

Chapitre 4

Conception de la solution proposée

Cette partie concerne la modélisation et la conception d'un système d'information décisionnel pour le COUD qui nous permettra de concrétiser notre solution, c'est-à-dire de parvenir aux objectifs que nous nous sommes fixés tout au début. Ainsi nous allons en premier lieu présenter la conception architecturale d'un SID et finir par la présentation de la modélisation de la solution.

4.1 Conception architecturale

L'architecture d'un système décisionnel peut être représentée en quatre (4) parties.

- La première partie est celle des sources de données à savoir le système de gestion de l'entreprise contenant des bases de données opérationnelles ainsi que des sources externes.
- La deuxième partie concerne la récupération, la transformation des données, puis l'alimentation d'un entrepôt de données.
- La troisième partie est celui de l'exploitation de l'entrepôt en question.
- La quatrième partie fournit à l'utilisateur final le moyen de composer sa propre analyse et restitution des données. [22]

4.1.1 Architecture Technique

Une architecture technique dédiée pour un système décisionnel est constituée d'une base de données (un datawarehouse) pour le stockage, d'un outil Extraction Transformation Loading (ETL) pour alimenter le datawarehouse à partir des systèmes sources et de différents outils pour restituer les informations aux utilisateurs (reporting, analyse, outil statistique, etc.). Nous allons, à ce niveau, détailler les différentes étapes nécessaires à la mise en place de notre SID.

- Acquisition de l'information : Cette phase va faire intervenir des processus ETL qui se chargeront de récupérer toutes les données nécessaires depuis les différentes sources de

stockage. Il convient alors de localiser ces données. Ces données peuvent provenir de différentes sources (fichiers CSV, fichiers XML, PostgreSQL, DB2, ORACLE). Dans l'acquisition des données, nous allons donc procéder par les tâches ci-dessous :

- ◇ Extraction : collection de données utiles ;
 - ◇ Préparation : transformation des caractéristiques des données dans le modèle de l'entrepôt (filtrer, trier, homogénéiser, nettoyer) ;
 - ◇ Chargement : insertion des données dans l'entrepôt de données après nettoyage.
- Stockage : Les données sont chargées dans le datawarehouse pour traiter des applications décisionnelles.
 - Restitution : Mise à disposition des données pour les utilisateurs finaux afin de pouvoir les interroger, les visualiser ou les analyser. Il existe plusieurs outils de restitution (tableaux de bord, requêtes SQL, analyse multidimensionnelle, datamining...)

4.1.1.1 Schéma de l'application

Les données que nous utilisons proviennent de l'enquête que nous avons eu à faire avec GF. Ces données sont d'abord stockées dans un fichier Excel que nous nommons Enquête (réponses).xls. À partir de ce dernier, les données sont transférées dans une table SRC_Données d'une base de données DB_LogementEtudiant_SID et c'est à partir de là que le datawarehouse est bâti. Après le stockage des données propres dans le datawarehouse, ces données sont mises dans un cube pour des besoins métiers. À partir du cube ou même du datawarehouse, les données sont mises à disposition dans des tableaux de bord pour les utilisateurs finaux. Ci-dessous, nous avons l'architecture de notre système.

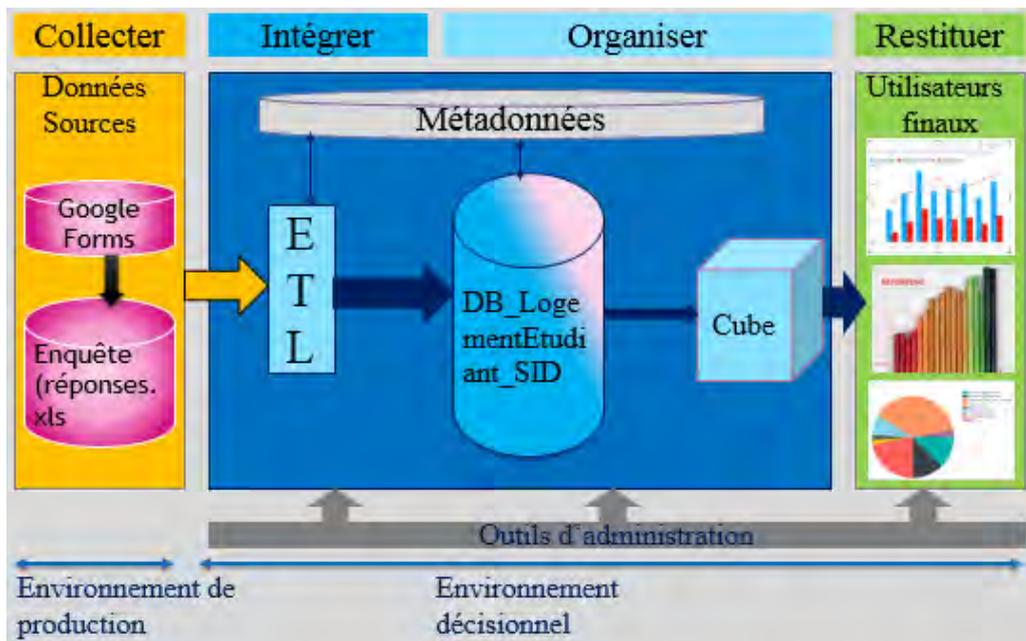


FIGURE 4.1 – Schéma de l'application

4.2 Modélisation de la solution

Dans notre modélisation, nous allons utiliser le schéma en flocon car elle normalise les dimensions et économise d'espace disque. Nous allons présenter une description du modèle.

4.2.1 Loger

Le modèle comprend quatorze (14) tables de dimensions et une table de fait centrale. Parmi les dimensions huit (8) sont directement liées au fait et les autres sont liées à certaines dimensions. Un logement fait intervenir un étudiant, le bâtiment où il est logé sachant que l'étudiant peut avoir un logement ou pas, le mois de son logement s'il est logé, la session de son logement, le département où est inscrit l'étudiant, l'état de santé de l'étudiant (si c'est un étudiant malade ou pas), le nombre de personnes qui occupe la chambre où est logé cet étudiant et le temps.

Un étudiant à son tour fait intervenir, la bourse (pour savoir si l'étudiant est boursier ou pas), sa situation matrimoniale, sa mention de l'année qu'il vient de quitter et de son âge.

Une Chambre fait intervenir le bâtiment où elle se trouve, c'est-à-dire connaître le bâtiment de la chambre en question.

Soit le modèle suivant :

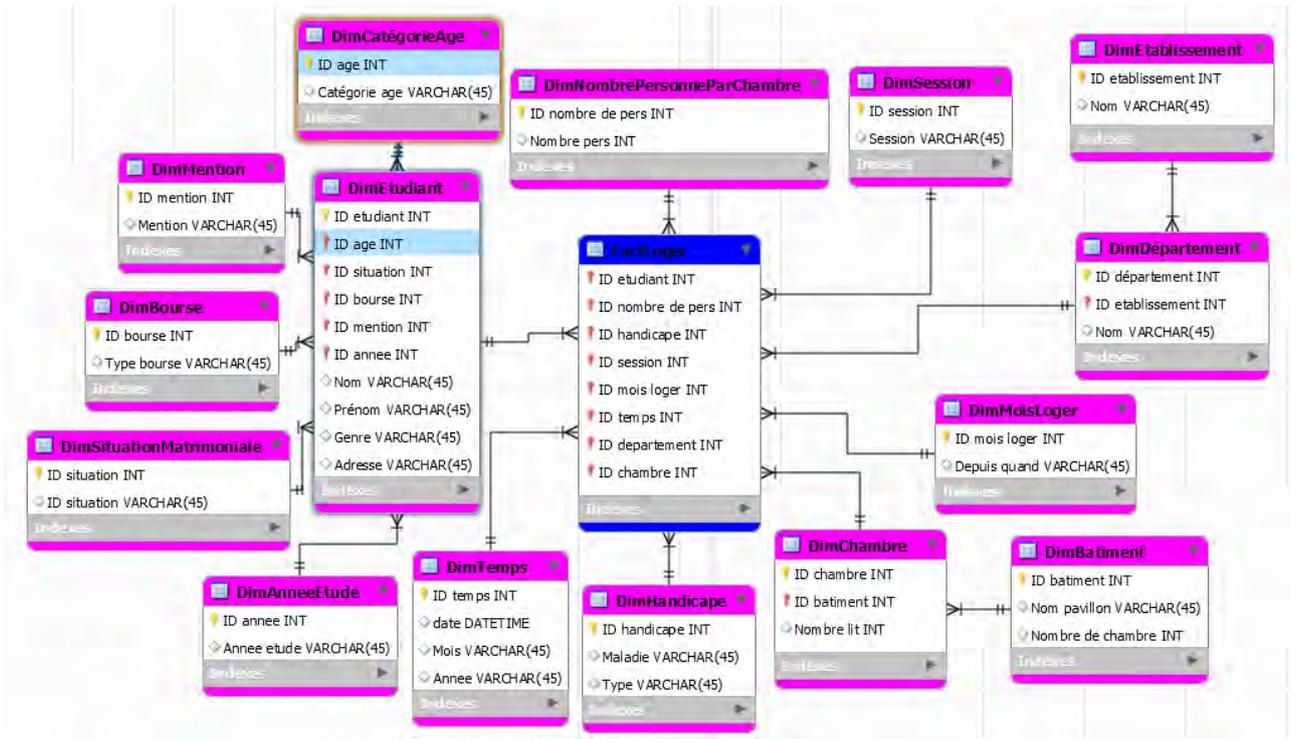


FIGURE 4.2 – Modèle conceptuel de la solution

Description des dimensions et de la table de fait

A noter que chaque dimension contient une clé de substitution ou clé technique (exemple : id_etudiant). Elle est arbitraire, c'est en général un numéro attribué par le système à une ligne de la dimension séquentiellement. Elle a pour but principal de garantir la première forme normale.

- La dimension **DimBourse** contenant comme attribut le type de bourse des étudiants
- La dimension **DimMention** contenant comme attribut les mentions des étudiants
- La dimension **DimCatégorieAge** contenant comme attribut la catégorie d'âge des étudiants
- La dimension **DimSituationMatrimoniale** contenant comme attribut la situation matrimoniale des étudiants
- La dimension **DimAnneeEtude** qui a comme attribut l'année où est inscrit l'étudiant
- La dimension **DimEtudiant** contenant la liste des étudiants avec comme attribut et clés techniques des dimensions qui l'entourent :
 - id_age
 - id_mention
 - id_bourse
 - id_situation

- id_annee
- nom
- prenom
- genre
- adresse
- La dimension **DimNombrePersonneParChambre** contenant le nombre de personne qui loge dans une chambre données
- La dimension **DimEtablissement** qui contient le nom de la faculté de l'étudiant
- La dimension **DimDepartement** contenant le nom du département et la clé technique de la dimension DimEtablissement (id_etablissement)
- La dimension **DimHandicap** contenant les étudiants qui sont malades ou pas avec comme attribut :
 - Maladie
 - Type
- La dimension **DimBatiment** qui contient la liste des pavillons qui existe dans cette université avec comme attribut le nom du pavillon et le nombre de chambre que contient le pavillon
- La dimension **DimChambre** contenant le nombre de lit des chambres dans un pavillon donné et la clé technique de la dimension DimBatiment (id_batiment)
- La dimension **DimSession** contenant comme attribut la session où l'étudiant est logé
- La dimension **DimMoisLoge** contenant l'attribut depuis quand
- La dimension **DimTemps** avec les attributs date, mois et année
- la table de faits **FactLogger** est composée des clés techniques des dimensions. Elle a comme attribut :
 - id_etudiant (provenant de DimEtudiant)
 - id_departement (provenant de DimDepartement)
 - id_temps (provenant de DimTemps)
 - id_nombre_pers (provenant de DimNombrePersonneParChambre)
 - id_handicap (provenant de DimHandicap)
 - id_session (provenant de DimSession)
 - id_moisloge (provenant de DimMoisLoge)
 - id_chambre (provenant de DimChambre)

Vu l'architecture de notre solution c'est seulement la première partie, correspondant à la collecte des données, qui a été réalisée. Nous avons pu faire la modélisation grâce à la collecte des informations en utilisant le modèle en flocon. Et les autres parties seront réalisées en se basant sur ce modèle.