les types de schémas.....	10
1.6. Les solutions disponibles	12
1.6.1. Open Source	12
1.6.2. Propriétaire	12
1.7. Conclusion	13
2. Chapitre 2 : La communication téléphonique en Algérie	14
2.1. Introduction.....	14
2.2. La communication téléphonique.....	14
2.2.1. Définition	14
2.2.2. Les moyens de communication téléphonique	14

2.3.	Les opérateurs téléphoniques en Algérie	14
2.3.1.	Définition d'opérateur	15
2.3.2.	Les Dénominations commerciaux des opérateurs	15
2.3.2.1.	Djezzy	15
2.3.2.2.	Ooredoo.....	15
2.3.2.3.	Algérie télécom	15
2.4.	Indicatif téléphonique en Algérie.....	16
2.5.	Marché de téléphonie en Algérie	17
2.5.1.	Un aperçu sur le marché	17
2.5.2.	L'Autorité de Régulation de la Poste et des Télécommunications (ARPT)	17
2.5.3.	Statistiques & graphes d'ARPT	17
2.6.	Conclusion	20
3.	Chapitre3 : Modélisation.....	21
3.1.	Introduction.....	21
3.2.	Quelques définitions.....	21
3.3.	Les approches de modélisation multidimensionnelle basés sur l'UML.....	22
3.3.1.	Extension de l'UML	22
3.3.3.	Le profile UML.....	23
3.3.4.	Le diagramme de classes	26
3.4.	Conclusion	28
4.	Chapitre 4 : Implémentation.....	29
4.1.	Introduction.....	29
4.2.	Les étapes de réalisation	29
4.3.	Création de la base de données.....	30
4.3.1.	Quelques définitions.....	30
4.3.2.	Les étapes de création la Base De Données.....	30
4.4.	Création du cube OLAP	34
4.4.1.	Quelques définitions.....	34
4.4.2.	Les étapes de création le cube OLAP	34
4.4.3.	Tester la connexion entre SSMS & Visual Studio.....	37
4.5.	Connexion Server-IIS	38
4.5.1.	Quelques définitions.....	39
4.5.2.	Les étapes de connexion IIS	40

4.5.3.	Tester la connexion entre SQL Server 2014 & IIS.....	41
4.6.	Création Application JAVA.....	41
4.6.1.	Quelques définitions.....	41
4.6.1.	Les étapes de création l'application java	43
4.7.	Exploitation du cube de données	44
4.7.1.	SQL Server management studio.....	44
4.7.2.	Power BI Desktop.....	45
4.7.3.	Application client OLAP	47
4.8.	Conclusion	48
	Conclusion générale.....	49
	Bibliographie	50

Liste de figure

Figure 1.1 : Architecture typique d'un système OLAP	4
Figure 1.2 : la représentation schématique de l'approche Top-Down.....	7
Figure 1.3 : la représentation schématique de l'approche Bottom-up.....	7
Figure 1.4 : table de faits	10
Figure 1.5 : table de dimensions.	10
Figure 1.6 : la représentation schématique d'un modèle en étoile.	11
Figure 1.7 : la représentation schématique d'un modèle de neige en flocon.	11
Figure 1.8 : la représentation schématique d'un modèle en constellation.....	12
Figure 2.1 : comparaison entre le nombre d'abonnés des opérateurs.	19
Figure 2.2 : l'évolution du parc de la téléphonie mobile (2001-2017)	19
Figure 2.3: l'évolution du parc de la téléphonie mobile selon les technologies	20
Figure 3.1 : le profile UML	24
Figure 3.2 : Le diagramme de classe.	27
Figure 4.1 : Organigramme de travail	29
Figure 4.2 : connexion au moteur de bases de données.	31
Figure 4.3: Création d'une nouvelle BDD.....	31
Figure 4.4: Création d'un nouveau tableau.....	31
Figure 4.5 : remplissage de données.....	32
Figure 4.6 : Insertion des données.....	32
Figure 4.7 : Diagramme flocon de neige	33
Figure 4.8 : Création d'un nouveau projet	34
Figure 4.9 : Création d'un projet d'informatique décisionelle	35
Figure 4.10 : Explorateur des soutions pour le projet mutidimensionnelle	35
Figure 4.11 : Sélectionner les mésures	36
Figure 4.12 : Sélectionner Une hiérarchie	36
Figure 4.13 : la déploiement de cube de données	37
Figure 4.14 : Connexion de serveur SSAS.....	37
Figure 4.15 : Architecture de IIS.....	38
Figure 4.16 : Les fichiers ISAPI de notre cube	40
Figure 4.17 : Connexion de SSAS à travers l'URL	41
Figure 4.18 : Créer un nouveau projet Java.....	43
Figure 4.19 : Créer un nouveau application java.....	43
Figure 4.20 : configurer le buid path	44
Figure 4.21 : ajouter des bibliothèques JAR.	44
Figure 4.22 : Le requête MDX pour le nombre des appels non null selon les villes.	45
Figure 4.23 : Exploitation par Power BI pour le nombre des appels.....	46
Figure 4.24 : diagramme de funnel pour l'exploitation par power BI.....	46
Figure 4.25 : Exploitation par Eclipse IDE	47