



Laboratoire d'Analyse, Géométrie et  
Applications  
Mathématiques Appliquées et Calcul  
Scientifique



## Rapport de stage

Titre

---

---

Réalisation et intégration d'une solution BI  
paramétrable pour l'application "AGL PAIE"

---

---

Par

**M. Yassine Lgarh**

Encadré par :

**M. EL HAFID ABOUNAIM**

# TABLE DES MATIÈRES

<b>Remerciements</b>	<b>5</b>
<b>1 Présentation de l'entreprise</b>	<b>6</b>
<b>2 Cadre général du projet</b>	<b>7</b>
2.1 Définition de l'informatique décisionnelle . . . . .	7
2.2 Architecture décisionnelle . . . . .	9
2.3 Data Cleaning . . . . .	9
2.4 Processus d'Extraction, Transformation et Chargement . . . . .	10
2.5 Modélisation dimensionnelle . . . . .	11
2.5.1 Modèle en étoile . . . . .	11
2.5.2 Modèle en flocon de neige . . . . .	11
2.5.3 Modèle en constellation de faits . . . . .	12
<b>3 Conception</b>	<b>13</b>
3.1 Modèle conceptuel de données . . . . .	13
3.2 Choix du modèle multidimensionnel . . . . .	14
3.3 Data warehouse . . . . .	14

---

3.4	Datamart . . . . .	15
3.5	Dimension . . . . .	15
3.6	Faits . . . . .	15
<b>4</b>	<b>Présentation des outils utilisés</b>	<b>16</b>
4.1	Sql Server Management Studio (SSMS) . . . . .	16
4.2	Visual Studio . . . . .	17
4.3	SAP Crystal Reports . . . . .	17
<b>5</b>	<b>Réalisation</b>	<b>19</b>
5.1	Tables . . . . .	19
5.2	Data Cleaning . . . . .	20
5.3	Phase d'extraction, Transformation et Chargement . . . . .	21
<b>6</b>	<b>Reporting</b>	<b>25</b>
<b>7</b>	<b>Déploiement</b>	<b>29</b>
7.1	La connexion à la base de donnée du client . . . . .	30
7.2	Programmation événementielle . . . . .	31
<b>8</b>	<b>Annexes</b>	<b>33</b>
8.1	Quelques requêtes "Data cleaning" et transformations . . . . .	33
8.1.1	Création du Data Warehouse . . . . .	36
8.2	Les indicateurs . . . . .	37
8.3	Intégration . . . . .	39
	<b>Bibliographie</b>	<b>39</b>

---

J'ai effectué mon stage de fin d'études au sein de l'entreprise AGL Software, éditeur de logiciels de gestion financière, du 01 mars 2016 au 31 août 2016. Au cours de ce stage, j'ai eu pour missions de réaliser un modèle décisionnel paramétrable et l'intégrer à une application "AGL Paie". Ce stage m'a permis de travailler sur les différentes bases de données SQL server, découvrir le processus des projets informatiques décisionnels et la programmation événementielle (VB.net), au sein des différentes mission j'ai pu découvrir aussi les métiers liés aux ressources humaines "AGL PAIE" ainsi que quelques aspects de la gestion commerciale "AGL Commerciale".

## REMERCIEMENTS

Je tiens particulièrement à remercier M. El Hafid ABOUNAIM pour m'avoir pris en charge et suivi durant ces 6 mois de stage, il a su me guider et m'accompagner sur toute la durée du développement de mon stage.

Je souhaite remercier toute l'équipe informatique pour son soutien, ses explications dans les différentes étapes de mon projet ainsi que son accueil chaleureux qui m'a permis de m'intégrer rapidement au sein du service, ainsi que tout le personnel que j'ai eu l'occasion de rencontrer au cours de ce stage.

## CHAPITRE 1

# PRÉSENTATION DE L'ENTREPRISE

AGL Software est un éditeur de logiciel de gestion d'entreprise et intégrateur de ses solutions. Depuis 1998, la société propose une gamme complète de services informatiques et produits informatiques pour accompagner les PME/PMI durant leurs différents processus métiers (RH-Commercial-Comptabilité-Gestion de trésorerie..) et différents domaines (Services-Distribution-Administration-Hôtellerie-Industrie-Finance et Assurances-Textile...).

AGL software est présente sur l'ensemble du territoire marocain, avec un peu plus de 800 clients, elle a aussi conclu un partenariat exclusif avec des éditeurs Français de renommée internationale, EBP France et Gestimum France.

## CHAPITRE 2

# CADRE GÉNÉRAL DU PROJET

Le rendement du personnel est l'un des facteurs clé de succès d'une entreprise, La concurrence est de plus en plus rude sur tous les domaines et tous les secteurs, l'évolution technologique est devenu une arme à double tranchant, le but étant de trouver une solution d'aide à la décision afin de rendre les stratégies élaborées par le comité de l'entreprise efficace, précise et rapide. Ce projet à pour objectif la mise en place d'une solution BI (Business intelligence) paramétrable pour le département ressources humaines et l'intégrer à l'application "AGL PAIE".

### 2.1 Définition de l'informatique décisionnelle

L'informatique décisionnelle (BI : Business intelligence) et l'informatique dédiée aux dirigeants des entreprises, elle regroupe tous les informations et les données qui permettent aux décideurs d'agir efficacement et d'une manière optimale concernant un service ou une stratégie, ou simplement pour avoir une vue générale des données de l'activité étudiée.

On peut modéliser le processus de BI avec la figure suivante :

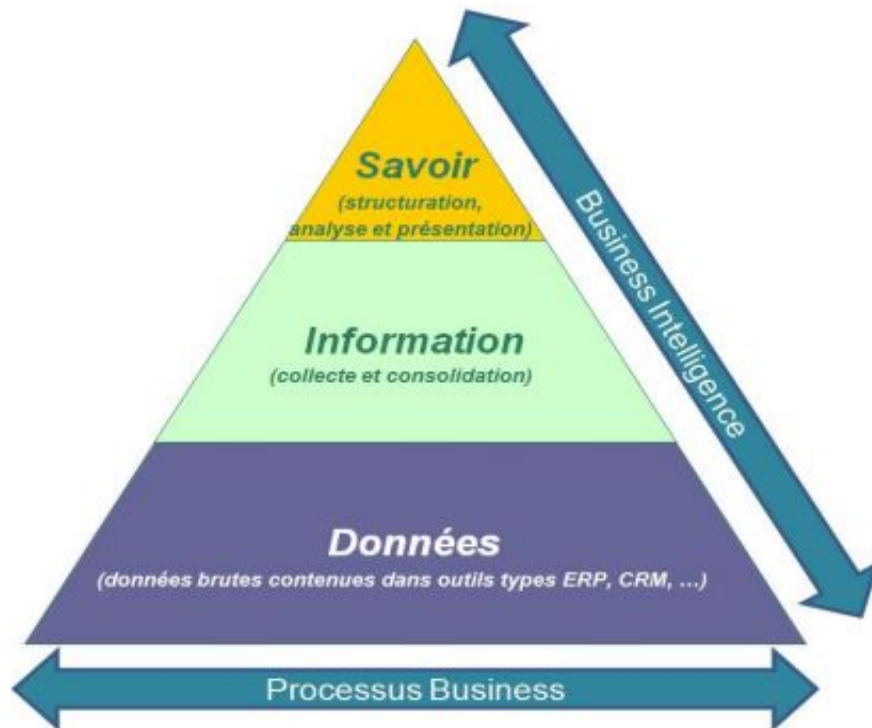


FIGURE 2.1 – Pyramide modélisant le processus de BI

L'informatique décisionnelle est un sujet phare, elle a connu une importante évolution ces dernières années, ce qui a enrichi l'ensemble de ses outils de reporting (Microsoft Power Bi, SAP Crystal reports, QlikView...) et de consolidations de données, ses outils essayent au maximum de simplifier la sélection et la gestion des données et proposent des options supplémentaires comme l'interactivité des indicateurs et l'hierarchie des dimensions. 7



## 2.2 Architecture décisionnelle

Une architecture BI (figure 2.2) est une entité qui regroupe l'ensemble des éléments (outils, méthodes, concepts ...) qui rentrent dans le processus BI.

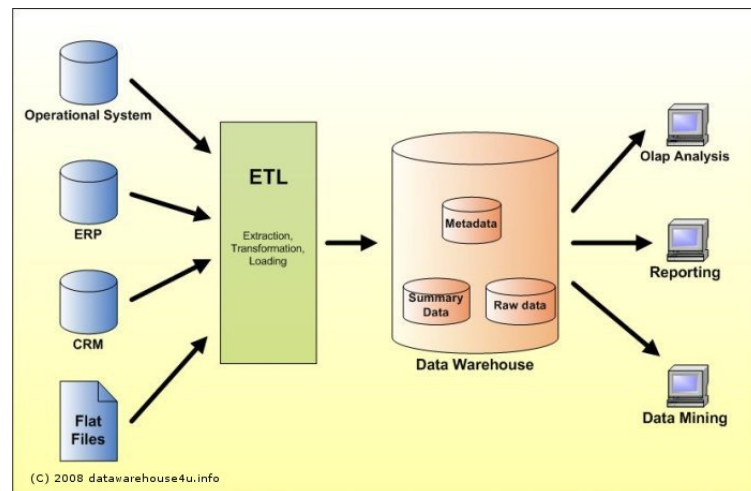


FIGURE 2.2 – Architecture BI

## 2.3 Data Cleaning

Le nettoyage de données (Data Cleaning) est le processus qui consiste à analyser les informations contenus dans une source de données, repérer les données erronées, et établir un dispositif de correction ou de suppression selon le cas.

Le nettoyage de données est essentiel pour un entrepôt de données, les données contenus dans ce dernier peuvent contenir plusieurs types d'erreurs et d'informations invalides qui peuvent nuire à la qualité de la solution BI finale et affecter les décisions prises en fonctions des éléments de cette solution.

## 2.4 Processus d'Extraction, Transformation et Chargement

Extraction-Transformation-Chargement (ETL pour Extract-Transform-Load), il s'agit d'une procédure permettant le transfert de données d'une source de données (dans notre cas c'est une base de données SQL) à l'entrepôt de données (Data warehouse), autrement dit L'ETL consiste à alimenter notre Data warehouse par les données issues de notre base de donnée.

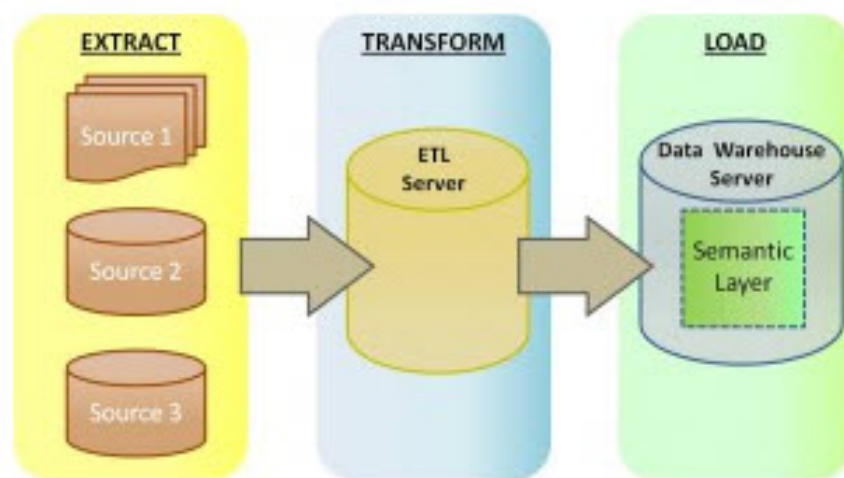


FIGURE 2.3 – Processus de l'ETL

À l'origine, les solutions d'ETL sont apparues pour le chargement régulier de données agrégées dans les data warehouses, avant de se diversifier vers les autres domaines logiciels. Ces solutions sont largement utilisées dans le monde bancaire et financier, ainsi que dans l'industrie, au vu de la multiplication des nombreuses interfaces.

## 2.5 Modélisation dimensionnelle

La modélisation dimensionnelle consiste en deux concepts : les faits et les dimensions.

Chaque modèle dimensionnel contient une table sur laquelle se centralise toute l'activité, c'est la table des faits, les autres tables représentent des dimensions, elles contiennent des informations complémentaires à la table des faits, et ces derniers sont liés avec la tables des faits à l'aide d'une clé primaire.

### 2.5.1 Modèle en étoile

Le modèle en étoile est un modèle dont le schéma dimensionnel est représenté par une étoile (figure 2.4).

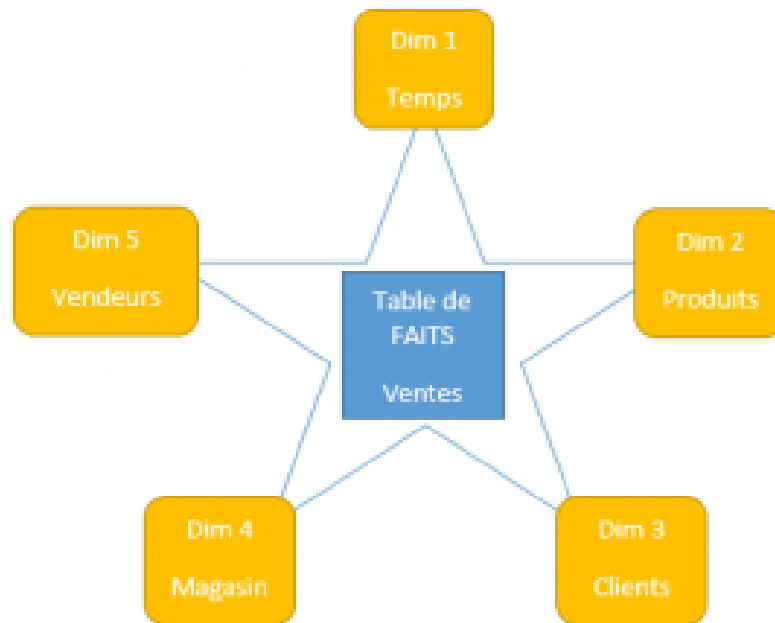


FIGURE 2.4 – Exemple d'un modèle en étoile

### 2.5.2 Modèle en flocon de neige

Le modèle en flocon de neige est un modèle qui ressemble au modèle en étoile à une différence près, ici les dimensions sont hiérarchisées d'une façon très détaillée, ces sous dimensions sont liées par hiérarchie, ce qui donne au modèle une forme de flocon de neige (figure 2.5) .

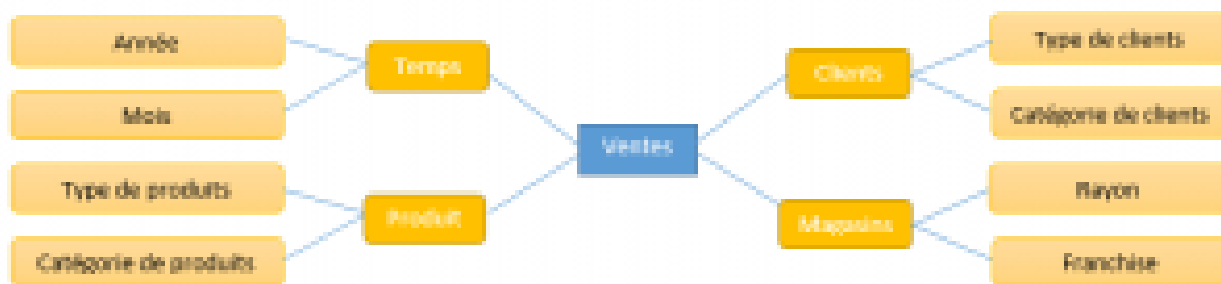


FIGURE 2.5 – Exemple d'un modèle en flocon de neige

### 2.5.3 Modèle en constellation de faits

Ce modèle est un ensemble de schémas en étoiles et/ou en flocon dans lesquels les tables de faits se partagent certaines tables de dimensions. C'est de cette accumulation que découle un modèle en constellation.<sup>2</sup>



FIGURE 2.6 – Exemple d'un modèle en constellation de faits

## CHAPITRE 3

---

## CONCEPTION

Dans ce chapitre , on va présenter le coeur du travail effectué, autrement dit, la conception de l'entrepôt de données. Pour commencer on va d'abord présenter le modèle conceptuel de données réalisé, ensuite les indicateurs du tableau de bord détaillés et le schéma de notre conception multidimensionnelle.

### 3.1 Modèle conceptuel de données

On a été amené à concevoir un modèle conceptuel de données qui a permis de représenter et de mettre en clair les données relatives aux services auxquels nous nous intéressons.

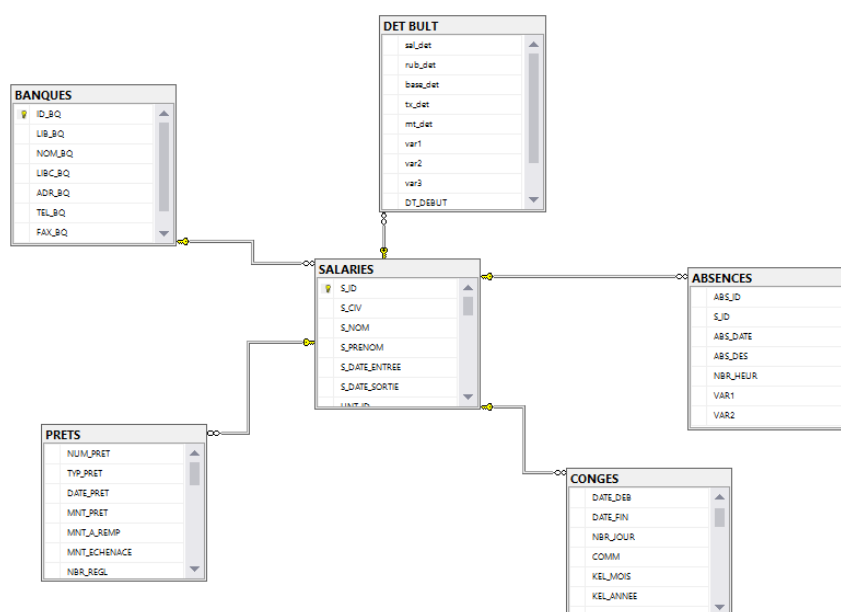


FIGURE 3.1 – Modèle conceptuel de données

## 3.2 Choix du modèle multidimensionnel

On a présenté dans le premier chapitre les différents schémas d'un DWH (Datawarehouse) : en étoile, en flocon de neige et en constellation. Dans notre cas on a opté pour le modèle en étoile, qui est le modèle classique. Il est considéré comme étant le modèle avec lequel l'ETL est plus simple et les jointures moins compliquées à effectuer.

Bien que le modèle en étoile présente aussi quelques inconvénients comme la redondance dans les dimensions, surtout dans les dimensions qui présentent une hiérarchie, mais il s'agit du modèle le plus adapté à notre cas.

## 3.3 Data warehouse

Un entrepôt de données, ou data warehouse, est un regroupement de toutes les informations de l'entreprise. C'est une structure qui se distingue de la base de donnée opérationnelle de l'entreprise, car elle ne contient que les données utiles pour élaborer les stratégies et pour la prise de décisions.<sup>7</sup>

Le Data warehouse peut être très volumineux des fois, en général, on le décompose en

petites entités nommés Datamart.

## 3.4 Datamart

Un datamart ou magasin de données est un sous-ensemble d'un data warehouse qui ne contient que le modèle dimensionnel spécifique à un seul service (ou activité). L'usage des datamart est très important, il permet une certaine organisation dans le data warehouse ce qui empêche l'interaction des données des différents services.

## 3.5 Dimension

Une dimension est une table qui contient les axes d'analyse qui vont être examinés afin d'offrir les renseignements nécessaires à la prise de décision.

## 3.6 Faits

La table des faits représente la table sur laquelle va porter l'analyse, c'est la table qui contient l'information cible (la table SALARIES dans notre cas), cette table combinée avec les autres dimensions vont permettre de spécifier les propriétés de chaque salarié suivant les axes d'analyse.

## CHAPITRE 4

# PRÉSENTATION DES OUTILS UTILISÉS

Dans ce chapitre, nous allons citer les outils utilisés au cours de ce projet pour les différents étapes de conception, reporting et intégration.

### 4.1 Sql Server Management Studio (SSMS)



FIGURE 4.1 – SQL SERVER Management Studio 2014



SQL Server Management Studio (SSMS) est un environnement dédié aux bases de données, il permet de visualisation, la gestion, l'administration et le développement des différents composants SQL server. SSMS intègre une interface graphique qui contient plusieurs fonctionnalités et offre à l'utilisateur une vue générale de tous les données en main. Source : Microsoft

## 4.2 Visual Studio



FIGURE 4.2 – Visual Studio

Visual Studio est l'environnement de développement Microsoft, il permet de générer des applications Webd Asp.net, des services Web XML, des application bureau ou mobiles. C#, Vb.net et Visual C++ utilisent le même environnement de développement IDE (Integrated Development Environment) ce qui leur permet de partager plusieurs outils (classes-références...) afin de faciliter la création des solutions et centraliser les ressources. (Source: Microsoft)

## 4.3 SAP Crystal Reports

Crystal Reports est un progiciel dédié à l'informatique décisionnelle, qui permet de concevoir des rapports à l'aide de différents source de données informatiques, il génère des présen-



FIGURE 4.3 – SAP Crystal Reports

tations graphiques très conviviales ce qui le rend un outil idéal pour le reporting Parmi les données d'entrées accessibles à partir de Crystal Reports :

- Bases de données telles que Sybase, SAP HANA, IBM DB2, Microsoft Access, Microsoft SQL Server, MySQL Interbase, SQLite et Oracle.
- Classeurs Microsoft Excel.
- Fichiers HTML XML.
- SAP : BW..

## CHAPITRE 5

---

## RÉALISATION

Ce chapitre va comprendre la mise en oeuvre de la solution Bi, c'est la dernière étape du projet décisionnel.

La détermination des indicateurs clés nécessaires à intégrer passe d'abord par la détermination d'axes d'analyse selon le besoin du service des ressources humaines et selon les données disponibles.

Les axes d'analyse essentiels sur lesquels notre étude va porter sont :

- Temps
- Départements
- Services
- Salariés
- Fonctions
- Banques

### 5.1 Tables

Les données servant d'axes d'analyse de ce projet ont déjà été présentées dans la figure 3.1, on voit bien que la table "SALARIES" est la table fait dans ce modèle, et que les autres tables représentent les dimensions.

## 5.2 Data Cleaning

On a choisi d'effectuer cette partie de nettoyage de données à l'aide de l'outil SSIS intégré à Visual studio qui nous permet de schématiser les procédures d'une façon claire et propre.

On Commence par créer une tâche de flux, cette tâche de flux permet le déplacement des données entre les différents sources et destinations(ADO .NET, OLE DB, EXCEL, XML, fichier flat...), elle permet aussi à l'utilisateur de transformer et modifier ces données à l'aide d'un outil de mapping ou à l'aide des commandes SQL.

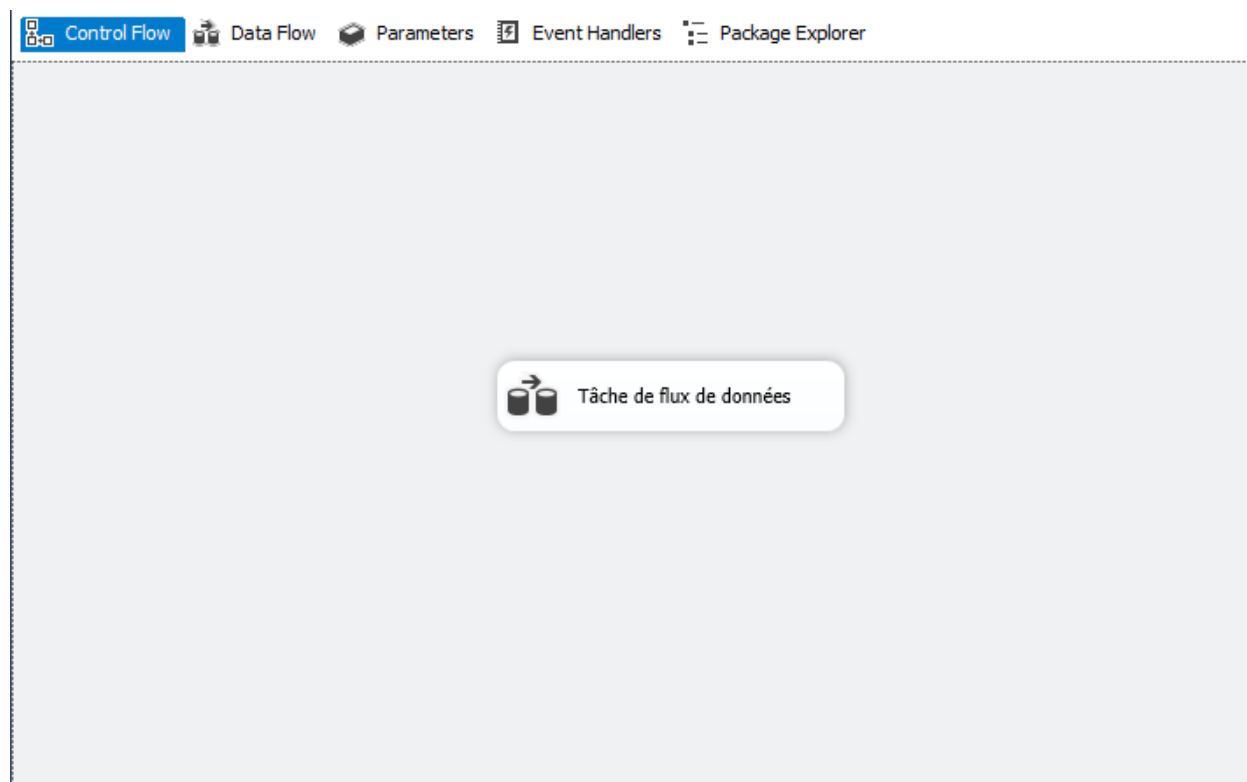


FIGURE 5.1 – Tâche de flux de données

On assure ensuite les connexions et les destinations avec les Objets OLE DB qui permettront à ADO .NET de localiser les données dans les serveurs "SQL Server". Concernant le processus du nettoyage (modification et suppression) on agit directement sur la table en question avec des commandes SQL.

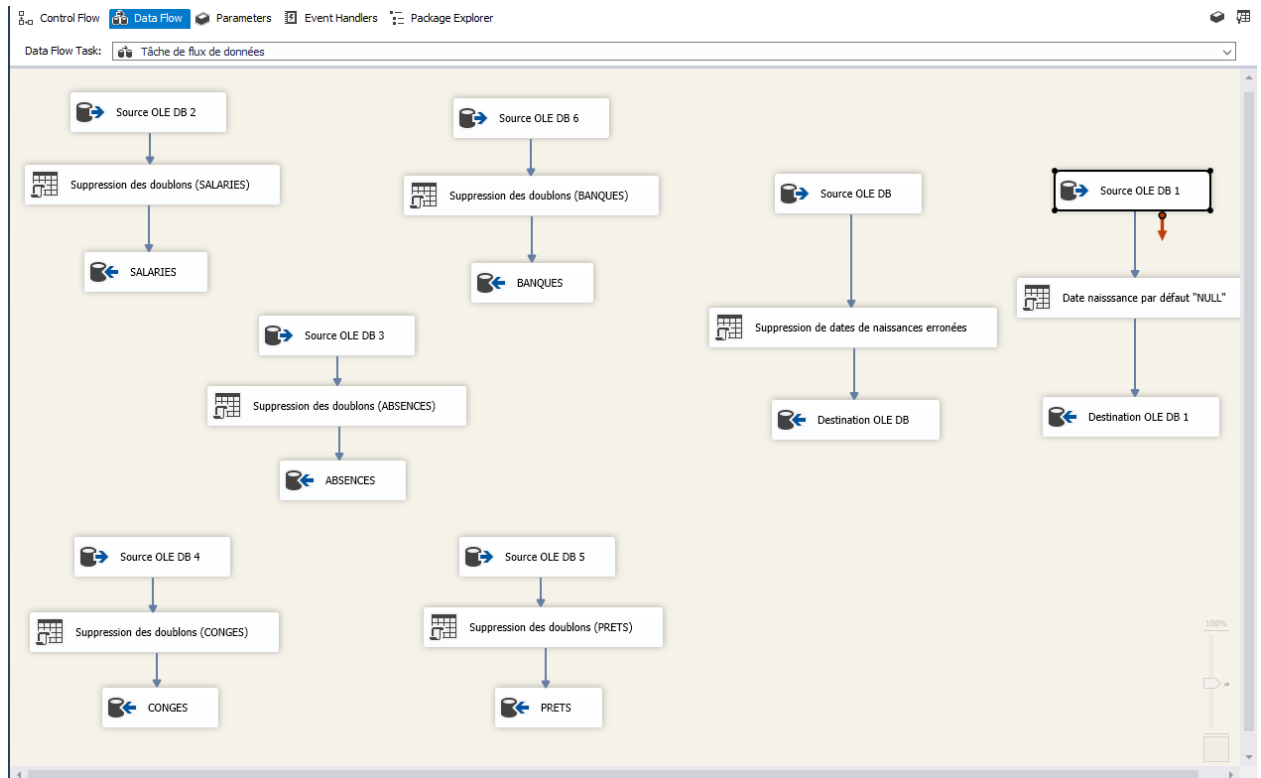


FIGURE 5.2 – Procédé data cleaning

### 5.3 Phase d'extraction, Transformation et Chargement

La Phase ETL est la première phase d'un projet décisionnel, son but est de créer l'entrepôt de données. A cet égard, nous avons travaillé dans l'environnement SSIS sur Visual studio qui permet d'extraire les données à partir d'une ou plusieurs bases opérationnelles, les transformer si besoin à l'aide de la boîte à outils, et les charger finalement dans une destination (Data warehouse).

Pour réaliser cette partie dont l'objectif est d'alimenter notre Data warehouse, nous avons chargé les données de la base de données à l'aide du gestionnaire de connexions OLE DB qui permet la connexion entre un package et une source de données, c'est le choix le plus judicieux dans notre cas afin d'utiliser directement le fournisseur OLE DB pour SQL Server.

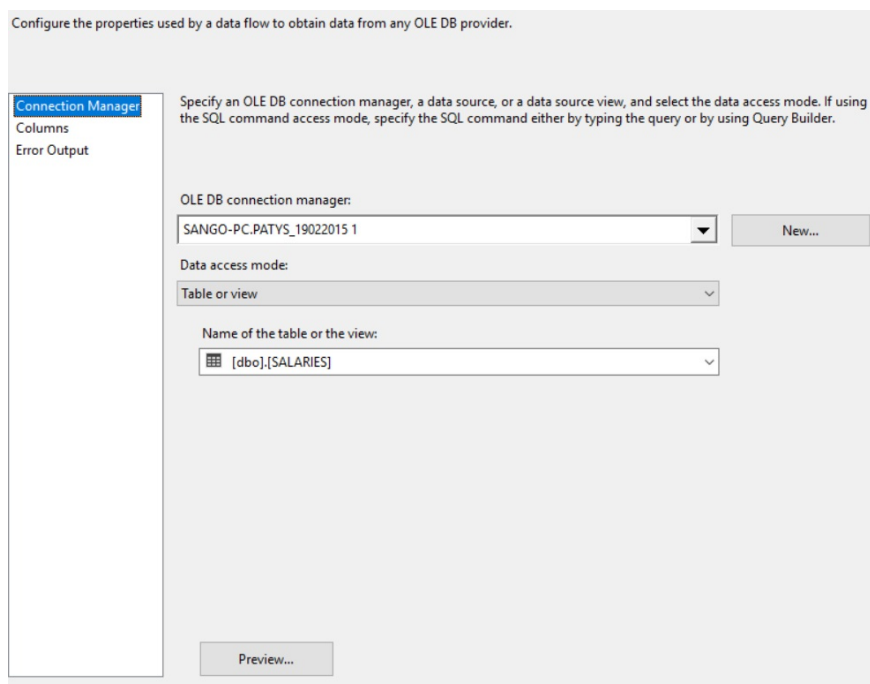


FIGURE 5.3 – Extraction des données de la table SALARIES

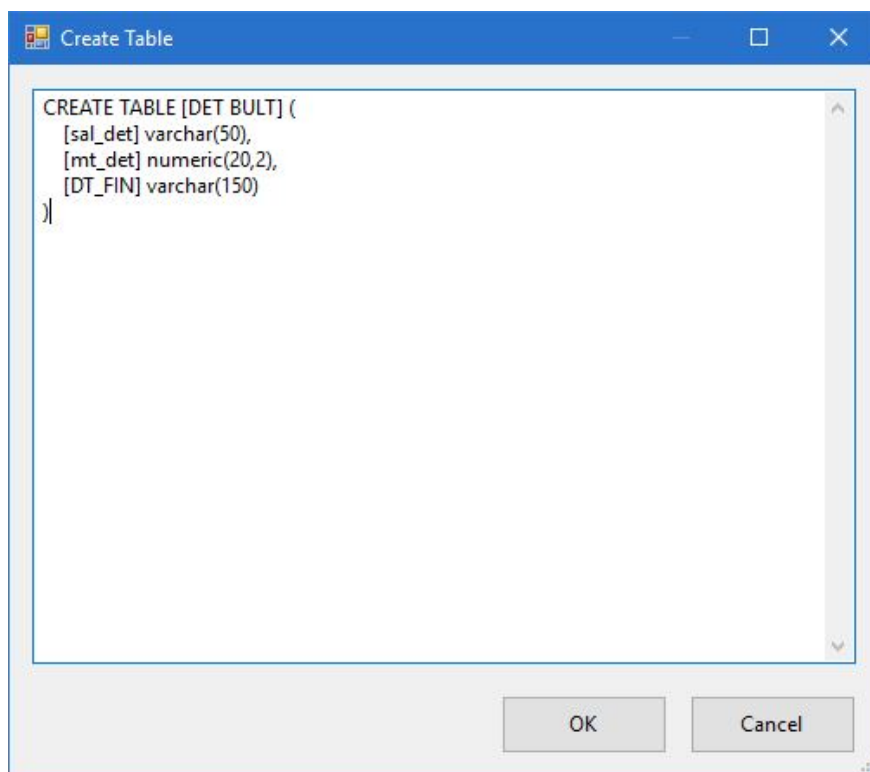


FIGURE 5.4 – Création de la table DET BULT dans le warehouse

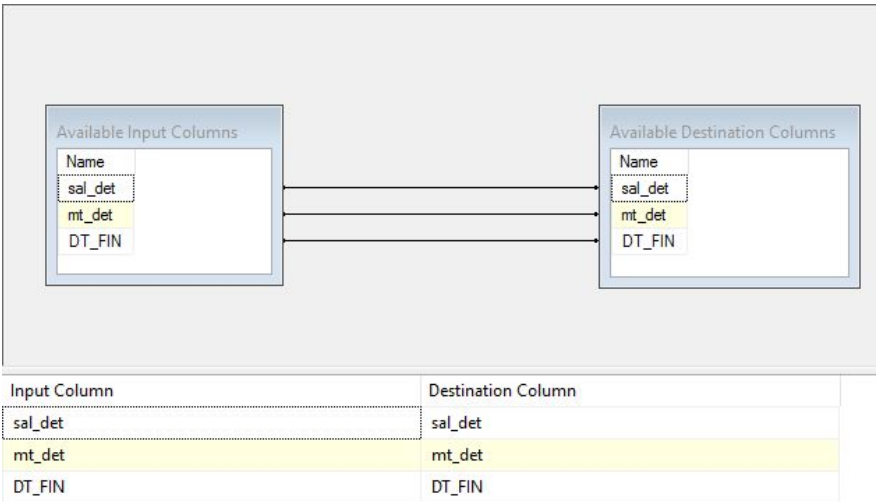


FIGURE 5.5 – Mapping des colonnes de la table "DET\_BULT"

Pour la partie transformation, les seules transformations nécessaires à ce stade se résument à la sélection des colonnes nécessaires pour notre rapport ainsi que la déclaration des clés primaires et étrangères à l'aide des Contraintes SQL lors de la création des tables du data warehouse étant donné que les transformations nécessaires ont déjà été effectuées durant le data cleaning.

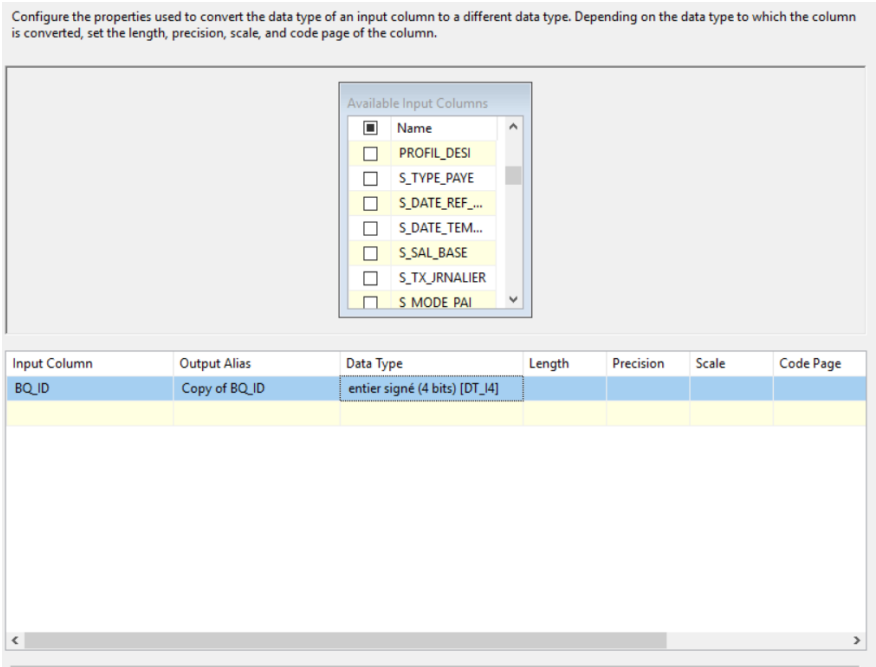


FIGURE 5.6 – Conversion de la clé étrangère BQ\_ID de la table SALARIES

Le chargement des tables vers l'entrepôt de données s'effectue à l'aide de l'outil destination OLE DB.

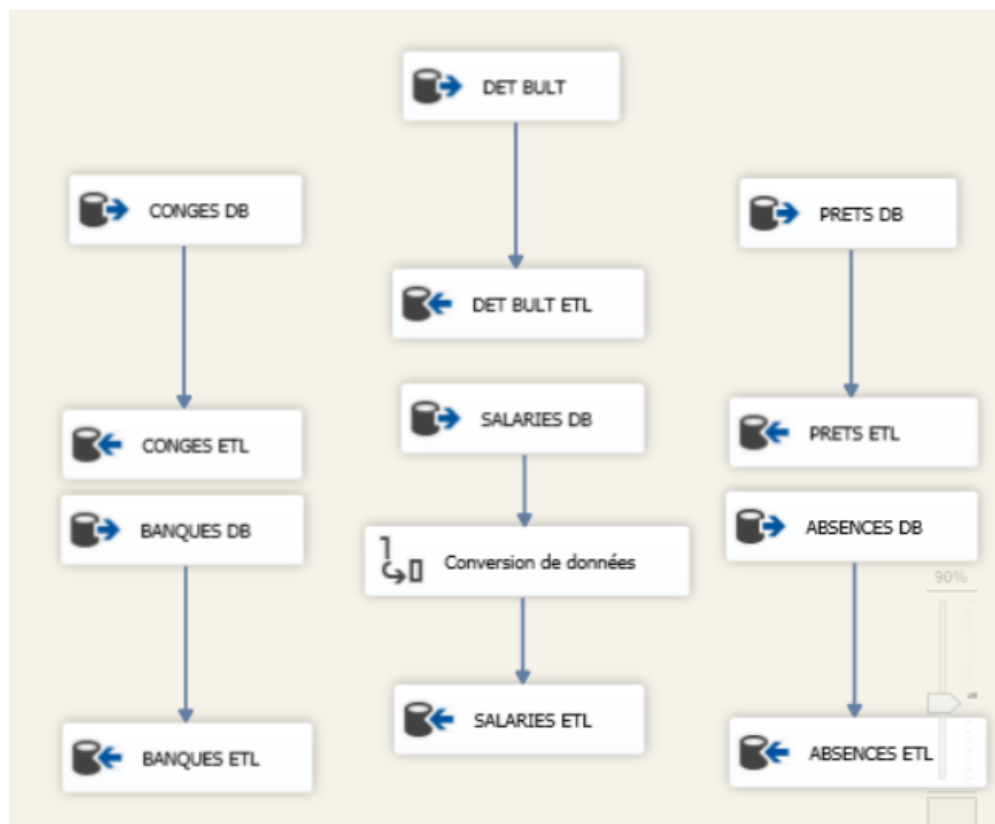


FIGURE 5.7 – la vue ETL



Cette dernière étape consiste à présenter les informations à valeur ajoutée de telle sorte qu'elles apparaissent plus lisibles et significatives afin d'offrir à l'utilisateur une expérience intuitive.

Les données sont principalement modélisées par des représentations à base de requêtes afin de constituer des tableaux de bord ou des rapports via des outils d'analyse décisionnelle.

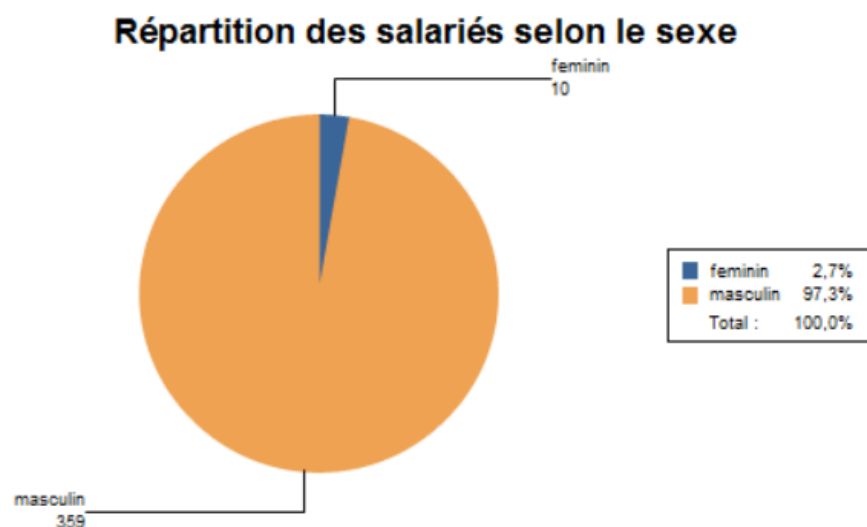


FIGURE 6.1 – Répartition des salariés selon le sexe

La notion d'effectif est un élément clé et essentiel de la gestion des ressources humaines. Elle constitue ainsi le centre d'intérêt des services R.H au sein des organisations et, des pôles de contrôle de gestion sociale quand ils ont été mis en place. Or, beaucoup d'entreprises ignorent les détails liés à leurs effectifs à un moment donné.

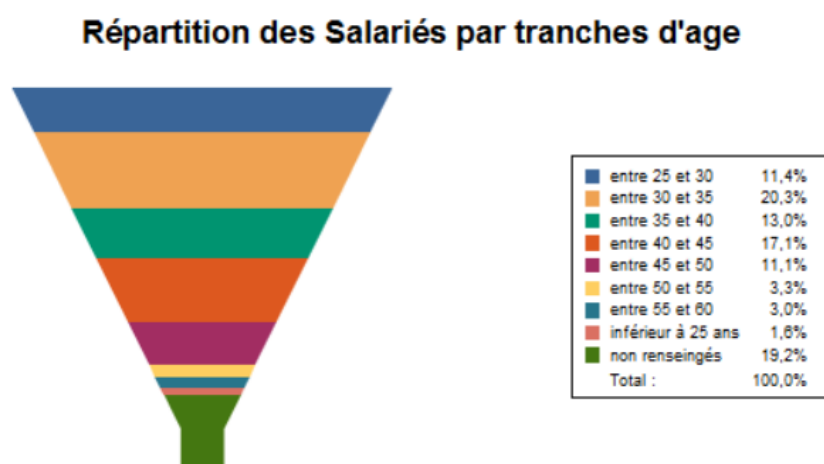


FIGURE 6.2 – Répartition des effectifs selon la tranche d'âge

La gestion des âges est un élément majeur de la culture organisationnelle qui a une valeur importante aux yeux des dirigeants RH, et qui doit toujours être vérifiée et traitée en priorité.

La gestion des âges repose aussi sur les principes de la justice sociale ainsi que l'égalité et le respect de chaque individu quel que soit son âge.

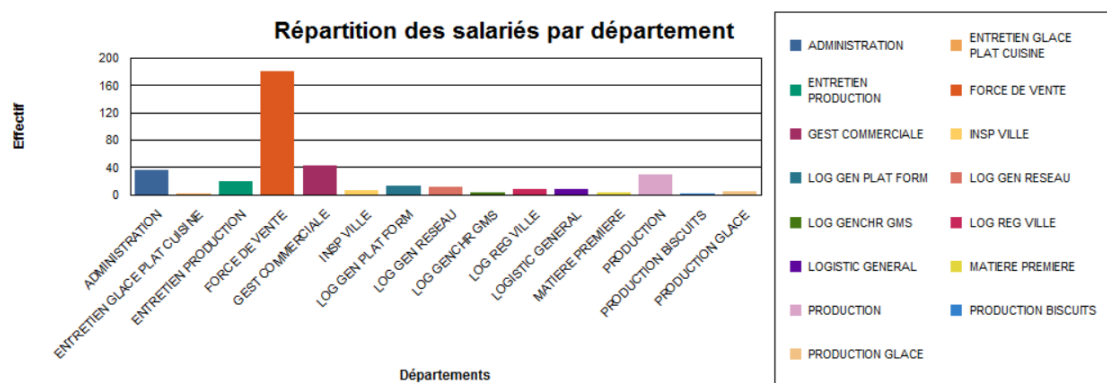


FIGURE 6.3 – Répartition des salariés par département

L’effectif des départements est un autre élément clé pour les chargés RH, constater un déséquilibre stratégique ou recruter des nouveaux collaborateurs, cet indicateur reste d’une importance primordiale pour le service.

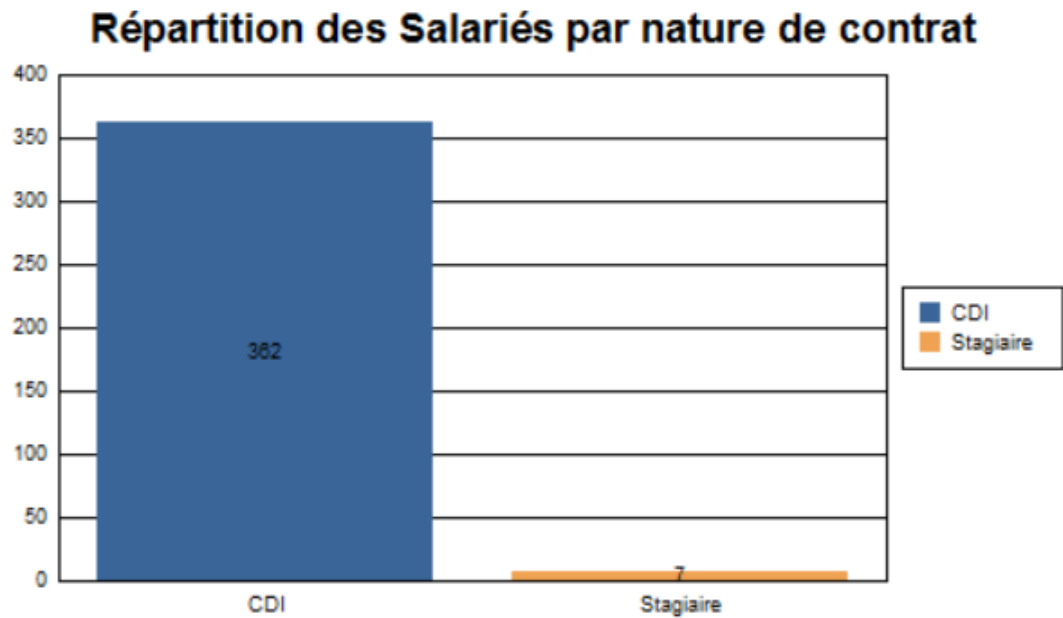


FIGURE 6.4 – Répartition des salariés par nature de contrat

Il faut toujours garder un œil sur le type de contrat de chaque salarié afin d’agir selon les lois et les réglementations au bon moment, et aussi pour préserver le quota imposé par l’état pour les apprentis par exemple.

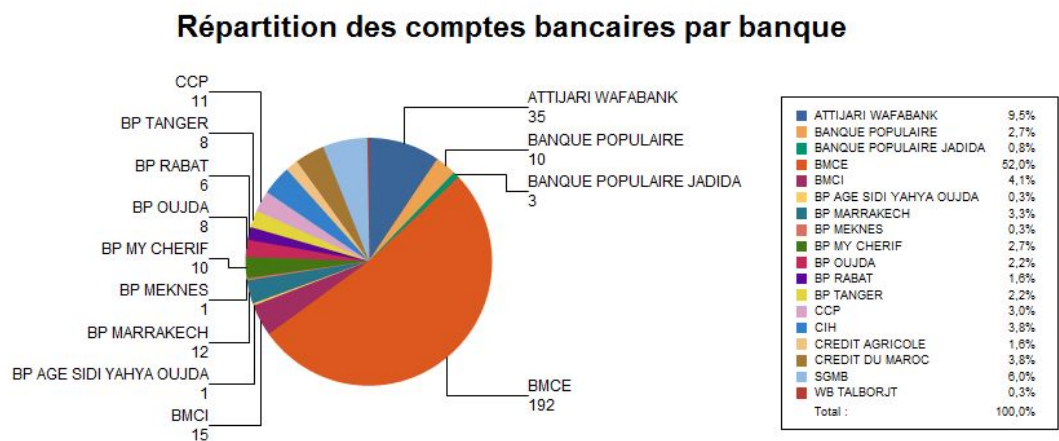


FIGURE 6.5 – Répartition des comptes bancaires des salariés par banque

Connaître la répartition des comptes bancaires peut économiser plusieurs frais supplémentaires à la société.

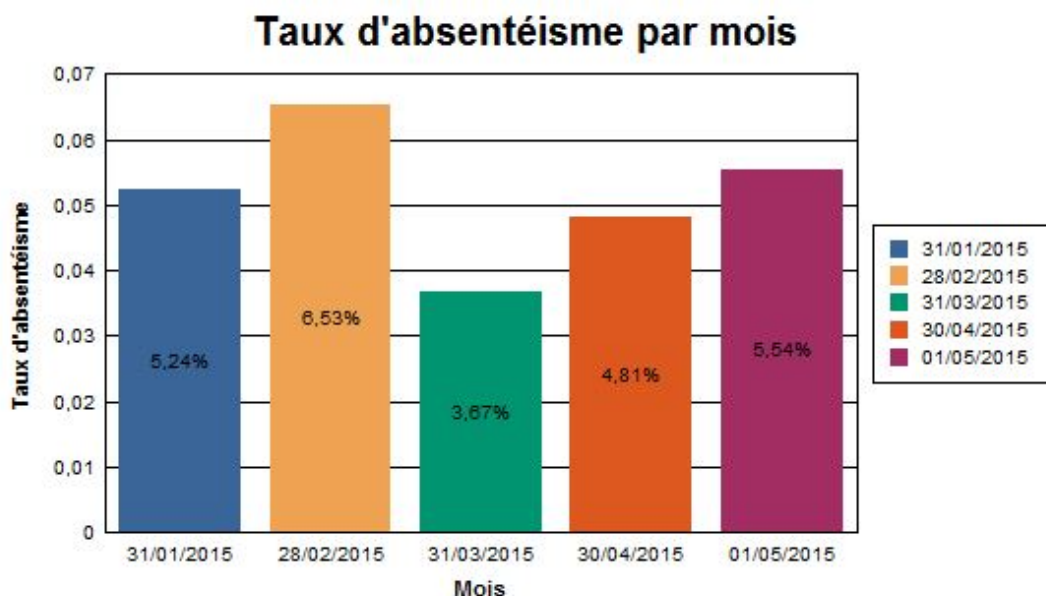


FIGURE 6.6 – Taux d’absentéisme par mois

Le taux d’absentéisme, généralement calculé à partir des absences circonstanciées et non prévisibles (maladie, retard, motif personnel, non autorisé), représente le nombre total d’heures d’absences exprimé en pourcentage des heures disponibles pour travailler. L’indicateur d’absentéisme mesure la productivité et la mobilisation d’une entreprise, Il peut également être utilisé comme un prédicateur de Turn-over.

Le tableau de bord est ensuite enregistré au format RPT qui a l’avantage d’être interactif et peut donner en direct un filtrage. Ce format peut stocker les données provenant de sources multiples et différents types de bases de données ce qui permettra par la suite au rapport de s’adapter aux données de chaque client.

## CHAPITRE 7

---

## DÉPLOIEMENT

Après la création du tableau de bord sous l'extension .rpt il ne reste plus que son intégration dans l'application, le rapport devra être accessible pour le client à partir de l'application.

On a opté pour SAP Crystal Reports qui est l'outil de création d'états idéal pour les programmes .NET, il est fourni avec Visual Studio, il permet l'alimentation des rapports des paramètres à partir de l'application et la connexion avec la base de données via ADO.NET ainsi que le transfert des paramètres à partir de l'application. Pour effectuer ce déploiement

on utilisera les espaces de noms "CrystalDecisions" suivants :

- CrystalDecisions.CrystalReports.Engine : assure la prise en charge du moteur d'état.
- CrystalDecisions.ReportSource : fournit les classes qui comprennent une couche entre le Windows Forms Viewer ou le Web Forms Viewer et le moteur CrystalReports. Ces classes traitent les demandes des visualiseurs. Elles déterminent ensuite comment contacter le moteur et effectuer l'appel pour le visualiseur.
- CrystalDecisions.Shared : fournit des classes, des interfaces et des énumérations qui sont partagées par le Web Forms Viewer, le Windows Forms Viewer et le moteur Crystal Reports.

- CrystalDecisions.Windows.Forms : assure la prise en charge du Windows Forms Viewer et de ses classes associées.

## 7.1 La connexion à la base de donnée du client

Afin de se connecter à une base de données sur ADO.NET on utilise un fournisseur de données (Data Provider) correspondant à la base de données qui contient l'information visée.

Sur le Framework 3.5 et les versions plus récentes, il existe quatre types de fournisseurs intégrés :

- Sql Server
- OLE DB
- ODBC
- Oracle

Toutes les classes permettant d'utiliser ces fournisseurs se trouvent dans l'espace de nom System.Data.

Chaque fournisseur de données assure la communication avec un type spécifique de base de données à l'aide d'une API (Application Programming Interface est une interface qui permet l'accès à un logiciel), ces fournisseurs nous servent d'intermédiaire entre la base de données et notre application. 5

Ainsi le fournisseur de SQL Server est optimisé pour fonctionner avec nos bases de données SQL Server.

Ado.net propose deux modes d'accès aux bases de données :

- Le mode connecté : Dans ce mode, Ado.net effectue une connexion avec la base de données et agit sur cette dernière directement avec des requêtes SQL. L'application reste connectée à la base de données jusqu'à ce que la connexion soit fermée.
- Le mode déconnecté : Dans ce mode, on travaille avec ce qu'on appelle un Dataset, c'est une copie en mémoire des données de la base de données.

On a choisi le mode connecté pour réduire l'espace mémoire utilisée et pour actualiser le rapport avec les dernières données disponibles.

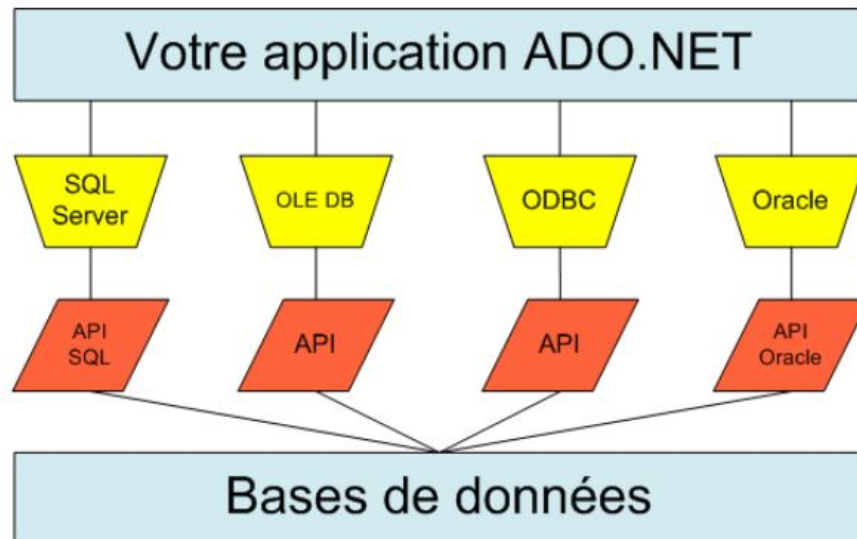


FIGURE 7.1 – Schémas représentant le processus de connexion par les 4 fournisseurs de données.

La Connexion entre l'application et la base de données client se fait à l'aide d'un fichier qui contient les informations suivantes :

- Le nom du serveur contenant la base de données.
- L'identifiant pour ce serveur.
- Le mot de passe.
- le nom de la base de données.

## 7.2 Programmation événementielle

La programmation événementielle (ou procédurale) est une programmation fondée sur les événements, à l'opposé de la programmation séquentielle, le programme sera principalement défini par ses réactions aux différents événements qui peuvent se produire, c'est-à-dire des changements d'état de variable, par exemple l'incrément d'une liste, un mouvement de souris ou de clavier.



32



## 8.1 Quelques requêtes "Data cleaning" et transformations

```
/* Suppression les doublons de la table SALARIES */
WITH SALARIESCMPT AS
(SELECT S_ID, S_NOM, S_PRENOM , ROW_NUMBER() OVER(PARTITION BY S_ID ORDER
        BY S_ID) AS RowNumber
FROM SALARIES)
DELETE FROM SALARIESCMPT WHERE RowNumber > 1

/*Suppressions des doublons de la table BANQUES*/
WITH BANQUESCMPT AS
(SELECT ID_BQ, ROW_NUMBER() OVER(PARTITION BY ID_BQ ORDER BY ID_BQ) AS
        RowNumber
FROM BANQUES)
/*select * from BANQUESCMPT*/
DELETE FROM BANQUESCMPT WHERE RowNumber > 1

/*Supprime doublons ABSENCES*/
```

```
WITH ABSENCESCMT AS
(SELECT ABS_ID, ROW_NUMBER() OVER(PARTITION BY ABS_ID ORDER BY ABS_ID)
      AS RowNumber
FROM ABSENCES)
/*select * from ABSENCESCMT*/
DELETE FROM ABSENCESCMT WHERE RowNumber > 1

/*Supprime doublons CONGES*/
WITH CONGESCMPT AS
(SELECT id_conge, ROW_NUMBER() OVER(PARTITION BY id_conge ORDER BY
      id_conge) AS RowNumber
FROM CONGES)
/*select * from CONGESCMPT*/
DELETE FROM CONGESCMPT WHERE RowNumber > 1

/*Supprime doublons PRETS*/
WITH PRETSCMPT AS
(SELECT NUM_PRET, ROW_NUMBER() OVER(PARTITION BY NUM_PRET ORDER BY
      NUM_PRET) AS RowNumber
FROM PRETS)
/*select * from PRETSCMPT*/
DELETE FROM PRETSCMPT WHERE RowNumber > 1

/* Suppression des dates erronées */
update SALARIES set S_DATE_NAISS=NULL WHERE YEAR(S_DATE_NAISS) < 18

/* Suppression de quelques lignes erronées */

delete CONGES where S_ID= '00257P' or S_ID='0890'

delete PRETS where PRETS.S_ID not in (Select PRETS.S_ID from PRETS
      INNER JOIN SALARIES on SALARIES.S_ID=PRETS.S_ID)
```

```
/* Transformations */
alter table [SALARIES] alter column S_ID VArchar(50) not null

alter table SALARIES alter column BQ_ID int

UPDATE SALARIES SET [SALARIES].S_DATE_NAISS = NULL WHERE YEAR(
    S_DATE_NAISS)=1900

UPDATE SALARIES SET [SALARIES].S_DATE_SORTIE = NULL FROM SALARIES WHERE
    YEAR(S_DATE_SORTIE)=1900

/* Déclaration des clés primaires et des clés étrangères */

ALTER TABLE [PATYS_19022015].[dbo].[SALARIES]
ADD PRIMARY KEY (S_ID)

ALTER TABLE BANQUES
ADD PRIMARY KEY (ID_BQ)

ALTER TABLE CONGES
ADD CONSTRAINT const_Congés
FOREIGN KEY (ID_SAL)
REFERENCES SALARIES(S_ID)

ALTER TABLE PRETS
ADD CONSTRAINT const_Prêts
FOREIGN KEY (S_ID)
REFERENCES SALARIES(S_ID)

ALTER TABLE SALARIES
ADD CONSTRAINT const_SALARIES
FOREIGN KEY (BQ_ID)
REFERENCES BANQUES (ID_BQ)
```

```
Select S_NOM, BQ_ID from SALARIES  
DELETE from SALARIES where BQ_ID is null
```

### 8.1.1 Création du Data Warehouse

```
/* Création de l'entrepôt de données */  
CREATE DATABASE Warehouse  
  
/* Création des tables */  
  
CREATE TABLE [SALARIES] (  
[S_ID] varchar(50) NOT NULL PRIMARY KEY,  
[DIV_DESI] varchar(300),  
[STA_DESI] varchar(100),  
[BQ_ID] int NOT NULL,  
[S_DATE_NAISS] datetime,  
[S_SEX] varchar(20),  
)  
  
CREATE TABLE [ABSENCES] (  
[ABS_ID] int NOT NULL PRIMARY KEY,  
[S_ID] varchar(50) NOT NULL,  
[NBR_HEUR] int,  
)  
  
CREATE TABLE [BANQUES] (  
[ID_BQ] int NOT NULL PRIMARY KEY,  
[NOM_BQ] varchar(50),  
)  
  
CREATE TABLE [PRETS] (  
[NUM_PRET] INT NOT NULL PRIMARY KEY,  
[TYP_PRET] varchar(150),  
[DATE_PRET] varchar(50),  
[MNT_PRET] numeric(24,2),  
[S_ID] INT NOT NULL,  
)  
  
CREATE TABLE [DET BULT] (  

```

```

[sal_det] INT NOT NULL,
[mt_det] numeric(20,2),
[DT_FIN] varchar(150)
)

CREATE TABLE [CONGES] (
[S_ID] INT NOT NULL,
[DATE_DEB] datetime,
[DATE_FIN] datetime,
[id_conge] int NOT NULL PRIMARY KEY,
)

```

## 8.2 Les indicateurs

```

/* Indicateur des comptes bancaires */
select ID_BQ, NOM_BQ, count(S_ID) as compteur from BANQUES INNER JOIN
    SALARIES ON [SALARIES].BQ_ID=[BANQUES].ID_BQ group by ID_BQ, NOM_BQ
    order by compteur Desc

/* Indicateur pyramide des âges */
SELECT *, CASE
WHEN (Year(getdate())-Year(S_DATE_NAISS)) < 25 THEN 'inférieur à 25 ans'
WHEN 30>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))> 25 THEN 'entre 25 et 30'
WHEN 35>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS)) > 30 THEN 'entre 30 et 35'
WHEN 40>(Year(getdate())-Year(S_DATE_NAISS))and (Year(getdate())-Year(
    S_DATE_NAISS))>35 THEN 'entre 35 et 40'
WHEN 45>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))>40 THEN 'entre 40 et 45'
WHEN 50>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))>45 THEN 'entre 45 et 50'
WHEN 55>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))>50 THEN 'entre 50 et 55'
WHEN 60>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))>55 THEN 'entre 55 et 60'
WHEN 65>(Year(getdate())-Year(S_DATE_NAISS)) and (Year(getdate())-Year(
    S_DATE_NAISS))>60 THEN 'entre 60 et 65'

```

```
ELSE 'non renseingés'
END
FROM SALARIES

/* Indicateur du taux d'absentéisme */

Select avg(mt_det), (26-avg(mt_det))*100/26 as TA,DT_FIN from DET_BULT
where rub_det=1023 group by DT_FIN

/* Quelques indicateurs pour le tableau de bord Gestion commerciale */

SELECT top 3 DETAILS_DC.DD_LIB, sum(DOCUMENTS.DC_MT_TTC) as DC_MT_TTC
FROM [dbo].[DETAILS_DC] Inner join DOCUMENTS on DOCUMENTS.DC_NUM=
DETAILS_DC.DC_NUM where DOCUMENTS.DC_TYPE='V' and DOCUMENTS.DC_STYPE=
'F' and DD_TYPE='P' group by DD_LIB order by DC_MT_TTC DESC

Select TIER_RS, Sum((ECH ARECEV-ECH APAYER)-(ECH RECU-ECH PAYER)) as
ECHEANCES, ECH_DATE from ECHEANCIER group by TIER_RS, ECH_DATE

SELECT DETAILS_DC.DD_LIB, DETAILS_DC.DD_QTE, DOCUMENTS.DC_DATE as
La_date
FROM [dbo].[DETAILS_DC] Inner join DOCUMENTS on DOCUMENTS.DC_NUM=
DETAILS_DC.DC_NUM where DOCUMENTS.DC_TYPE='V' and DOCUMENTS.DC_STYPE=
'F' and DD_TYPE='P' group by DD_LIB, DD_QTE, DC_DATE order by
DETAILS_DC.DD_QTE DESC

SELECT distinct DETAILS_DC.DD_LIB, sum(DETAILS_DC.DD_QTE) as Quantity
FROM [dbo].[DETAILS_DC] Inner join DOCUMENTS on DOCUMENTS.DC_NUM=
DETAILS_DC.DC_NUM where DOCUMENTS.DC_TYPE='V' and DOCUMENTS.DC_STYPE=
'F' and DD_TYPE='P' group by DD_LIB order by Quantity DESC
```

## 8.3 Intégration

```
'' Language = VB.NET  
'' En-tête  
Imports CrystalDecisions.CrystalReports.Engine  
Imports CrystalDecisions.Shared  
Imports CrystalDecisions.Windows.Forms  
Imports CrystalDecisions.ReportSource  
  
'' effectuer la connexion avec la base de données client à l'aide de l'  
    objet sqlConnection  
Conn.open()  
'' Créer l'objet rapport  
Dim repDoc As ReportDocument = New ReportDocument()  
repDoc.Load(".....\CrystalReport1.rpt")  
Dim etatName As String = "Rapport_Paie.rpt"  
CrystalReportViewer1.ReportSource = repDoc  
  
'' Rajout d'un paramètre raison_sociale  
Dim paramFields As ParameterFields = New ParameterFields()  
Dim paramField As ParameterField = New ParameterField()  
Dim paramDiscreteValue As ParameterDiscreteValue = New  
    ParameterDiscreteValue()  
Dim raison_sociale As String = raison_sociale()  
'' Transfert du paramètre "raison sociale" au tableau de bord  
paramField.ParameterFieldName = "raison_sociale"  
paramDiscreteValue.Value = raison_sociale  
paramField.CurrentValues.Add(paramDiscreteValue)  
paramFields.Add(paramField)  
Conn.close()
```

---

## BIBLIOGRAPHIE

- [1] ARCHAMBEAU Tony, "Base Du Langage SQL et Des Bases de Données",
- [2] GRIM Yazid and BLAIN Fleur-Anne, "Conception D'un Entrepôt de Données (Data Warehouse)," Club Des Développeurs et IT pro,
- [3] LASSERRE Philippe, "Cours VB.NET",
- [4] PÉDEHOURCQ David, "Création et Déploiement D'états Crystal Reports Avec VS .NET" (Club des développeurs et IT pro, July 15, 2004)
- [5] GRAND Patrice, "ADO.NET".
- [6] LAFARE Matthieu, Thèse Professionnelle "Business Intelligence" (HEC Paris, 2006)
- [7] Business Intelligence : Place de la BI et pilotage des projets décisionnels dans les grandes entreprises-CIGREF-2009.
- [8] Wikipedia