PROJECT REPORT ON

« Project data warehouse for university students »

Submitted By :

Chebicheb Kheireddine

NARDJIS BELOUNIS

MOKHTARI HADJ MOHAMED

LANTRI FAROUK

MIMOUNE ABDELAZZIZ

Under The Guidance Of :

Prof.Sir Omar Talbi

Department of Informatique

14 November 2022

Universitè Ibn Khaldoun Faculty of Mathematics and Computer Science

Contents

Figure 3
Context 6
Problem 6
What are the processes and the right analyzing method to achieve the decisions that we need ? 6
Objective of project ? 6
Implementation: Data warehouses 6
Organization of our report 6
Chapter 1 8
Data Warehouse 8
I. INTRODUCTION 9
II. DATA WAREHOUSE HISTORY 9
III. Data warehouse definition 9
Data Warehouse Architecture 11
IV. Definitions of the data warehouse's function and significance 13
Data Warehouse of a University? 13
Benefits of a University Data Warehouse? 13
Uses of University Data Warehouse 14
V. Conclusion 15
Chapter 2 16
Steps to build: Data warehouses 16
.1 INTRODUCTION 17
2. Steps for creating data warehouses 17
.2.1 Determine the Business Objectives of a University's Data 17
2.2. Configuring Physical Environments 17
2.3. Introducing Data Modeling 23
2.3.1. Dimensional Modeling 24
2.3.2. Elements of Dimensional Data Model 24
2.3.2.1. Fact 24
2.3.2.2. Dimension 24
2.3.2.3. Attributes 24
2.3.2.4. Fact Table 25
2.3.2.5. Dimension Table 25
2.4. Choosing ETL Solution 26

tDBConnection 33
tDBClose 34
tDBClose Standard properties 34
tDBInput Standard properties 34
The tMap: 35
tDBOutput 35
tDBOutput Standard properties 35
The problem: 38
Finally 39
2.5. Online Analytics Processing (OLAP) Cube 39
.2.5.1 What is OLAP Schema Workbench? 40
Uses of OLAP Schema Workbench 40
2.5.2. Make OLAP Schema 40
2.5.3. Add a Data Source 41
2.6. Creating A Front End 47
2.6.1. What exactly is Jasper Server? 47
2.6.2. Steps to design the report 48
Chapter 3 55
Implement website 55

Figure

Figure 1 Sample Enterprise Data Warehouse (EDW) 10
Figure 2 Operational Data Store 11
Figure 3 Data mart diagram 11
Figure 4 Data Warehouse Architecture 13
Figure 5 before and after delete the tables 18
Figure 6 before and after Delete duplicate columns 19
Figure 7 an example of a student you have to eliminate 19
Figure 8 an example of a student have different information 20
Figure 9 an example of breaking certain tables into smaller tables 21
Figure 10 an example of fixing certain table errors 21
Figure 11 an example of fixing certain table errors 2 22

Figure 12 some table add to building the data warehouse 23
Figure 13 University Data Model Star Schema 26
Figure 14 the step of creating a job in the repository 28
Figure 15 The database connection in Talend 29
Figure 16 Access connection in Talend 30
Figure 17 MySQL connection in Talend 31
Figure 18 Platte section to develop a Talend component 31
Figure 19 The second way to develop a Talend component 32
Figure 20 The tPrejob and tPostjob Components 33
Figure 21 tDBConnection Components 33
Figure 22 tDBCommit Components 34
Figure 23 tDBClose Components 34
Figure 24 tDBInput Components 35
Figure 25 tMap component 35
Figure 26 tDBOutput Components 36
Figure 27 The crucial job that will be used in all projects 36
Figure 28 The first job container under the name job is access to MySQL. 37
Figure 29 the second job under the name job fact 37
Figure 30 the previous job, job olap, uses tmap to establish the relationship between the tables 38
Figure 31 ETL based data warehouse architecture diagram 39
Figure 32 pentaho schema workbench interface 41
Figure 33 Add a Data Source using the Pentaho Schema Workbench Interface 42
Figure 34 an example of how to create a Schema 43
Figure 35 an example of how to create a Schema 43
Figure 36 an example of making a cube in schema 44
Figure 37 an example of setting Dimension in cube 44
Figure 38 an example of a dimension attribute 45
Figure 39 45
Figure 40 an example of a dimension with a hierarchy 45
Figure 41 an example of a level attribute 46
Figure 42 an example of every dimension in our cube 46
Figure 43 This cube's ultimate output for our project 47
Figure 44 an example of a log in page and creating a new OLAP data source by giving the OLAP cube schema connection information 48
Figure 45 An example of how to select data and create a new report 49

Figure 46 web interface to design and customize the report layout, which will include adding dimensions and measures from the OLAP cube to the report fields, ... 49
Figure 47 An example of graphs data report 50
Figure 48 All of our project's reports 50
Figure 49Material rate by year 51
Figure 50Total number of students by year 51
Figure 51Number of males and females by year 52
Figure 52Number of males and females by year and cycel 52
Figure 53Number of university baccalaureate disciplines by year 53
Figure 54 Total number of students by cycle 53
Figure 55 Students Degrees by university years Specialisation. 54
Figure 56 Total grades for each year. 54
Figure 57 Unit rate by year 54
Figure 58 our logo 56
Figure 59Data bases of our website 58
Figure 60 Home page 59
Figure 61 About page 59
Figure 62 Our team 59
Figure 63 Contact Page 60
Figure 64 add files 60

Context

Data collection, organization, and analysis are all part of data analysis. Numerous sources, including surveys, interviews, and databases, can be used to gather data. To find patterns, trends, and relationships, the data must next be arranged and examined. This facilitates decision-making and identifies areas that need improvement.

Problem

What are the processes and the right analyzing method to achieve the decisions that we need ?

Objective of project ?

Having a data warehouse creates a single source of truth and aids in breaking down data silos. The accuracy and integrity of data are also improved.

Universities may better understand their operations and make more informed decisions by using data warehouses, which provide them a comprehensive picture of the data.

Implementation: Data warehouses

We must comprehend what it is in order to apply data warehouses? and how to use it?

Organization of our report

Our structure is divided into three chapters:

Chapter 1

We discuss data warehouse history, definition, and application processes in this chapter.

Chapter 2

This chapter will discuss the design and construction of data warehouses.

Chapter 3

Introduction

This chapter will put our project into practice on the website to track it and promote our progress.

Chapter 1

Data Warehouse

I.INTRODUCTION

This chapter outlines the background, definition, and implementation procedures for creating a university's data warehouse.

II.DATA WAREHOUSE HISTORY

The concept of data warehouses first came into use in the 1980s when IBM researchers Paul Murphy and Barry Devlin developed the business data warehouse. American computer scientist Bill Inmon is considered the “father” of the data warehouse due to his authorship of several works, such as the Corporate Information Factory and other topics on the building, usage, and maintenance of the data warehouse. Inmon wrote the first book, held the first conference, and offered the first classes on data warehouses and is known for his creation of the definition of a data warehouse – “a subject-oriented, nonvolatile, integrated, time-variant collection of data in support of management’s decisions.”

III.Data warehouse definition

A data warehouse is a system used for reporting and data analysis. It is a central repository of data from multiple sources, including operational systems and external sources. Data warehouses store current and historical data in one single place that are used for creating analytical reports for knowledge workers throughout the enterprise.

Data warehouses allow businesses to consolidate data from multiple sources, including operational systems, and external sources. This data is stored in a single, centralized repository, allowing for faster and more efficient analysis and reporting. Data warehouses also provide a platform for data analysis that is more secure and reliable than traditional data sources.

Types of Data Warehouse

Three main types of Data Warehouses (DWH) are:

1. Enterprise Data Warehouse (EDW):

The Enterprise Data Warehouse (EDW) is a centralized warehouse. It offers decision support services to the entire organization. It provides a consistent method for arranging and displaying data. Additionally, it offers the option of categorizing data by subject and granting access in accordance with such divisions.

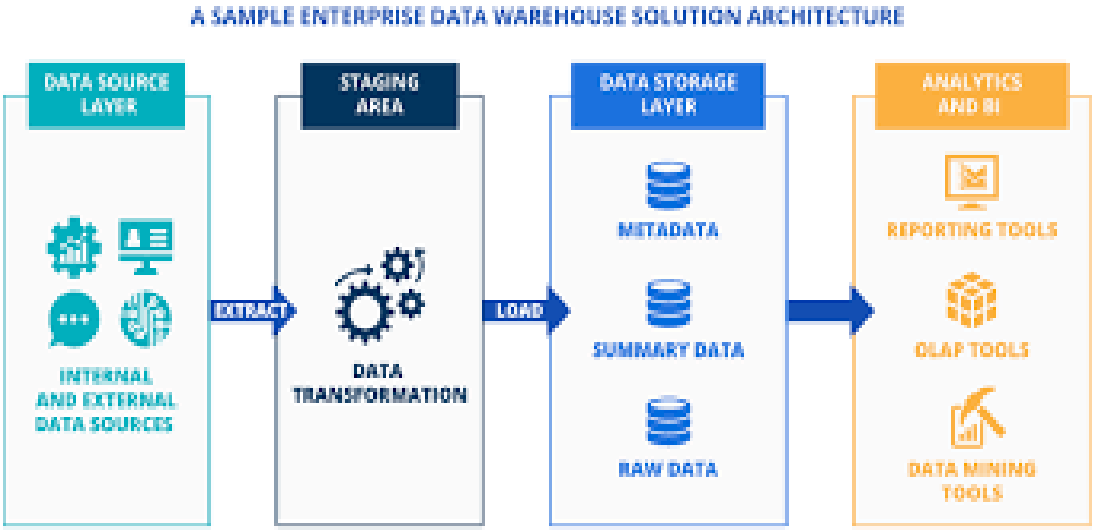


Figure 1 Sample Enterprise Data Warehouse (EDW)

2. Operational Data Store:

Operational Data Stores, commonly known as ODS, are nothing more than data stores needed when neither data warehouse nor OLTP systems are able to meet an organization's reporting requirements. ODS's data warehouse receives real-time updates. For regular tasks like keeping employee records, it is therefore extensively used.

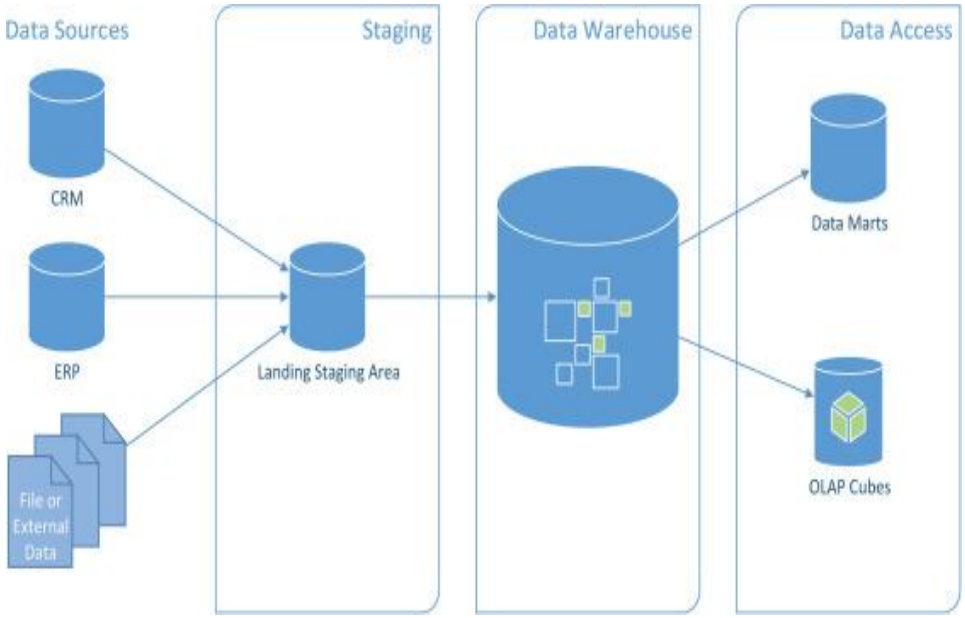


Figure 2 Operational Data Store

3. Data Mart:

The data warehouse includes a data mart as a subset. It was created specifically for a certain industry, such sales, finance, sales, or finance. Data can be gathered from sources directly and stored in an independent data mart.

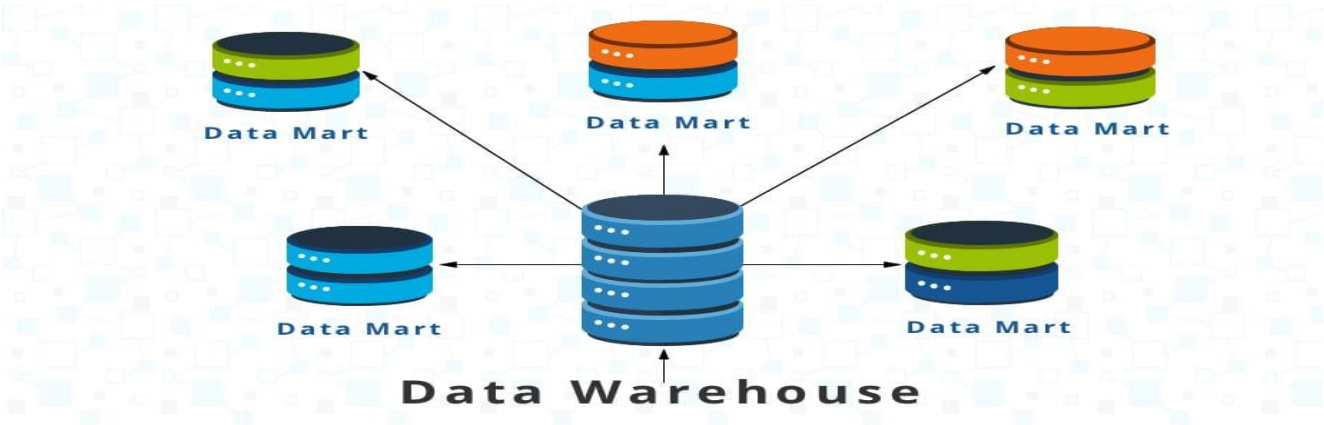


Figure 3 Data mart diagram

Data Warehouse Architecture

There is only one hardware layer in the single tier data warehouse design. One hardware layer makes up this hardware layer. Three methods—single tier, two tier, and three tier—can be used to build a data warehouse layer.

Definitions of the data warehouse's function and significance

Single-tier architecture: A single-layer design meant to conserve as little data space as possible. The actual world seldom ever employs this arrangement.

Two-tier architecture: A data warehouse is a collection of data in a format that is simple to modify and load into a database. There are several ways to establish data warehouses, so it's critical to choose the one that best suits your company's requirements. Scalability is the most crucial factor to think about. Consider employing a data warehouse if you need to store a lot of data in a small place.

Three-Tier Data Warehouse Architecture: The term "Top Tier" refers to the top, middle, and bottom tiers of this data warehouse architecture.

1. Relational database technology makes up the Datawarehouse's basic layer. Usually, a relational database system makes up this database system. In this layer, data is cleaned, transformed, and loaded.
2. Either ROLAP- or MOLAP-based intermediate tier OLAP servers are available. It performs the function of a middle layer OLAP server, abstracting OLAP from the end user. Middle tier OLAP servers are described as data warehouses that enable end-user interaction with the database and OLAP servers that abstract OLAP from the end user.
3. Being the first point of engagement with the data, the top-front-end tier's client layer is crucial. It is the place where data is displayed to the user and where choices are made using the data. Real-time data must be used and processed fast by the top-tier front-end client layer. Working with data that is in a format that experts can utilize and comprehend is also crucial. Top-tier data is often stored in a relational database, although it might also be a file or a stream. Validated, well-organized, and formatted in a way that makes data profiling and analytics simpler are all requirements for top-tier data.

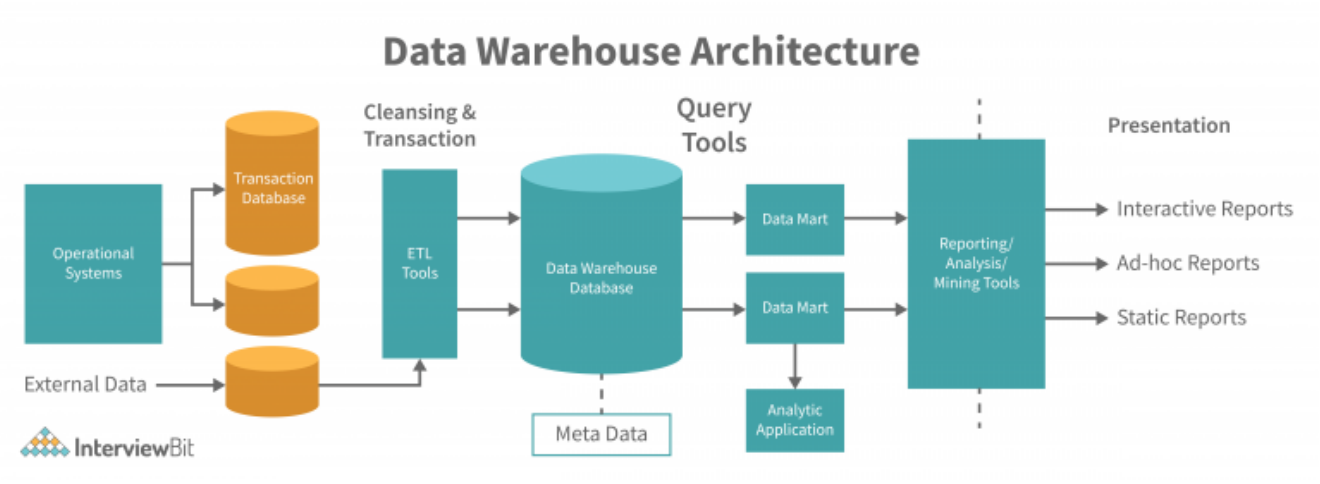


Figure 4 Data Warehouse Architecture

IV.Definitions of the data warehouse's function and significance

Data Warehouse of a University?

A university's data warehouse serves as a single location for all university-related data, such as student records, enrollment statistics, financial information, and more. Reporting, analysis, and decision-making employ this data. The data warehouse is intended to give the institution a single source of truth, enabling greater data governance and increased accuracy.

The enrollment and performance of students, faculty accomplishments, and financial information may all be monitored using a university data warehouse. It may also be used to produce reports and insights into the operation of the university, enabling better-informed choices. Additionally, performance patterns over time may be tracked using the data warehouse, enabling more efficient planning and budgeting.

Benefits of a University Data Warehouse?

A university data warehouse offers several advantages, including greater data governance, more accurate data, and more effective reporting and analysis. Additionally, it offers the institution a single source of reality, facilitating improved decision-making. Finally, performance patterns over time may be tracked using the data warehouse, enabling more efficient planning and budgeting.

A university data warehouse also gives the institution insights into its performance, enabling it to make decisions with greater knowledge. It may also be used to monitor

teacher performance, financial data, and student enrolment and performance. This information may be used to produce reports and insights into the operation of the institution, enabling more effective planning and budgeting.

Uses of University Data Warehouse

There are several uses for a university data warehouse, including reporting, analysis, and decision-making. It may also be used to monitor teacher performance, financial data, and student enrolment and performance. This information may be used to produce reports and insights into the operation of the institution, enabling more effective planning and budgeting.

In order to improve planning and budgeting, a university data warehouse may be utilized to track performance patterns over time. Insights about the university's performance can also be gained, enabling better decision-making. Finally, the data warehouse may be utilized to pinpoint problem areas and create improvement plans.

The implementation of our university data repository requires careful planning and implementation. The first step is to identify the data sources that will be used to fill the data repository. This includes operational systems, external sources and other data sources. Once data sources are identified, all data must be tempered, collected and stored in the data repository.

The next step is to use the ETL tool (extract, conversion, load) to transfer data from one system to another, usually for the purpose of storing data. To be cleaned, validated and consolidated before loading into the data repository. This ensures that the data are accurate and up-to-date, and can be used for reporting and analysis.

Finally, the data warehouse design. This includes the design of the data repository structure, as well as data models and data markers. Once the data repository is designed, it must be provided with data from specific data sources. Finally, the data repository must be kept and monitored for accuracy and performance.

V.Conclusion

A university data warehouse is a powerful tool for reporting, analysis, and decision-making. It provides a single source of truth for the university, allowing for better data governance and improved accuracy. It can also be used to track performance trends over time, allowing for more effective planning and budgeting. Implementing a university data warehouse requires careful planning and execution.

In conclusion, a university data warehouse is an invaluable tool for reporting, analysis, and decision-making. It provides a single source of truth for the university, allowing for better data governance and improved accuracy. It can also be used to track performance trends over time, allowing for more effective planning and budgeting. Implementing a university data warehouse requires careful planning and execution.

Chapter 2

Steps to build: Data warehouses

1. INTRODUCTION

The data warehouse will be used to find important patterns and create precise forecasting models. It's crucial to comprehend the following procedures and carry out each correctly while creating a data warehouse.

2. Steps for creating data warehouses

2.1. Determine the Business Objectives of a University's Data

Using the university data repository, keep track of student enrollment and performance, faculty accomplishments; It may also be used to provide reports and insights about the university's activity, enabling more informed decisions. The data repository also makes it possible to track performance trends over time, making planning more effective.

2.2. Configuring Physical Environments

To achieve the greatest results from the data warehouse, it is first necessary to clean the database. In our situation, we shall arrange a disorganized university database.

a. What is Data Cleaning?

Data cleaning is the process of preparing data for analysis. It involves identifying and correcting errors, ensuring data is complete and consistent, and transforming data into a format that is appropriate for analysis.

b. Tools for Data Cleaning

Data cleaning tools are designed to help automate the process of data cleaning. These tools can help to identify errors, fill in missing data, and transform data into the right format for analysis.

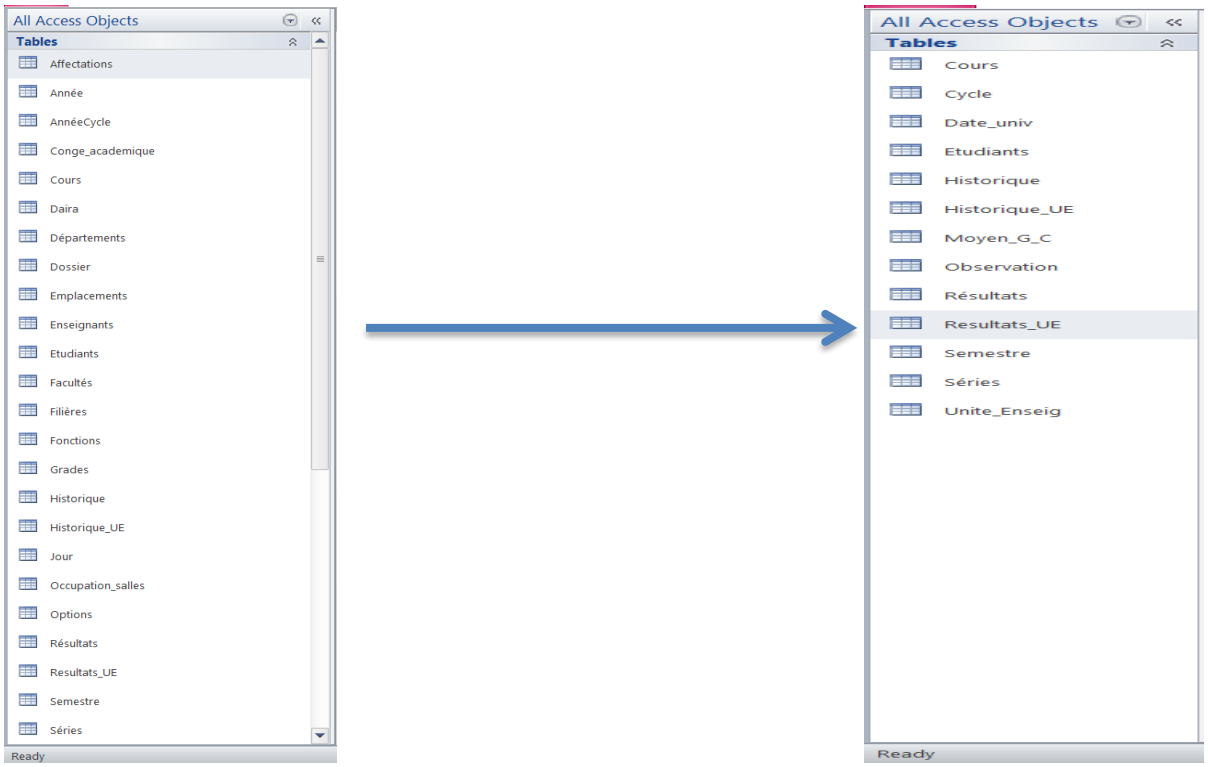
Data cleaning tools can be used to streamline the process of data cleaning and make it easier to analyze data. They can also help to reduce the amount of time and effort required for data cleaning.

c. Data cleaning steps

Our data cleaning process involves several steps. These steps include identifying errors, filling in missing data for example:

1- The first step is to delete the tables that we do not need:

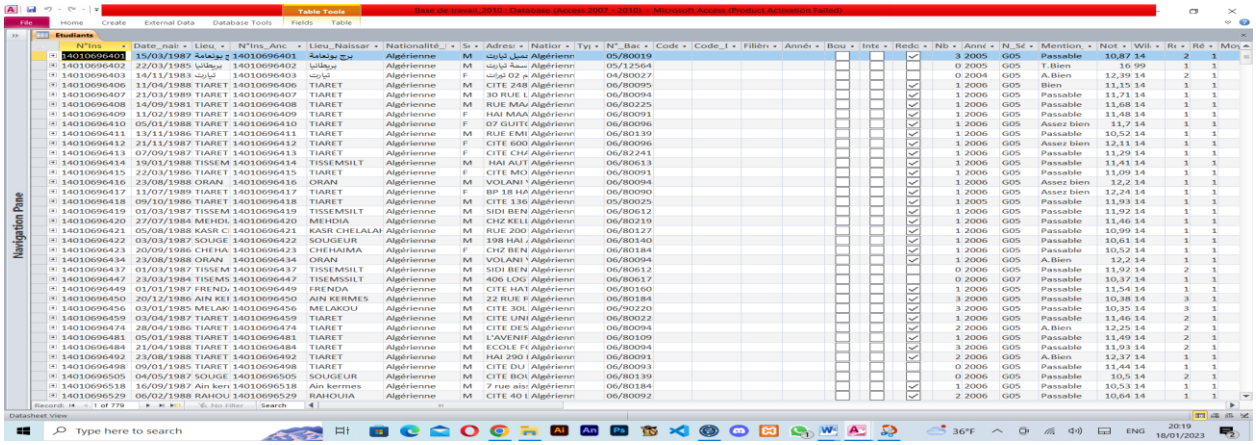
Affectations - Année - Année cycle - Conge académique - Daïra - Enplacements - Enseignants - Facultés - Filières - Fonctions - Grades - Jour - Occupation salles - Options - Statistiques - Table des erreurs - Tables de travaille - Tables emploi du temps - Table Semestre - Type Enseignant - Types inscription - Universités - Willaya



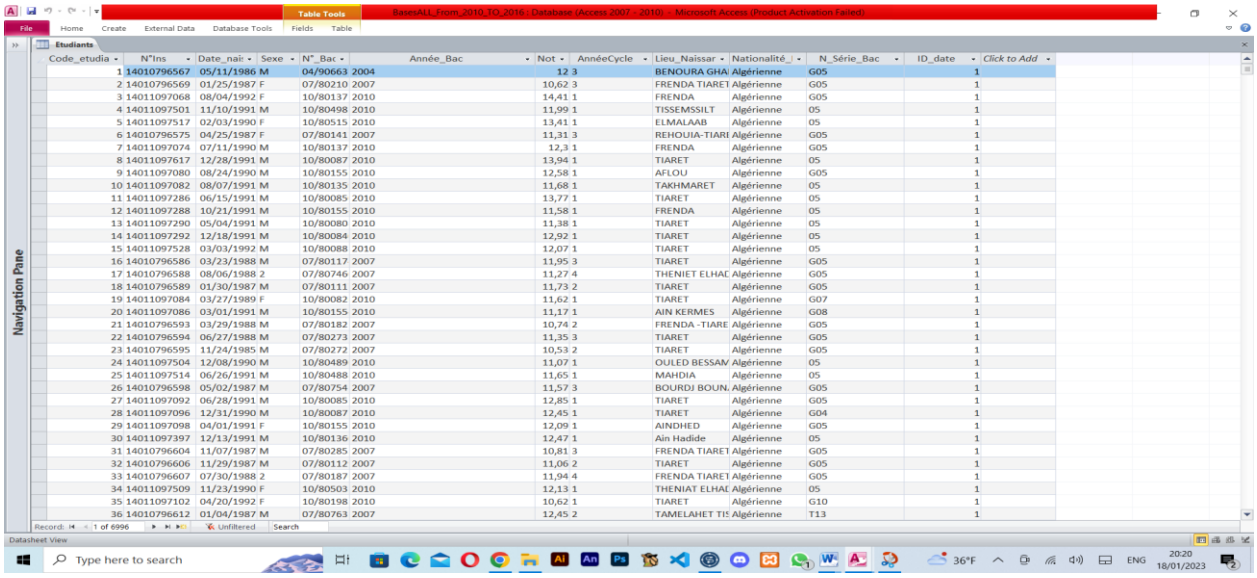
BEFORE **AFTER**

Figure 5 before and after delete the tables

2- Delete duplicate columns from the information and also delete columns that are not filled and columns that we do not need:



BEFORE



AFTER

Figure 6 before and after Delete duplicate columns

3- Delete some of the students who made a mistake in FilièreEtude Because 01 is not there Code_fillière:

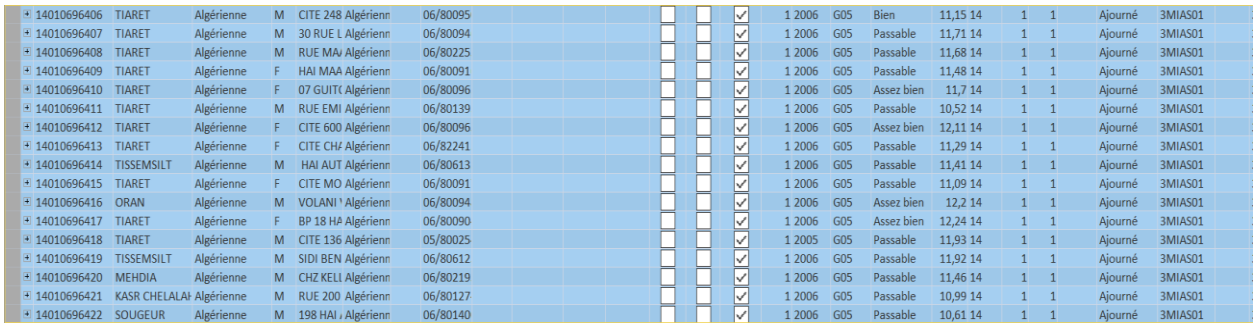


Figure 7 an example of a student you have to eliminate

4- Some students deleted a different year from the database example:

Like databases en 2010

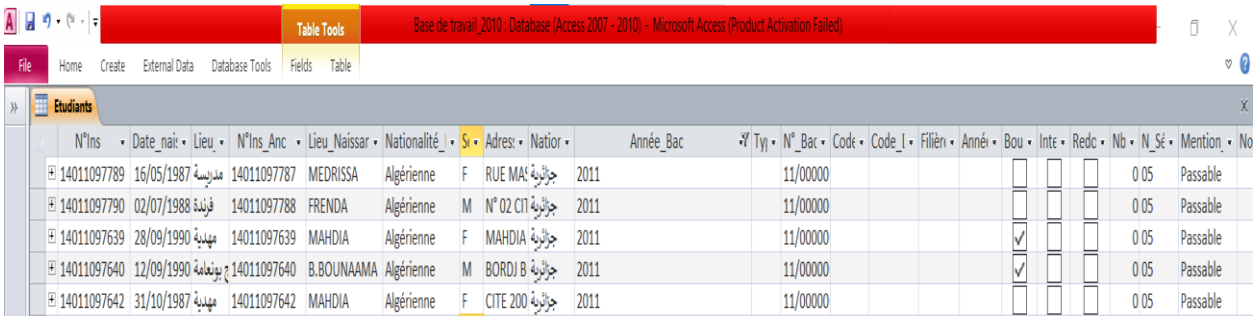
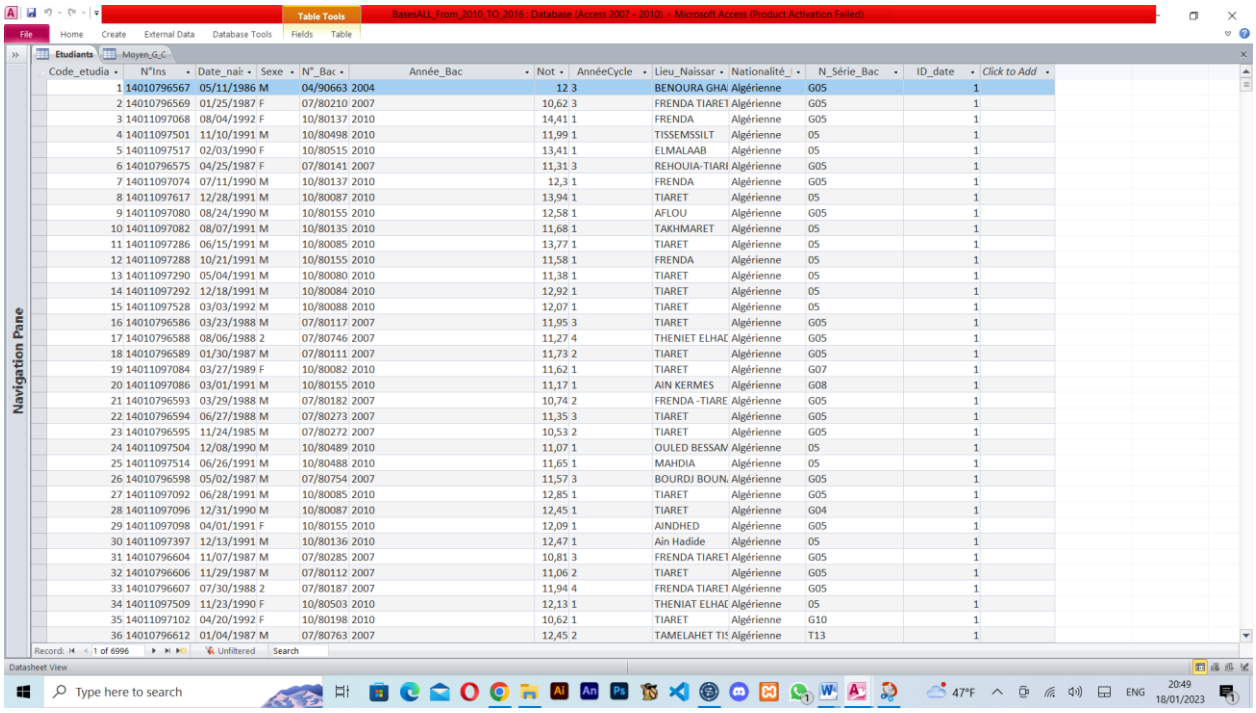
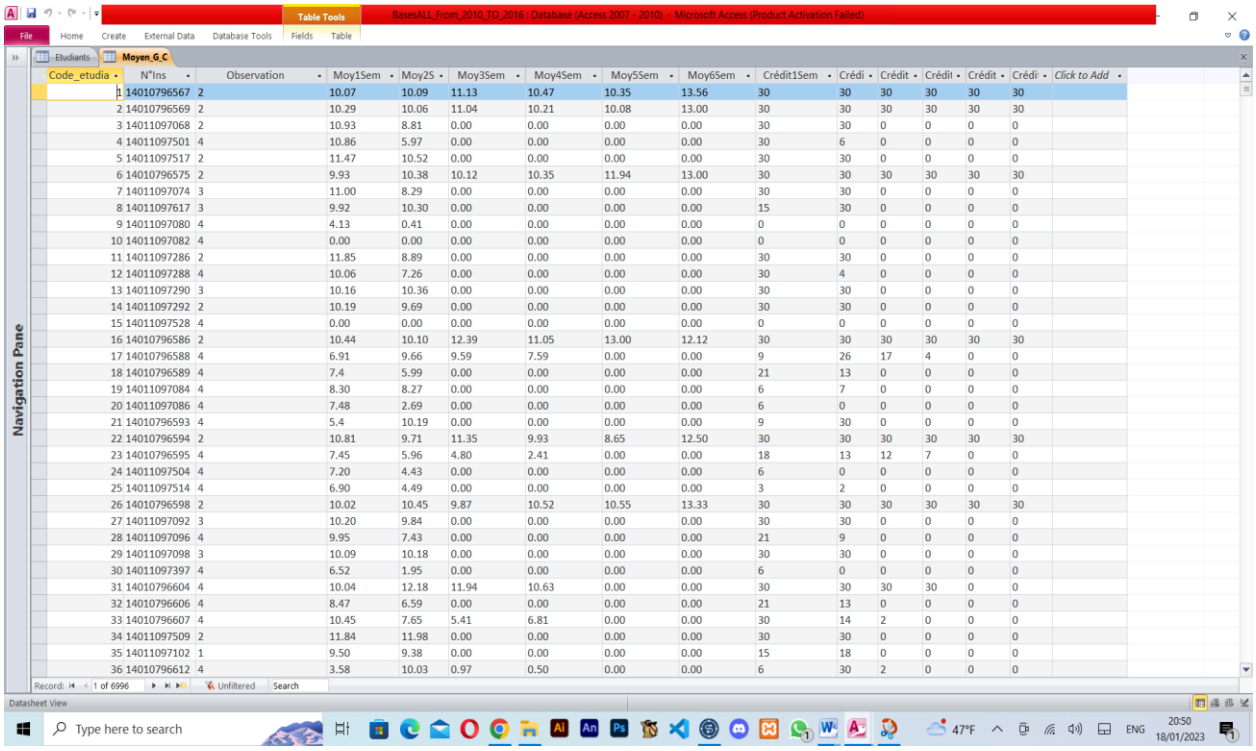


Figure 8 an example of a student have different information

5- dividing some tables into small tables (reparting):

An example of dividing the students table into two parts, the first table is student information, the second table is moyen_gc:



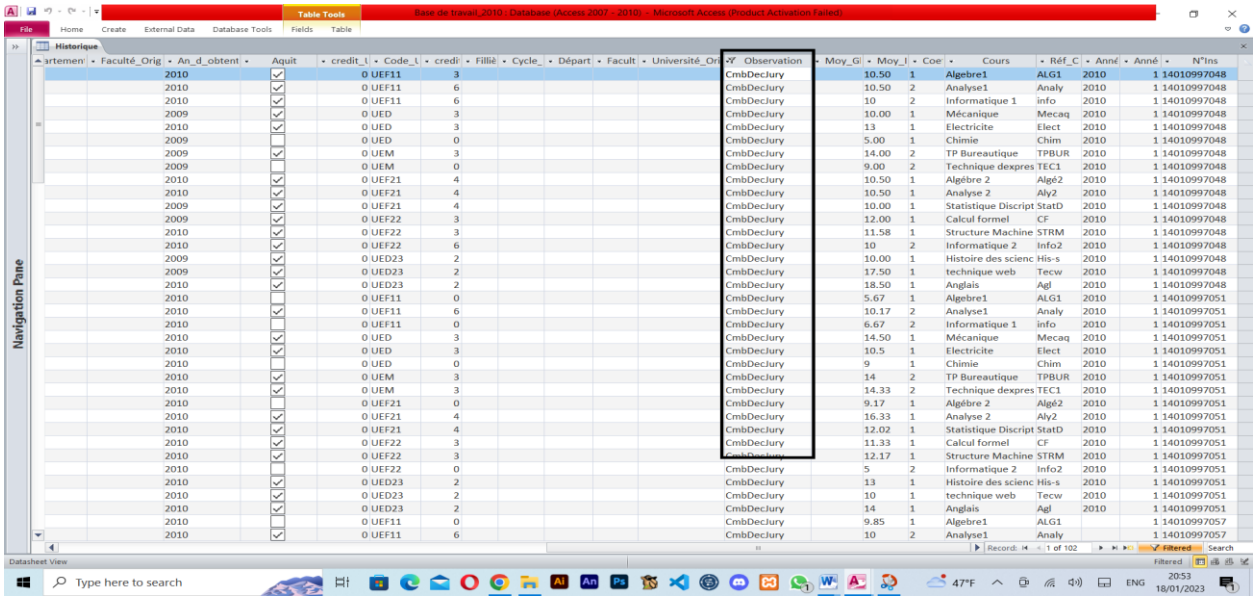


Code_etudia	N°Ins	Observation	Moy1Sem	Moy2S	Moy3Sem	Moy4Sem	Moy5Sem	Moy6Sem	Crédit1Sem	Crédi	Crédit	Crédit	Crédit	Crédi	Click to Add
1	14010796567	2	10.07	10.09	11.13	10.47	10.35	13.56	30	30	30	30	30	30	
2	14010796569	2	10.29	10.06	11.04	10.21	10.08	13.00	30	30	30	30	30	30	
3	14011097068	2	10.93	8.81	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
4	14011097501	4	10.86	5.97	0.00	0.00	0.00	0.00	30	6	0	0	0	0	
5	14011097517	2	11.47	10.52	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
6	14010796575	2	9.93	10.38	10.12	10.35	11.94	13.00	30	30	30	30	30	30	
7	14011097074	3	11.00	8.29	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
8	14011097617	3	9.92	10.30	0.00	0.00	0.00	0.00	15	30	0	0	0	0	
9	14011097080	4	4.13	0.41	0.00	0.00	0.00	0.00	0	0	0	0	0	0	
10	14011097082	4	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	
11	14011097286	2	11.85	8.89	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
12	14011097288	4	10.06	7.26	0.00	0.00	0.00	0.00	30	4	0	0	0	0	
13	14011097290	3	10.16	10.36	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
14	14011097292	2	10.19	9.69	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
15	14011097528	4	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	
16	14010796586	2	10.44	10.10	12.39	11.05	13.00	12.12	30	30	30	30	30	30	
17	14010796588	4	6.91	9.66	9.59	7.59	0.00	0.00	9	26	17	4	0	0	
18	14010796589	4	7.4	5.99	0.00	0.00	0.00	0.00	21	13	0	0	0	0	
19	14011097084	4	8.30	8.27	0.00	0.00	0.00	0.00	6	7	0	0	0	0	
20	14011097086	4	7.48	2.69	0.00	0.00	0.00	0.00	6	0	0	0	0	0	
21	14010796593	4	5.4	10.19	0.00	0.00	0.00	0.00	9	30	0	0	0	0	
22	14010796594	2	10.81	9.71	11.35	9.93	8.65	12.50	30	30	30	30	30	30	
23	14010796595	4	7.45	5.96	4.80	2.41	0.00	0.00	18	13	12	7	0	0	
24	14011097504	4	7.20	4.43	0.00	0.00	0.00	0.00	6	0	0	0	0	0	
25	14011097514	4	6.90	4.49	0.00	0.00	0.00	0.00	3	2	0	0	0	0	
26	14010796598	2	10.02	10.45	9.87	10.52	10.55	13.33	30	30	30	30	30	30	
27	14011097092	3	10.20	9.84	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
28	14011097096	4	9.95	7.43	0.00	0.00	0.00	0.00	21	9	0	0	0	0	
29	14011097098	3	10.09	10.18	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
30	14011097397	4	6.52	1.95	0.00	0.00	0.00	0.00	6	0	0	0	0	0	
31	14010796604	4	10.04	12.18	11.94	10.63	0.00	0.00	30	30	30	30	0	0	
32	14010796606	4	8.47	6.59	0.00	0.00	0.00	0.00	21	13	0	0	0	0	
33	14010796607	4	10.45	7.65	5.41	6.81	0.00	0.00	30	14	2	0	0	0	
34	14011097509	2	11.84	11.98	0.00	0.00	0.00	0.00	30	30	0	0	0	0	
35	14011097102	1	9.50	9.38	0.00	0.00	0.00	0.00	15	18	0	0	0	0	
36	14010796612	4	3.58	10.03	0.97	0.50	0.00	0.00	6	30	2	0	0	0	

Figure 9 an example of breaking certain tables into smaller tables

6- Correcting some errors which include etudiant example:

-Observation: CmbDecJury → Admis/Dettes



Faculté_Orig	An_d_obtent	Aquit	credit_l	Code_l	credi	Fillè	Cycle_	Départ	Facult	Université_Ori	Observation	Moy_Gi	Moy_I	Coe	Cours	Réf_C	Anné	Anné	N°Ins
	2010	✓	0	UEF11	3						CmbDecJury		10.50	1	Algebre1	ALG1	2010	1	14010997048
	2010	✓	0	UEF11	6						CmbDecJury		10.50	2	Analyse1	Analy	2010	1	14010997048
	2010	✓	0	UEF11	6						CmbDecJury		10	2	Informatique 1	info	2010	1	14010997048
	2009	✓	0	UED	3						CmbDecJury		10.00	1	Mécanique	Mecaq	2010	1	14010997048
	2010	✓	0	UED	3						CmbDecJury		13	1	Electricite	Elect	2010	1	14010997048
	2009		0	UED	0						CmbDecJury		5.00	1	Chimie	Chim	2010	1	14010997048
	2009	✓	0	UEM	3						CmbDecJury		14.00	2	TP Bureautique	TPBUR	2010	1	14010997048
	2009		0	UEM	0						CmbDecJury		9.00	2	Technique dexpres	TEC1	2010	1	14010997048
	2010	✓	0	UEF21	4						CmbDecJury		10.50	1	Algébre 2	Algé2	2010	1	14010997048
	2010	✓	0	UEF21	4						CmbDecJury		10.50	1	Analyse 2	Aly2	2010	1	14010997048
	2009	✓	0	UEF21	4						CmbDecJury		10.00	1	Statistique Discript	StatD	2010	1	14010997048
	2009	✓	0	UEF22	3						CmbDecJury		12.00	1	Calcul formel	CF	2010	1	14010997048
	2010	✓	0	UEF22	3						CmbDecJury		11.58	1	Structure Machine	STRM	2010	1	14010997048
	2010	✓	0	UEF22	6						CmbDecJury		10	2	Informatique 2	Info2	2010	1	14010997048
	2009	✓	0	UED23	2						CmbDecJury		10.00	1	Histoire des scienc	His-s	2010	1	14010997048
	2009	✓	0	UED23	2						CmbDecJury		17.50	1	technique web	Tecw	2010	1	14010997048
	2010	✓	0	UED23	2						CmbDecJury		18.50	1	Anglais	Agl	2010	1	14010997048
	2010		0	UEF11	0						CmbDecJury		5.67	1	Algebre1	ALG1	2010	1	14010997051
	2010	✓	0	UEF11	6						CmbDecJury		10.17	2	Analyse1	Analy	2010	1	14010997051
	2010		0	UEF11	0						CmbDecJury		6.67	2	Informatique 1	info	2010	1	14010997051
	2010	✓	0	UED	3						CmbDecJury		14.50	1	Mécanique	Mecaq	2010	1	14010997051
	2010	✓	0	UED	3						CmbDecJury		10.5	1	Electricite	Elect	2010	1	14010997051
	2010		0	UED	0						CmbDecJury		9	1	Chimie	Chim	2010	1	14010997051
	2010	✓	0	UEM	3						CmbDecJury		14	2	TP Bureautique	TPBUR	2010	1	14010997051
	2010	✓	0	UEM	3						CmbDecJury		14.33	2	Technique dexpres	TEC1	2010	1	14010997051
	2010		0	UEF21	0						CmbDecJury		9.17	1	Algébre 2	Algé2	2010	1	14010997051
	2010	✓	0	UEF21	4						CmbDecJury		16.33	1	Analyse 2	Aly2	2010	1	14010997051
	2010	✓	0	UEF21	4						CmbDecJury		12.02	1	Statistique Discript	StatD	2010	1	14010997051
	2010	✓	0	UEF22	3						CmbDecJury		11.33	1	Calcul formel	CF	2010	1	14010997051
	2010	✓	0	UEF22	3						CmbDecJury		12.17	1	Structure Machine	STRM	2010	1	14010997051
	2010		0	UEF22	0						CmbDecJury		5	2	Informatique 2	Info2	2010	1	14010997051
	2010	✓	0	UED23	2						CmbDecJury		13	1	Histoire des scienc	His-s	2010	1	14010997051
	2010	✓	0	UED23	2						CmbDecJury		10	1	technique web	Tecw	2010	1	14010997051
	2010	✓	0	UED23	2						CmbDecJury		14	1	Anglais	Agl	2010	1	14010997051
	2010		0	UEF11	0						CmbDecJury		9.85	1	Algebre1	ALG1		1	14010997057
	2010	✓	0	UEF11	6						CmbDecJury		10	2	Analyse1	Analy		1	14010997057

Figure 10 an example of fixing certain table errors

- AnnéeUniv: There is an error 1ere→2010,

2eme→2009 We change it to 1ere→2008, 2eme→2009

C	AnnéeU	Année	N°Ins
	2009	1	14010997053
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2010	1	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056
	2009	2	14010997056

AnnéeUniv	Année	N°Ins
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2008	1	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056
2009	2	14010997056

Figure 11 an example of fixing certain table errors 2

7- We have added some tables to facilitate data extraction and the goal of building the data warehouse.

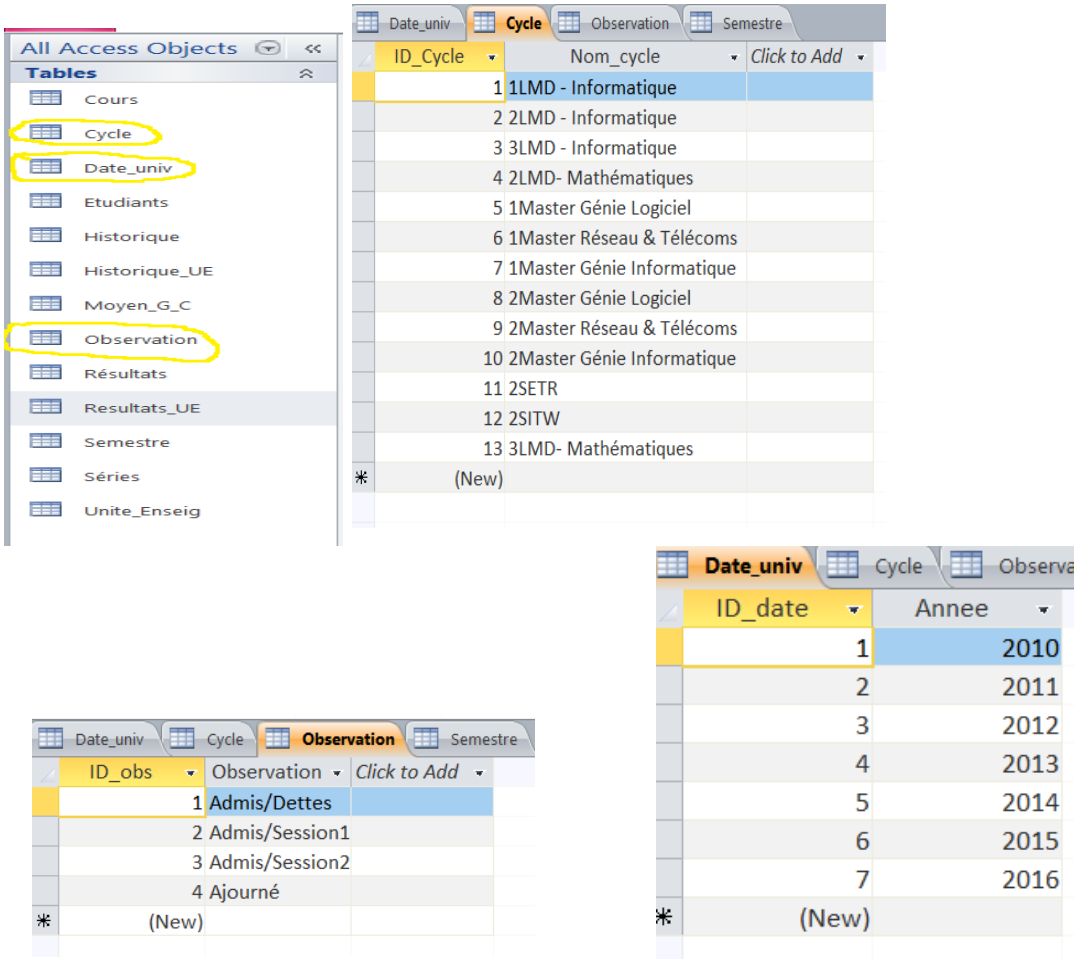


Figure 12 some table add to building the data warehouse

8- After updating all the database every year (2010…2016), we collect it in database with the addition of a column Code_etudiant due to repeat N°Ins

Finally

Data cleansing is a crucial component of data analysis. It can assist to shorten the time and effort needed for data processing while ensuring accuracy and consistency.

Data cleaning is a multi-step process that includes finding mistakes, adding missing data, and converting data to the proper format for analysis. Data analysts may make sure that their data is correct and trustworthy by taking the effort to clean it.

2.3. Introducing Data Modeling

At the data store level, data is modeled before branching out into the data warehouse. The Snowflake, Star, and Galaxy charts are the three most used data

models for warehouses. The structure of your data warehouse, the data markers it contains, and its general design will all depend on the model you choose.

2.3.1. Dimensional Modeling

Dimensional Modeling (DM) is a data structure approach suited for data warehouse storage. The goal of dimensional modeling is to improve the database for quicker data retrieval. Ralph Kimball created the Dimensional Modeling idea, which comprises of "fact" and "dimension" tables.

2.3.2. Elements of Dimensional Data Model

2.3.2.1. Fact

Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number

2.3.2.2. Dimension

Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be

- Who – Customer Names
- Where – Location
- What – Product Name

In other words, a dimension is a window to view information in the facts.

2.3.2.3. Attributes

The Attributes are the various characteristics of the dimension in dimensional data modeling.

In the Location dimension, the attributes can be

- State
- Country
- Zipcode etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

2.3.2.4. Fact Table

A fact table is a primary table in dimension modelling.

A Fact Table contains

1. Measurements/facts
2. Foreign key to dimension table

2.3.2.5. Dimension Table

- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes
- No set limit set for given for number of dimensions
- The dimension can also contain one or more hierarchical relationships

We developed this modeling by expanding and manipulating the data Using Star schema

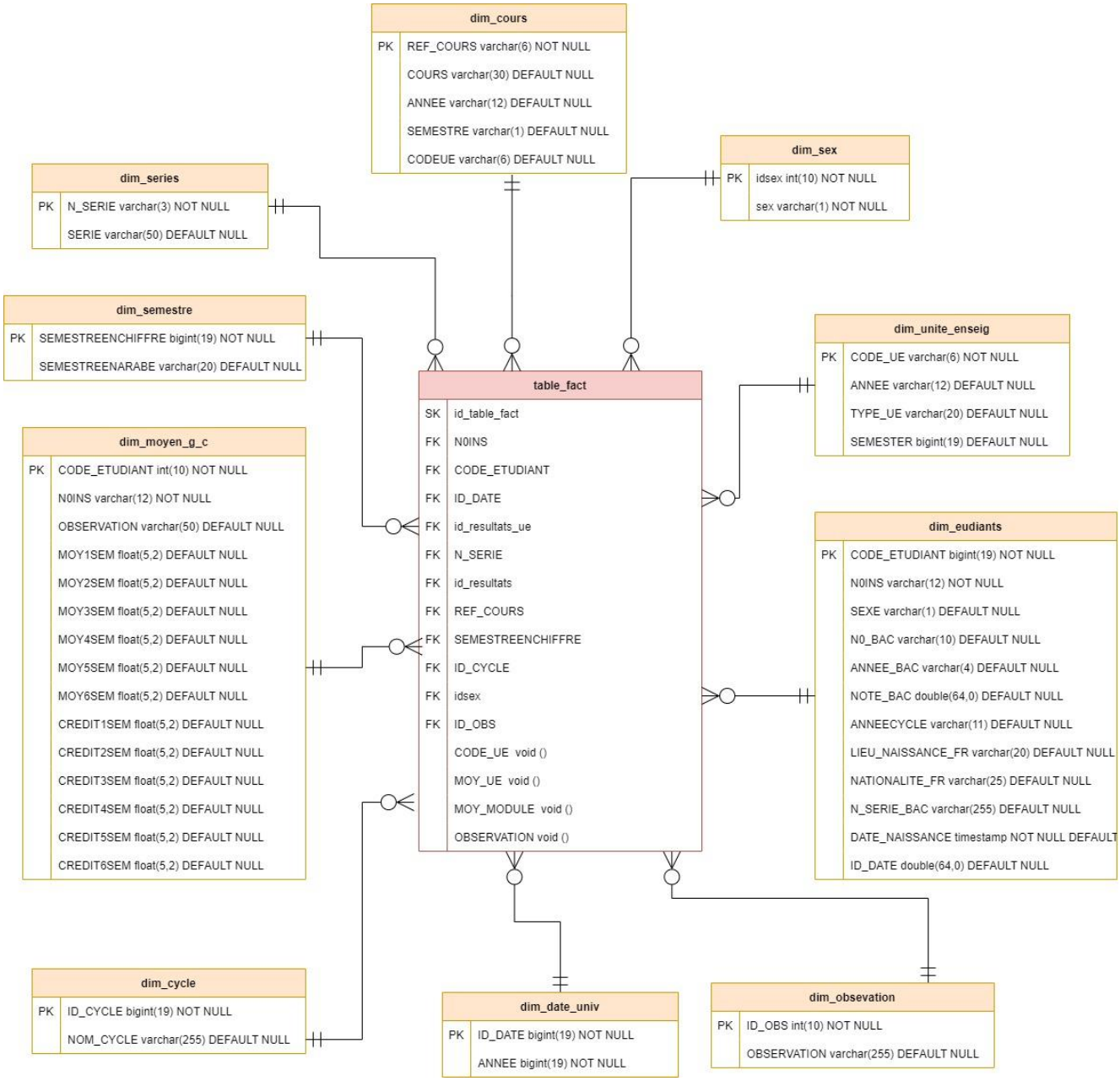


Figure 13 University Data Model Star Schema

2.4. Choosing ETL Solution

ETL stands for extraction, transformation, and loading and refers to the collecting and processing of data from many sources into a centralized data storage where it may be evaluated later. Access to several data sources simplifies your company.

And we chose Talend ETL as our tools for the job.

What is Talend ETL Tool?

The open source software platform Talend ETL Tool is used to organize, integrate, and convert data. For colleges, it is used to create data marts and warehouses. It makes it possible for institutions to access, modify, and transfer data from many sources and systems to a common data warehouse.

Data integration, data quality, data profiling, data virtualization, and data governance are just a few of the functions offered by Talend ETL Tool. Additionally, it includes a graphical user interface that streamlines and automates the data integration process. It makes it possible for institutions to establish data warehouses and data marts and manage and analyze data from many sources with ease.

a. Benefits of Talend ETL Tool

With the help of the Talend ETL Tool, institutions may access, convert, and transfer data from many sources and systems to a single data warehouse. Universities can handle and analyze data from many sources fast and simply since it streamlines and automates the process of data integration. Data integration, data quality, data profiling, data virtualization, and data governance are just a few of the many functions it provides.

Cost-effective and simple to use, Talend ETL Tool. All sizes of colleges can use it because it is an open source platform. Large datasets may be managed with it, and it is also very scalable. Universities may use it to rapidly and easily build data marts and warehouses, as well as to analyze and display data.

b. How to Use Talend ETL Tool

A user-friendly platform called Talend ETL Tool makes the process of data integration simpler and more automated. Universities may swiftly and easily transfer data from many sources and systems to a central data warehouse thanks

to this technology. Additionally, a wide range of functions are available, including data virtualization, data quality, data governance, and data integration. Universities must first register for an account and install the Talend ETL Tool program before using it. Universities can start building data warehouses and data marts after installing the program. They may access, modify, and transfer data from many sources and systems to a central data warehouse using the program as well. Last but not least, the program may be used by institutions to analyze and visualize data.

1. How to create a job:

The step to create a job in the repository section at job design with the right click on it and create the job and name it and press finished.

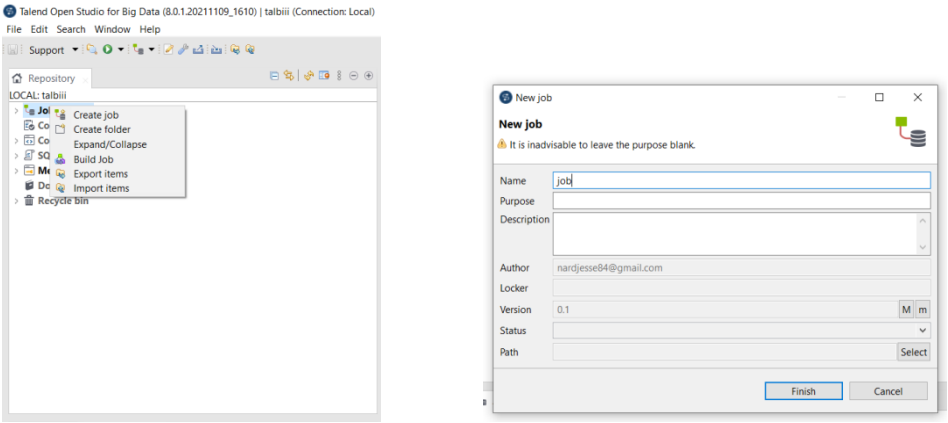


Figure 14 the step of creating a job in the repository

2. How to make connection to access and MySQL database:

To make a connection to the database you need to go to reposition section in the metadata and open it, DB connection with the right click of the mouse will show you a windows pressure to create a connection will show another window you will name your connection and pressure to next.

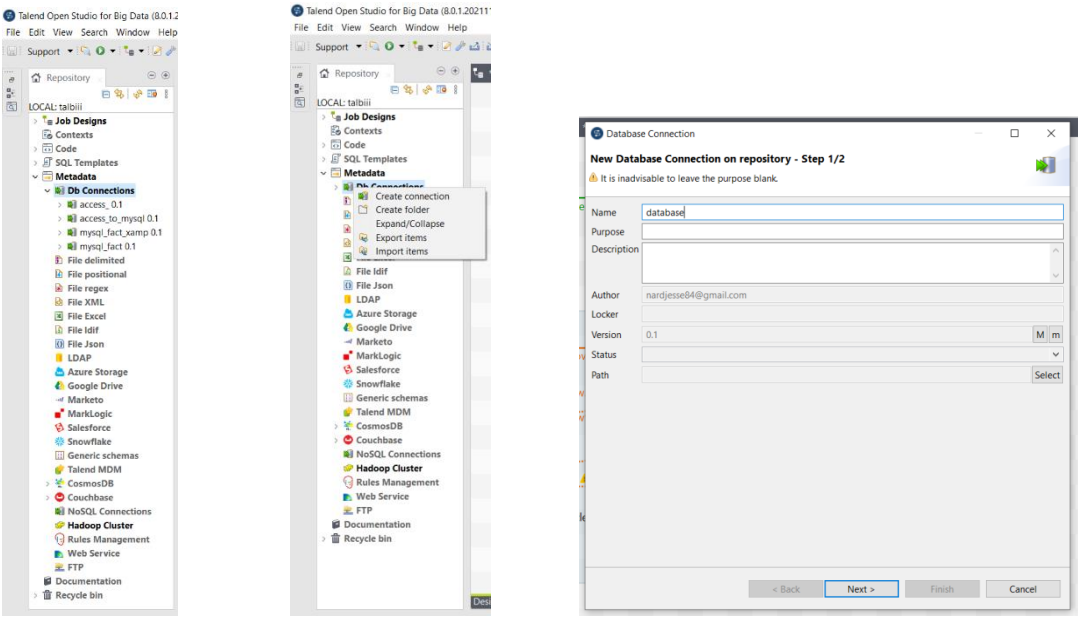


Figure 15 The database connection in Talend

The next step is to add information of the connection.

c. How to add access connection:

1- In the DB type choose access database.

2- In the additional parameter delete:

"jackcessOpener=org.talend.ucanaccess.encrypt.CryptCodecOpener;"

3- Press at browser and search of the access database.

4- Press “Test Connection” and finish.

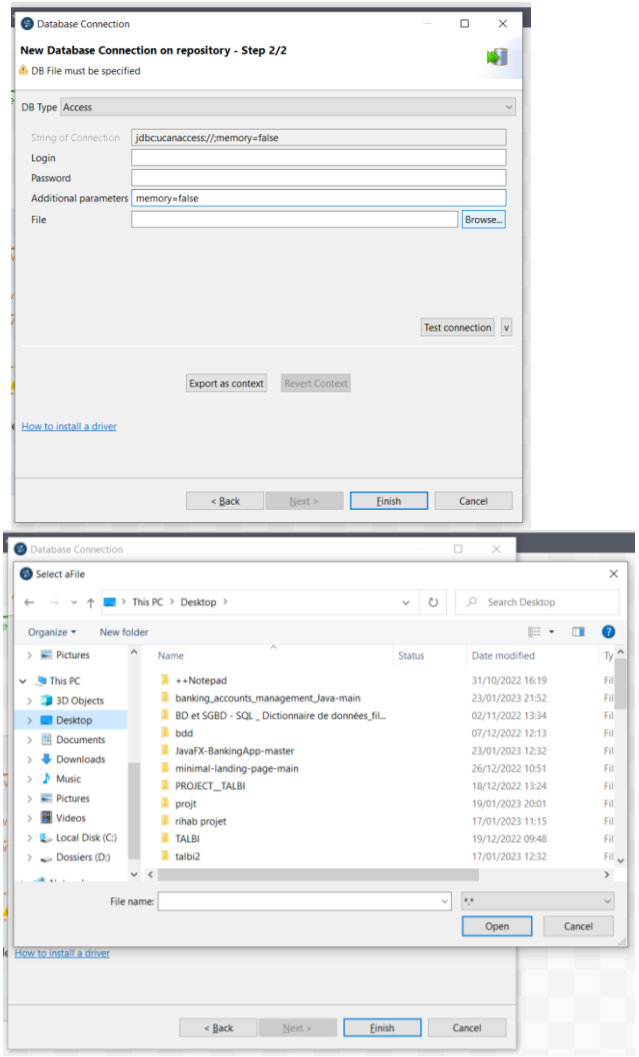


Figure 16 Access connection in Talend

d. How to add MySQL connection:

1- In the DB type choose MySQL database.

2- In the DB version choose the version of MySQL you have. - Better to work with MySQL 5 –

3- Put the name and the password and the server and the port and the database name.

4- In the additional parameter add this statement: " autoReconnect=true&useSSL=false".

5- Press “Test connection” and finish.

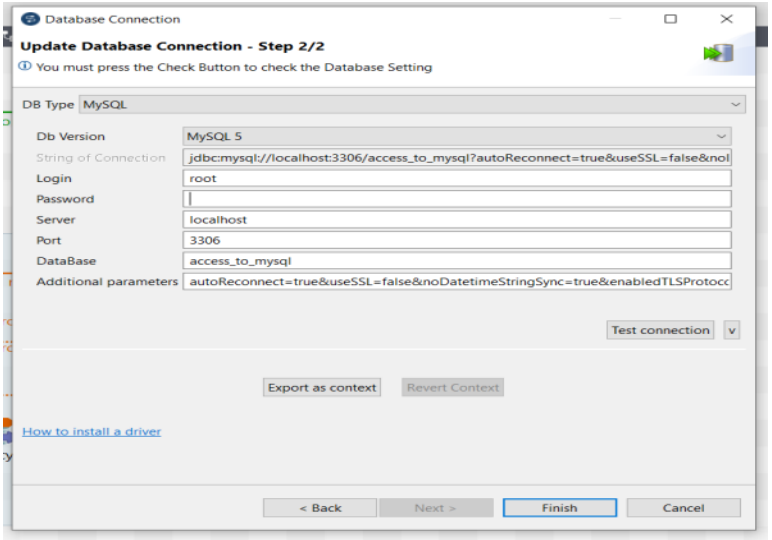


Figure 17 MySQL connection in Talend

e. How to create a component:

The first way is by searching for your element in the Platte section

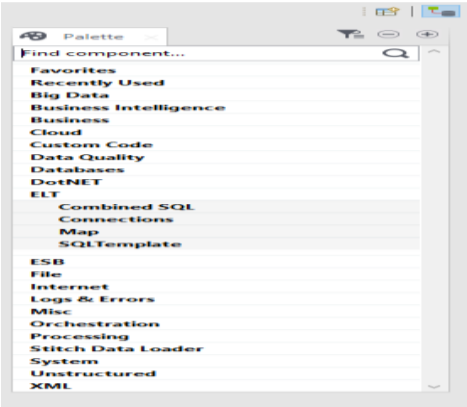


Figure 18 Platte section to develop a Talend component

The second way is by clicking on any place in the workplace and writing the name of the component

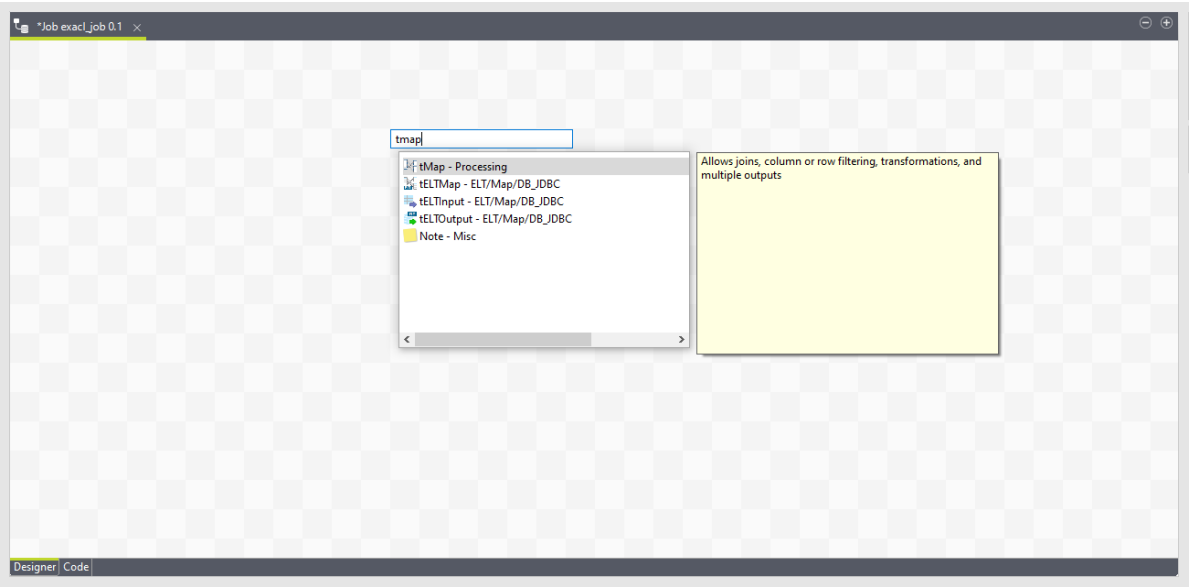


Figure 19 The second way to develop a Talend component

f. Components are used:

The tPrejob and tPostjob:

The tPrejob and tPostjob components are designed to make the execution of tasks before and after a given Job easier to manage. These components differ from other components in that they do not actually process data and they do not have any components properties to be configured. A key feature of these components is that they are always guaranteed to be executed, even if the main data Job fails. Therefore, they are very useful for setup and teardown actions for a given Job.

To use these tPrejob and tPostjob components, simply drop them onto the design workspace as you would do with any other components, and then connect tPrejob to a component or subJob that is meant to perform a pre-job task, and tPostjob to a component or subJob that is meant to perform a post-job task, using Trigger connections. An orange square on the pre- and post-job parts indicates that they are different types of subJobs.

Figure 20 The tPrejob and tPostjob Components

tDBConnection

Opens a connection to a database to be reused in the subsequent subJob or subJobs.

This component works with a variety of databases depending on your selection.

tDBConnection Standard properties

These properties are used to configure tDBConnection running in the Standard Job framework.

The Standard tDBConnection component belongs to the Databases family. The component in this framework is available in all Talend products.

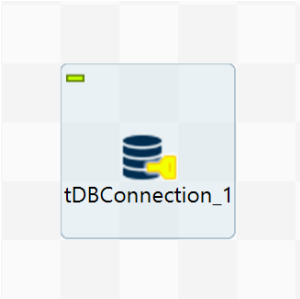


Figure 21 tDBConnection Components

tDBCommit

Validates the data processed through the Job into the connected database.

This component works with a variety of databases depending on your selection.

tDBCommit Standard properties

These properties are used to configure tDBCommit running in the Standard Job framework.

The Standard tDBCommit component belongs to the Databases family.

The component in this framework is available in all Talend products.

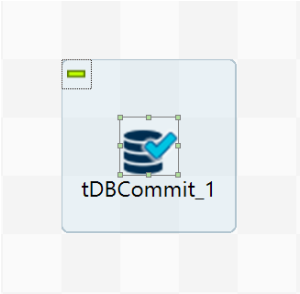


Figure 22 tDBCommit Components

tDBClose

Closes the transaction committed in a connected database.

This component works with a variety of databases depending on your selection.

tDBClose Standard properties

These properties are used to configure tDBClose running in the Standard Job framework.

The Standard tDBClose component belongs to the Databases family.

The component in this framework is available in all Talend products.

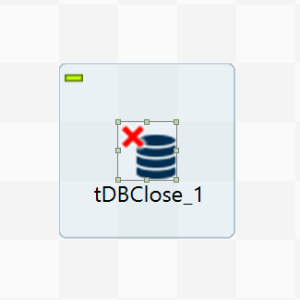


Figure 23 tDBClose Components

Extracts data from a database.

This component works with a variety of databases depending on your selection.

tDBInput Standard properties

These properties are used to configure tDBInput running in the Standard Job framework.

The Standard tDBInput component belongs to the Databases family.

The component in this framework is available in all Talend products.

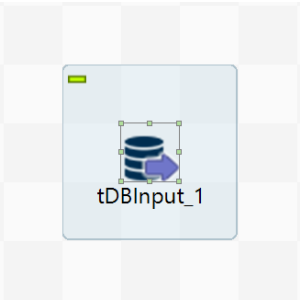


Figure 24 tDBInput Components

The tMap:

The tMap component is part of the Processing family of components. tMap is one of the core components and is primarily used for mapping input data to output data, that is, mapping one Schema to another.

As well as performing mapping functions, tMap may also be used to Join multiple inputs, and to write multiple outputs. Additionally, you can Filter data within the tMap component.

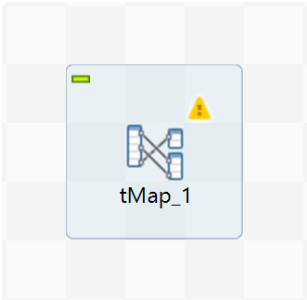


Figure 25 tMap component

tDBOutput

Writes, updates, makes changes or suppresses entries in a database.

This component works with a variety of databases depending on your selection.

tDBOutput Standard properties

These properties are used to configure tDBOutput running in the Standard Job framework.

The Standard tDBOutput component belongs to the Databases family.

The component in this framework is available in all Talend products.

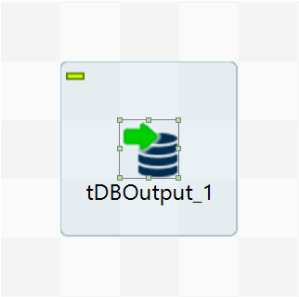


Figure 26 tDBOutput Components

What will we do in talend?

Note: *We add Prejob, Postjob, DBconnection,DBcommet, and DBclose for all the jobs.*

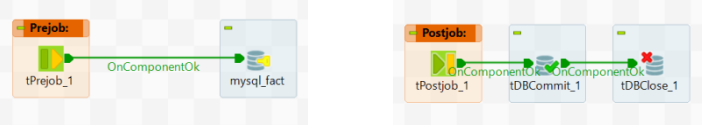


Figure 27 The crucial job that will be used in all projects

The first thing we do is create the first job under the name job is access to MySQL in this job we convert all databases in access to MySQL in this job we create an access input and connected to tMap and make the convert we need for the variable type and connect the tMap with output MySQL and ended with the postjob commit and close the database.

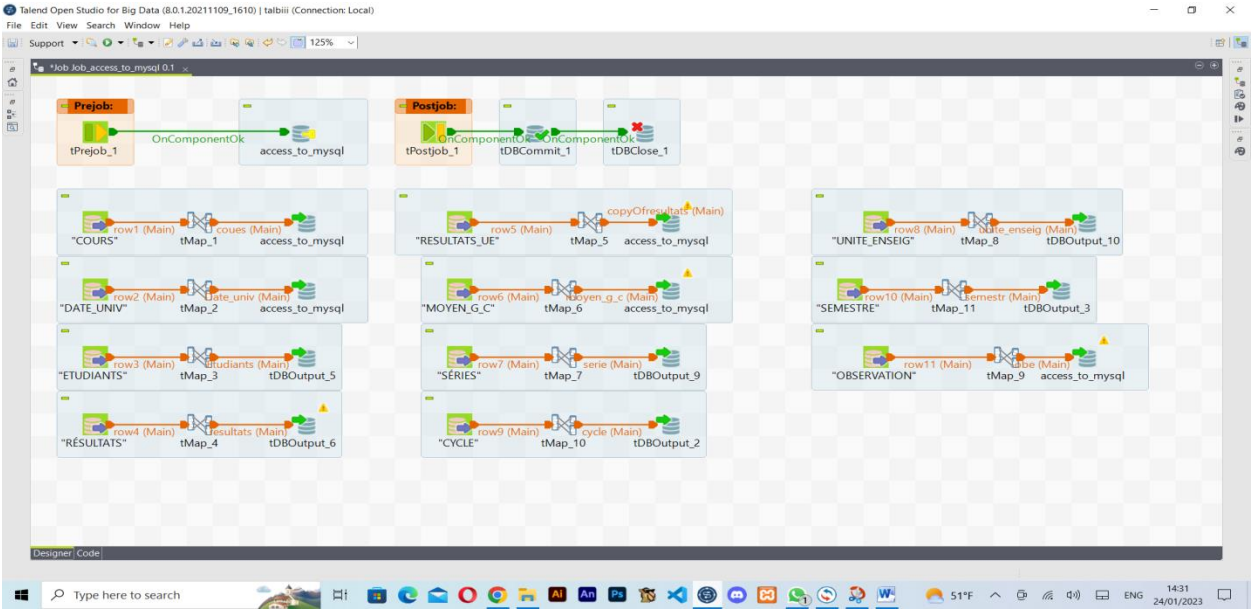


Figure 28 The first job container under the name job is access to MySQL.

For the second job under the name job fact we start with the Prejob and DB connection and make the tables dimension with the same step as the first job take input table from MySQL and to the tMap and the input DB.

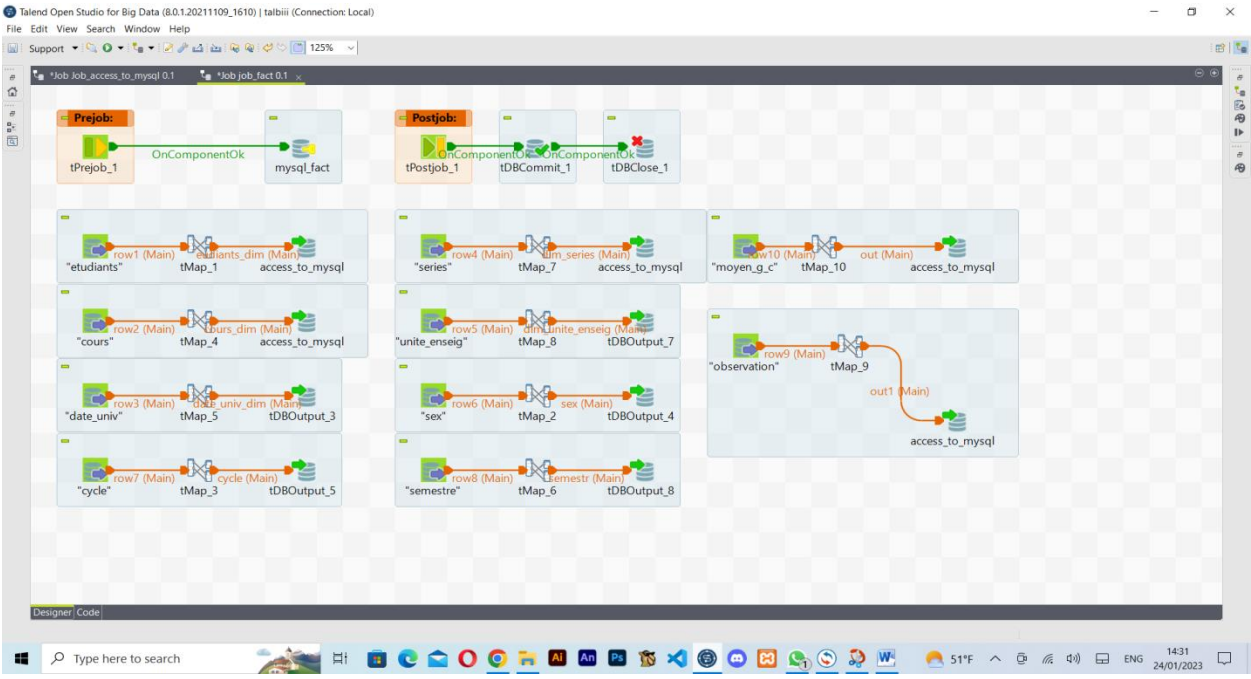


Figure 29 the second job under the name job fact

In the last job which was named job_olap we create our table fact and the schema, so we import all the tables we need, we start with prejob and the connection and the post job and the commit, and the close database, and import

all the table that exists in the schema and connect with the tmap and make the relation between the table and making the outbut table fact.

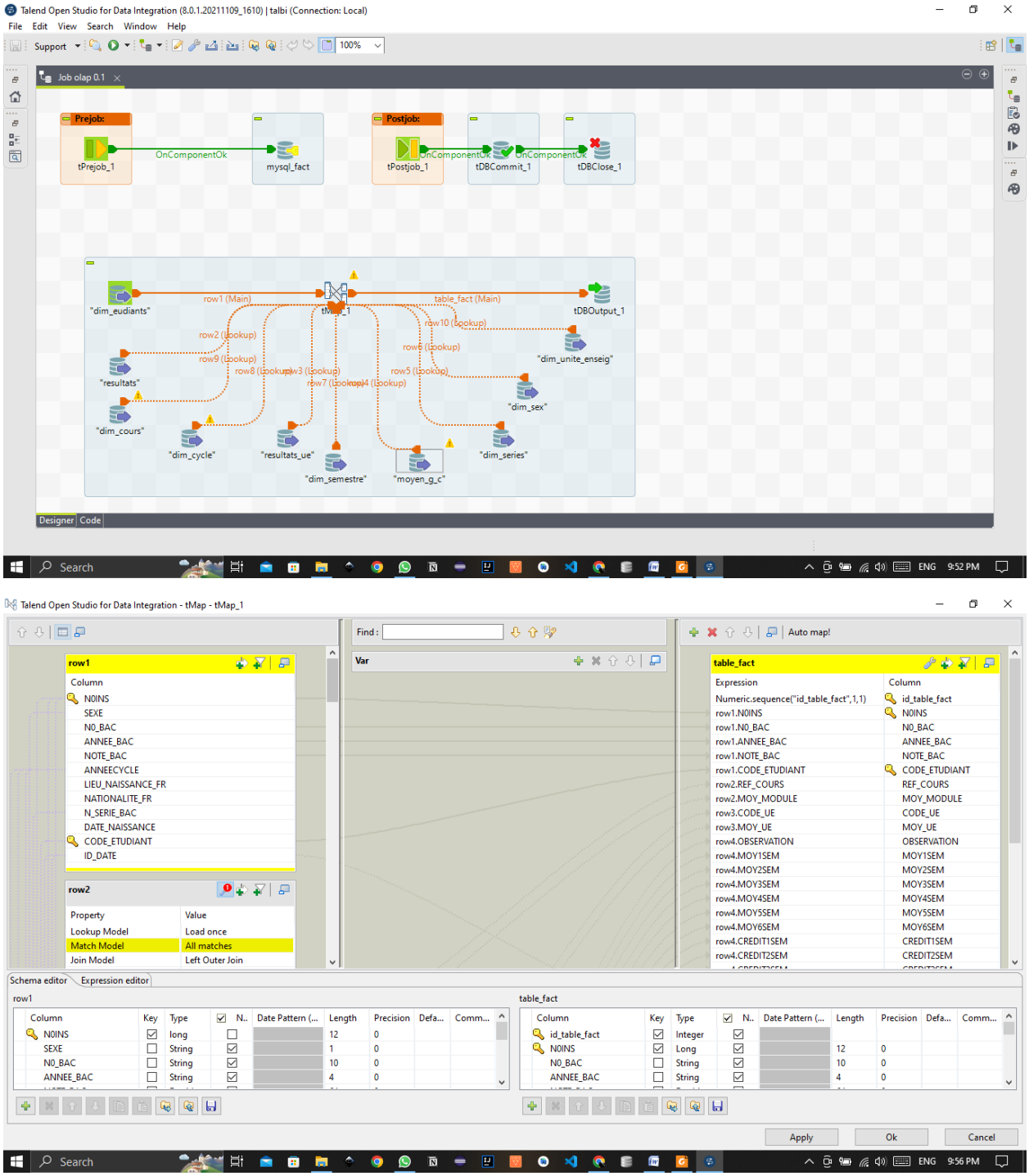


Figure 30 the previous job, job olap, uses tmap to establish the relationship between the tables

The problem:

- We changing the type of some variable.

–We have met some problem in the tmap we the all match and the all row and the relation between the table.

- The connection with the access and MySQL until delete or add some statement. –

Finally

The ETL assisted us in creating Data Modeling and embodiment from the previous steps "Introducing Data Modeling" and in combining information from diverse sources.

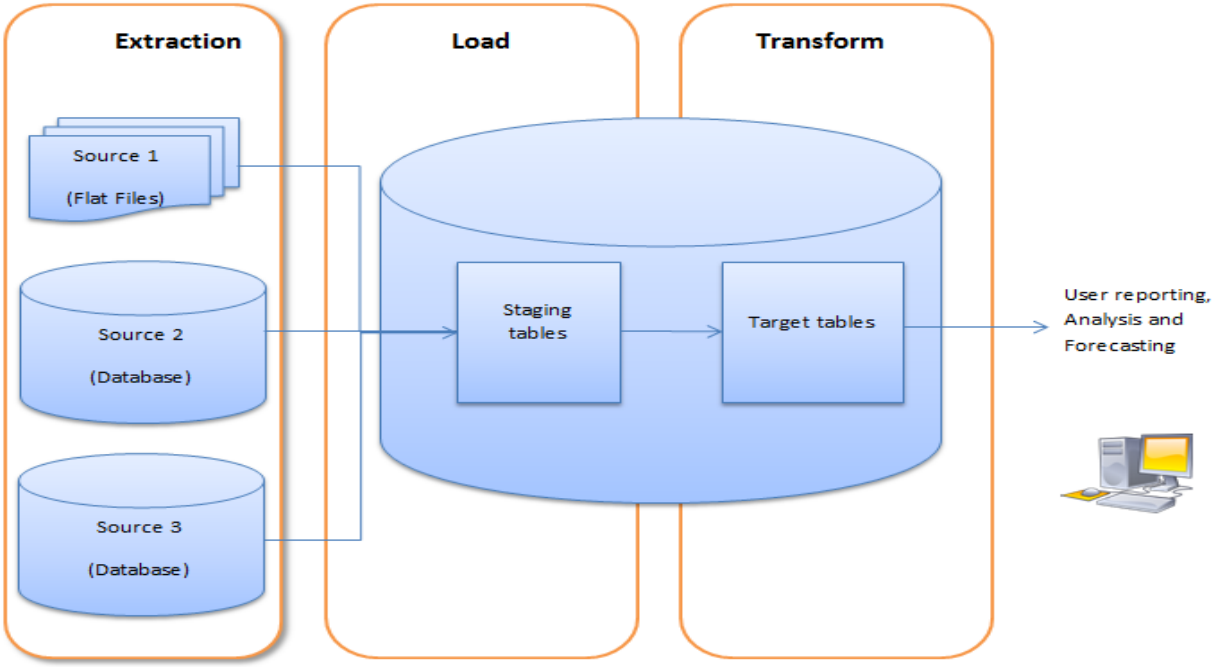


Figure 31 ETL based data warehouse architecture diagram

2.5. Online Analytics Processing (OLAP) Cube

The OLAP cube assists us in analyzing the data in our data repository or data store. Because our warehouse will be sorting data from many sources, the OLAP cube will assist us in organizing all of this data in a multidimensional style that is straightforward to examine.

We will utilize OLAP Schema Workbench in these steps.

2.5.1. What is OLAP Schema Workbench?

We used OLAP Scheme Workbench as a tool to create and maintain multidimensional schemes. allows users to design and change hierarchies, data cubes, and data structures. This tool makes it simple to locate dimensional connections and hierarchies, locate cube data sources, and save data in the database.

By showing a graphic depiction of the scheme, OLAP Scheme Workbench also aids users in visualizing data structures. This makes it simpler to comprehend the connections between various dimensions and hierarchies and to construct more intricate systems.

Uses of OLAP Schema Workbench

There are several uses for OLAP Schema Workbench. The data structures and connections between various dimensions may be shown, as well as how to design and maintain multidimensional schemas. Additionally, it may be used to save data in the database and attach data sources to cubes.

Users may avoid having to manually establish each relationship by using the tool to rapidly and efficiently create hierarchies and relationships between dimensions. Users may save time and effort as a result.

2.5.2. Make OLAP Schema

To develop a cube OLAP schema using Pentaho Schema Workbench, first launch the program and connect to your data source. Then, for the cube, establish a new OLAP cube project and set its dimensions, hierarchies, measurements, computed members, and aggregation levels. Before publishing the cube to the Pentaho Server, you may preview and test it using the Preview tab.

Following data uploading into the database, the first thing we should perform is run pentaho schema workbench:

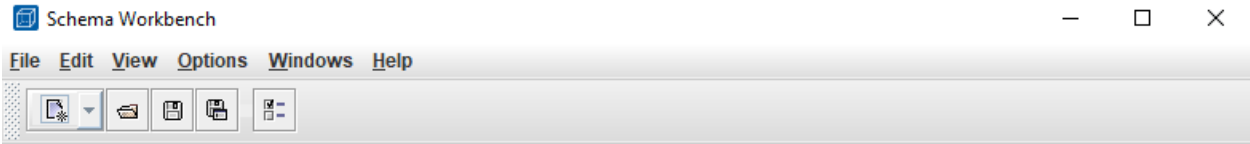


Figure 32 pentaho schema workbench interface

2.5.3. Add a Data Source

The data source must be accessible, its database driver JAR must be present in the /pentaho/design-tools/schema-workbench/drivers/ directory, and you must be aware of or have access to the database connection details and user account credentials for the data source.

To connect to a data source in Schema Workbench, follow the steps below.

1. Through the Options menu, choose Connection to connect to your data source. The dialog box for the database connection appears.

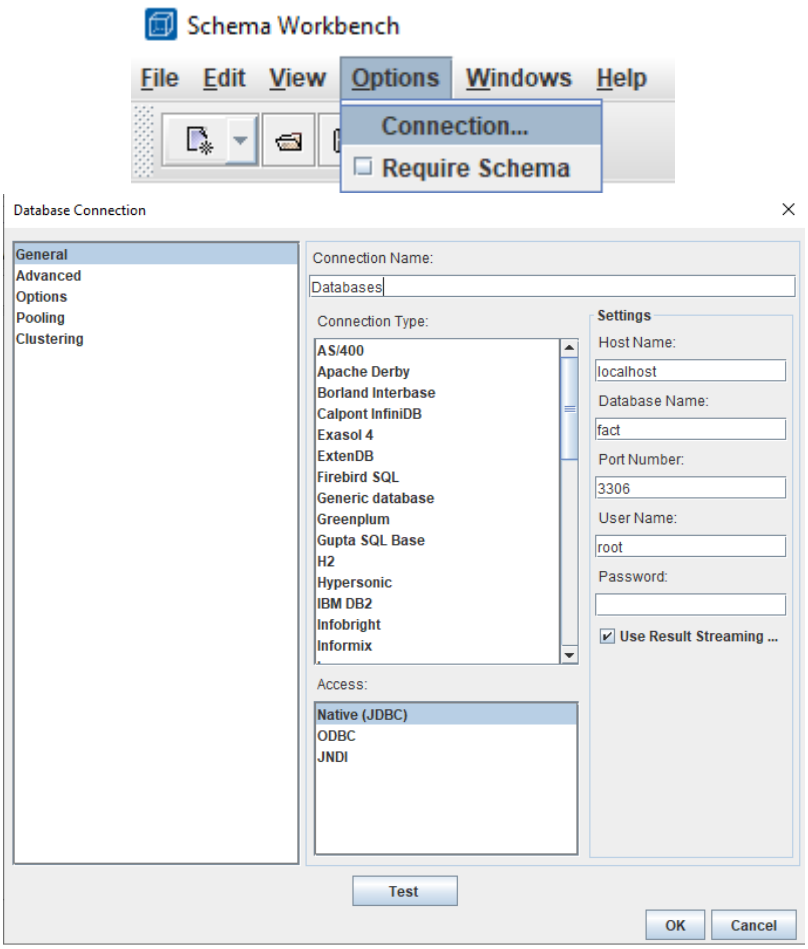


Figure 33 Add a Data Source using the Pentaho Schema Workbench Interface

2. Enter the appropriate database connection details after choosing your database type, then click Test. Click OK after making sure the connection settings function properly. The user credentials, port number, and database name are all included in the connection information. Consult the instructions provided by your database vendor or administrator if you are unsure of how to fill up any of these fields.

Note Schema Workbench enters a state where unpopulated items show up in the schema when the Require Schema check box in the Options menu is chosen.

Note Through selecting Require Schema, you may significantly reduce the load time for your Analysis schema if your data source is an Oracle one.

1.1.1. Create a Mondrian Schema

Before you start You should have already linked to your data source in Schema Workbench before beginning this procedure.

This section describes how to use Scheme Workbench to create a simple Mondrian scheme.

1. Click the New button or choose New from the File menu, then Schema to start a new Mondrian schema. There will be a brand-new schema sub-window. You may resize it to suit your tastes.

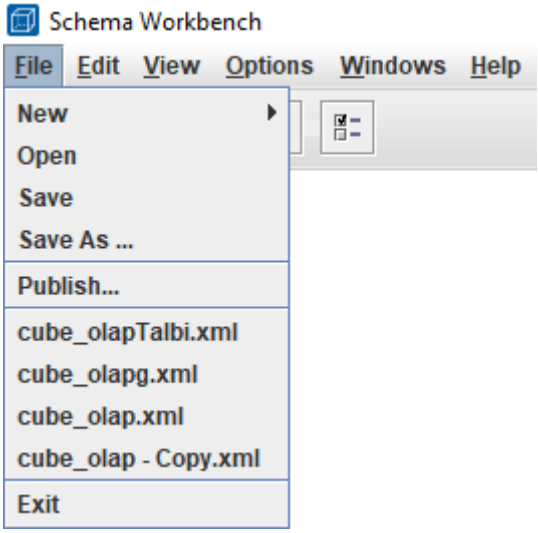


Figure 34 an example of how to create a Schema

2. The addition of a cube is typically the initial step in establishing a schema. To add a cube, use the context menu by right-clicking the schema icon in the schema window. The New Cube button in the toolbar is an alternative. Your schema will now have a brand-new default cube

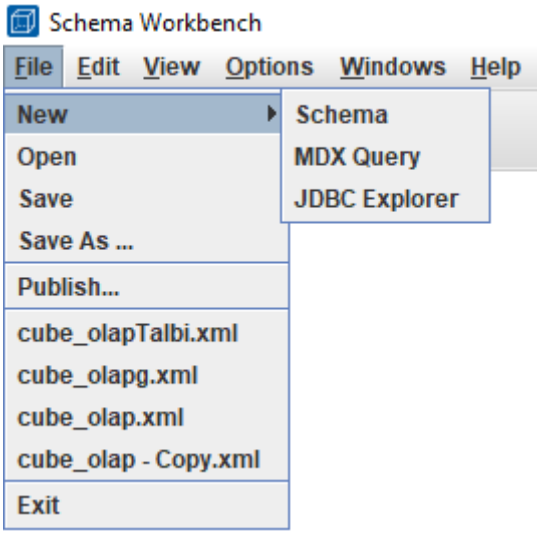


Figure 35 an example of how to create a Schema

3. Give your cube a name.

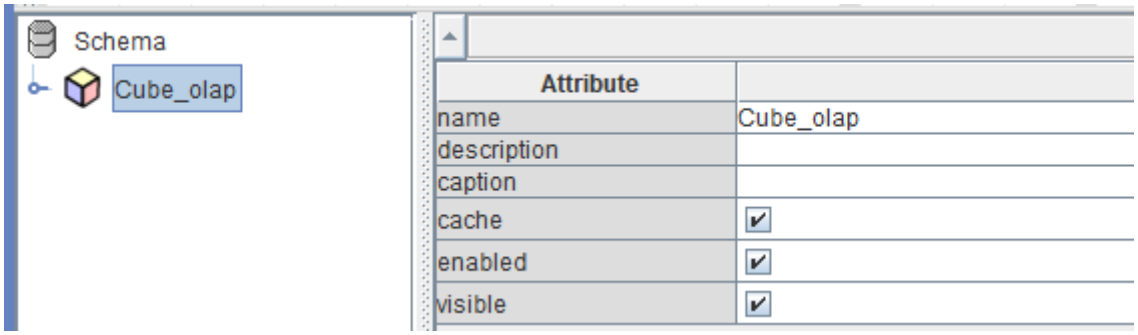


Figure 36 an example of making a cube in schema

4. Click the New Table button to add a table, or choose Add Table from the context menu when right-clicking your cube. This is going to be your fact sheet. If these are the data types you require for your fact table, you may also choose View or Inline Table.

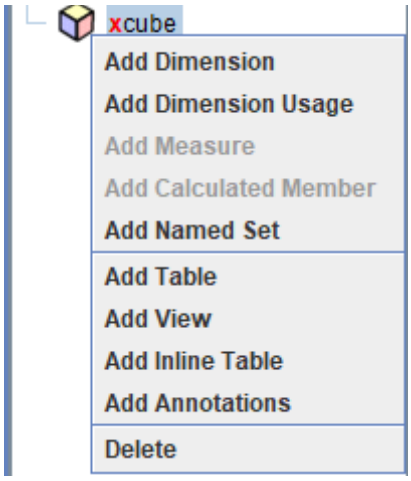


Figure 37 an example of setting Dimension in cube

5. Select or input the table name in the new table name box by clicking on the table element if you want to utilize a specific physical table as the factual table for this cube. Then select "add dimension" from the menu or just click "new dimension" to add dimension by clicking on the right cube name this dimension well in the name field. Select a foreign key for this dimension from the drop-down menu for foreign keys or from the field itself.

Attribute	
name	dim_cours
description	
foreignKey	RÉF_COURS
type	StandardDimension
usagePrefix	
caption	
visible	✔

Figure 38 an example of a dimension attribute

Figure 39

Additionally, if you add a new dimension, we automatically build a new hierarchy. By selecting the lever icon on the left side of the tree entry to the dimension, we may enlarge it. Next, we choose the new hierarchy 0 to create the hierarchy. Choose either an initial key table or an initial key.

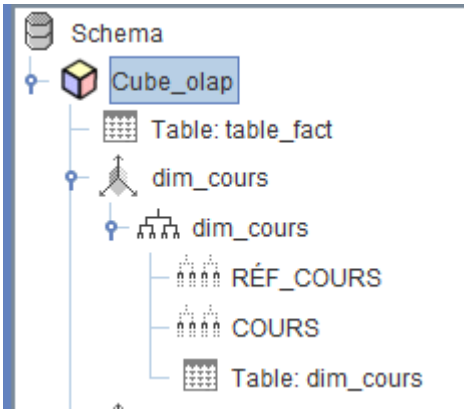


Figure 40 an example of a dimension with a hierarchy

You may insert a table into the hierarchy by selecting it from the context menu after clicking on the hierarchy a column should be used for the name property you may add a level to the hierarchy by clicking on it and selecting the contextual menu's "extra level" option.

Name level and column when set

Attribute	
name	dim_cours
description	
hasAll	✔
allMemberName	
allMemberCaption	
allLevelName	
defaultMember	
memberReaderClass	
primaryKeyTable	
primaryKey	RÉF_COURS
caption	
visible	✔

Figure 41 an example of a level attribute

Clicking on the appropriate level adds the member's feature to that level. Next, we choose the additional property from the context menu. Identify her and select her column.

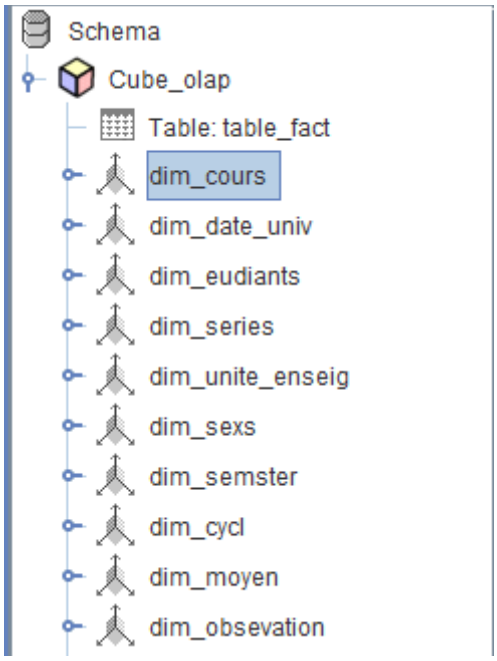


Figure 42 an example of every dimension in our cube

By selecting the new metric from the context menu after clicking on the right cube, adding a scale to the cube is as simple as choosing a column to give values, a pool to define how values are computed, and a column to provide values determine how the values should be calculated.

6. By right-clicking the cube and selecting Add Measure from the context menu, you may add a measure to it. Choose a column for which you wish to offer values, then choose an aggregator.

This cube's ultimate output for our project

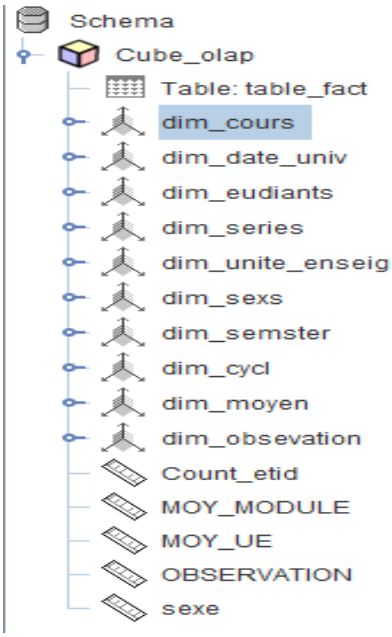


Figure 43 This cube's ultimate output for our project

2.6. Creating A Front End

In this part, we will go through how to use JasperReports to create a front-end to portray the results of an OLAP cube schema. Using the JasperReports Server REST API, users may connect JasperReports Server to an OLAP cube schema, generate a new report, design the report layout, and integrate the report into a front-end application.

2.6.1. What exactly is Jasper Server?

Jasper Server is a free and open source business intelligence platform that allows users to access, view, and analyze data from a variety of sources. It is a comprehensive set of tools for creating, deploying, and managing interactive reports, dashboards, and analytics.

The platform has a user-friendly interface, data integration, reporting, and analytics capabilities, among other things. Additionally, it provides data security, scalability, and performance. Jasper Server is an excellent option for any size corporation seeking to acquire insights into their data and make educated decisions.

2.6.2. Steps to design the report

To create and design the report, and use the REST API to integrate the report into your front-end application. Here are the general steps you should take:

Step 1: Install and configure JasperReports Server on your PC.

This step entails downloading and installing the JasperReports Server software on the PC or server where you wish to host the front-end. You will also need to set up and configure the server to connect to your OLAP cube data source, which may include setting up a JDBC or XMLA connection.

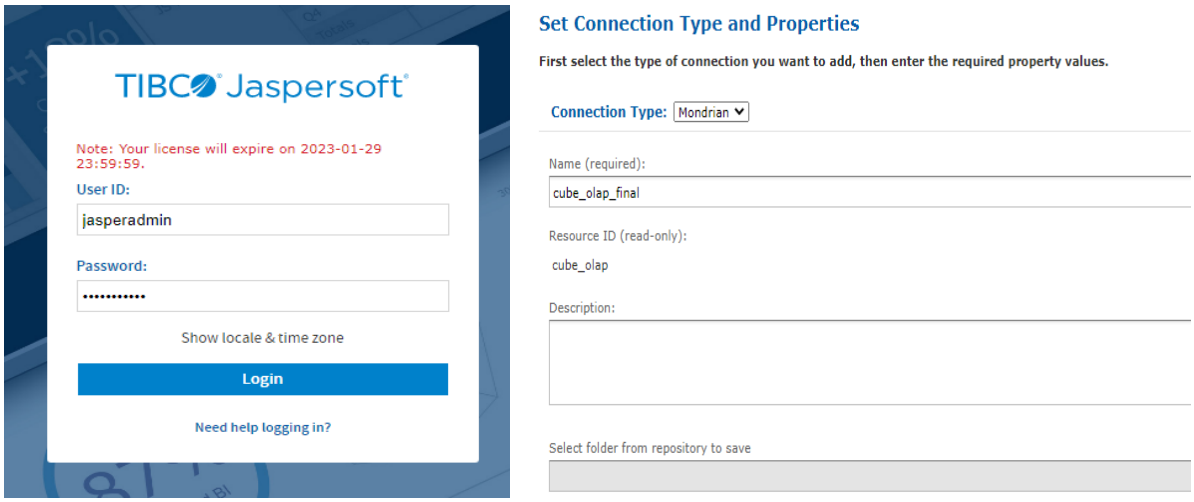


Figure 44 an example of a log in page and creating a new OLAP data source by giving the OLAP cube schema connection information

Step 2: In Jasper Reports Server, create a new report.

Log in to the web interface and click to the Reports area after the server is up and running.

Choose the option to generate a new report and the OLAP data source that you set up in step 1.

Design and configure the report layout using the web interface, which includes adding dimensions and measurements from the OLAP cube to the report fields and adding tables or crosstabs to present the data.

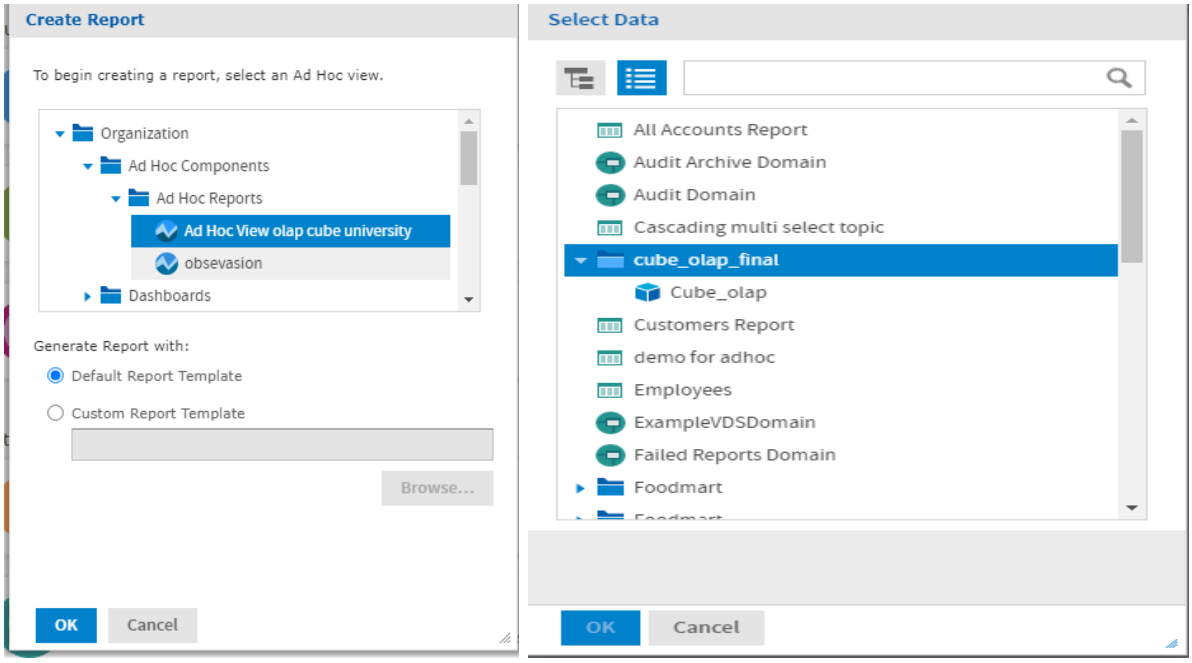


Figure 45 An example of how to select data and create a new report

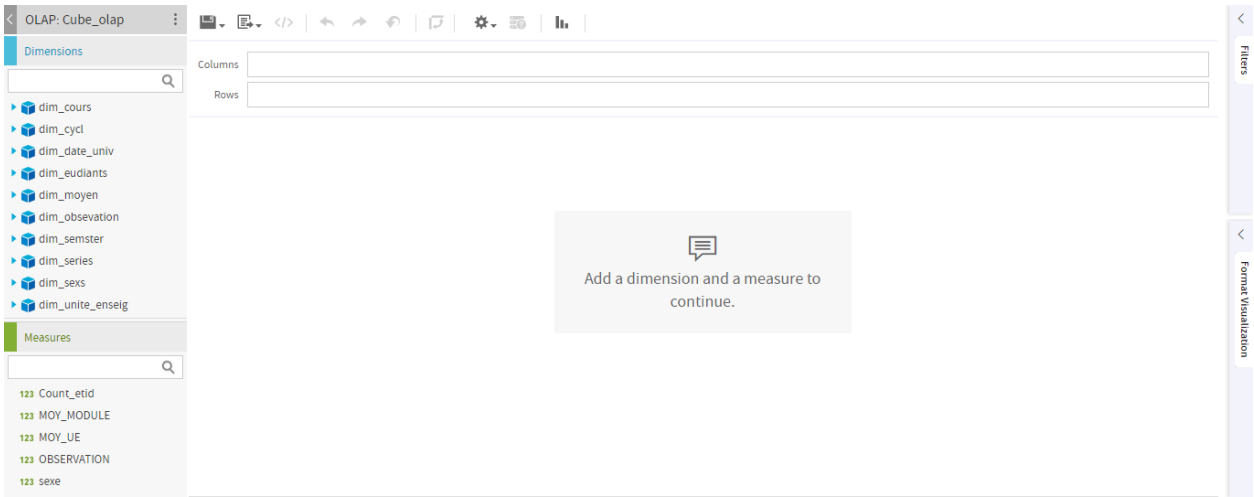


Figure 46 web interface to design and customize the report layout, which will include adding dimensions and measures from the OLAP cube to the report fields,

Step 3: Create charts, graphs, and other visual representations of the data using the built-in charting and visualization tools.

JasperReports Server has a number of built-in chart and graph creation capabilities, such as bar charts, line charts, pie charts, and more.

These tools may be used to generate visual representations of the data from the OLAP cube, making it easier to interpret and analyze the data.

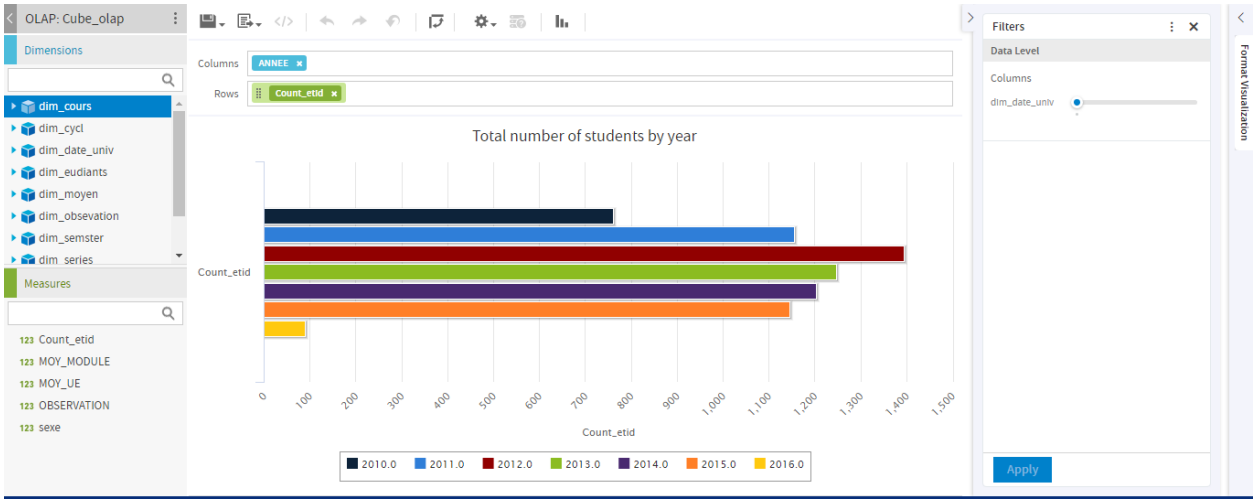


Figure 47 An example of graphs data report

Step 4: Review the report to make sure it shows the data correctly.

Before publishing the report, it must be viewed to ensure that it presents data correctly and that all areas and perceptions work as expected.

Step 5: Upload the report to the JasperReports server, allowing other users to access it and generate customised views.

Once the reports are complete, you may publish them on the JasperReports server for other users to read and engage with. Other users can construct their own views, which allows them to produce bespoke reports based on OLAP cube data.

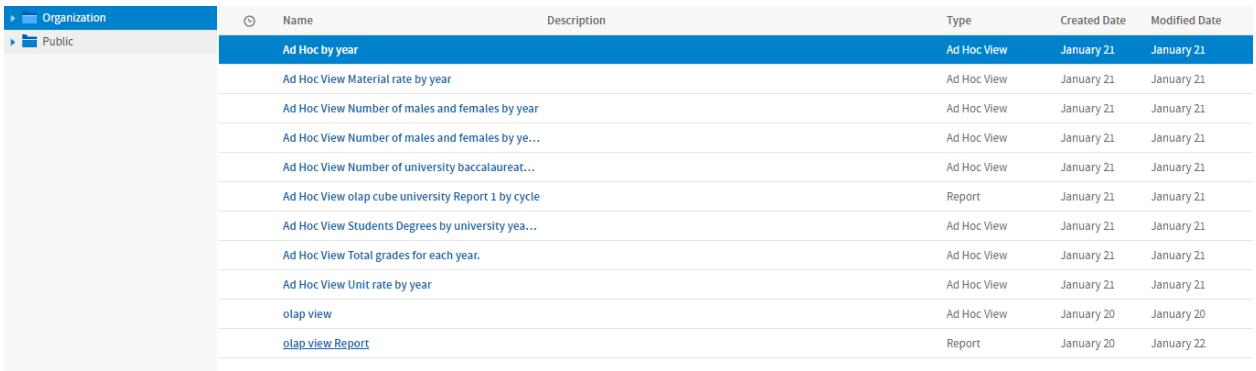


Figure 48 All of our project's reports

[Figure 49] shows us in which year certain subjects were studied. For example: Students who studied in the year 2015 studied heavy subjects like LCO, TECRX, which affected their results, causing the most of them to fail as seen in

[Figure 56]. And for that, we suggest balancing the subjects in the future.

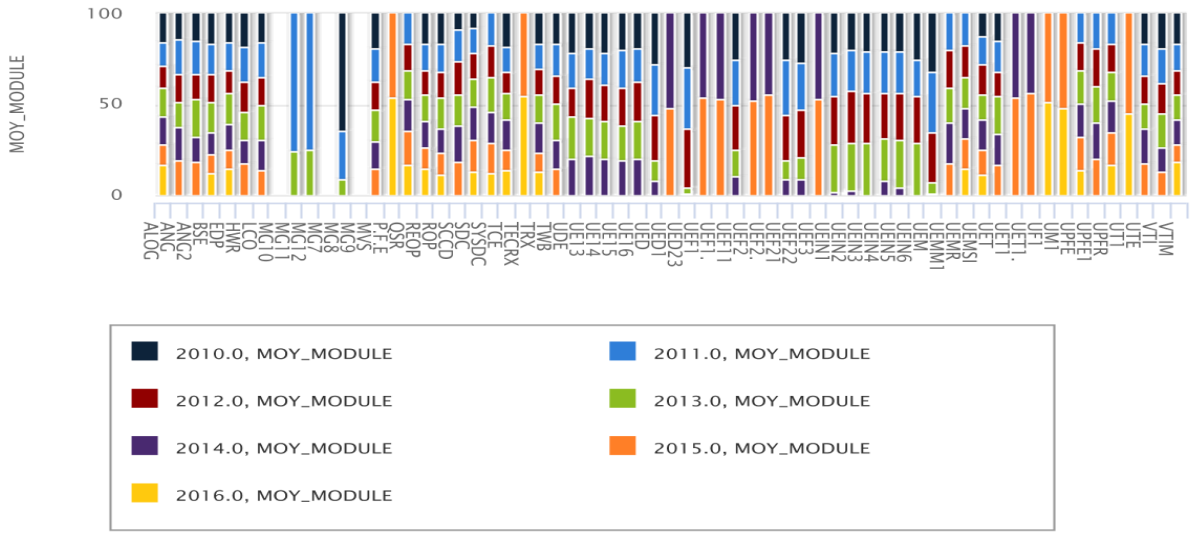


Figure 49Material rate by year

[Figure 50] shows us the total number of enrolled students each year. For example: The year 2012 had the most students enrolled, which was a big problem and caused almost half of them to fail as seen in [Figure 56]. And for that, we suggest putting a limit to how many students can enroll each year in the future.

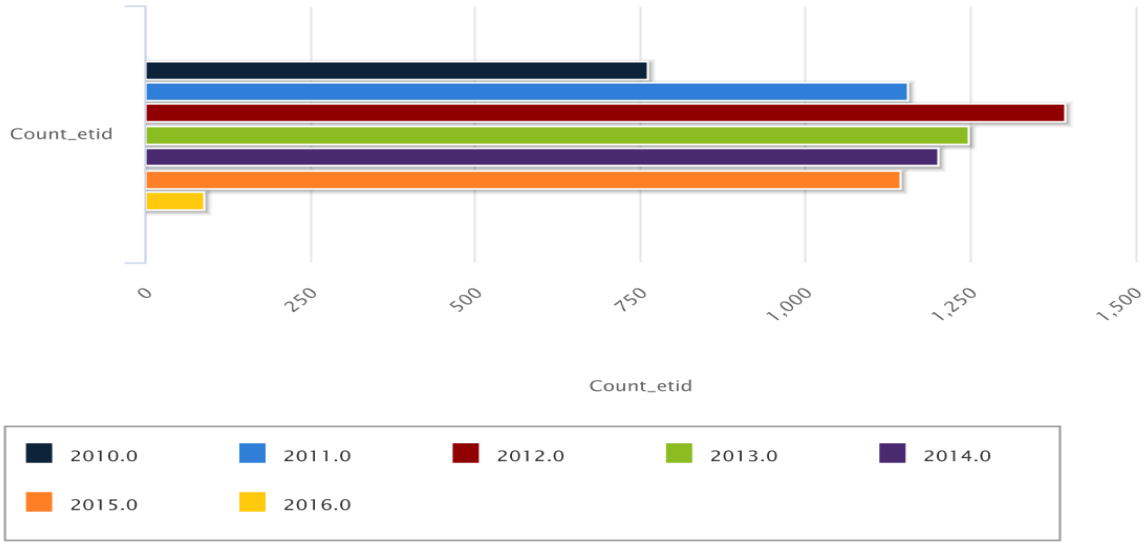


Figure 50Total number of students by year

[Figure 51] and [Figure 52] show us the distribution of male, female by year across different cycles and specialties of their education. For example: In 2016,

we can obviously see that there was an increase in the number of females enrolled. And in that year the success rate was almost 100% as shown in [Figure 56]. And for that, we suggest an increase in the female-to-male ratio.

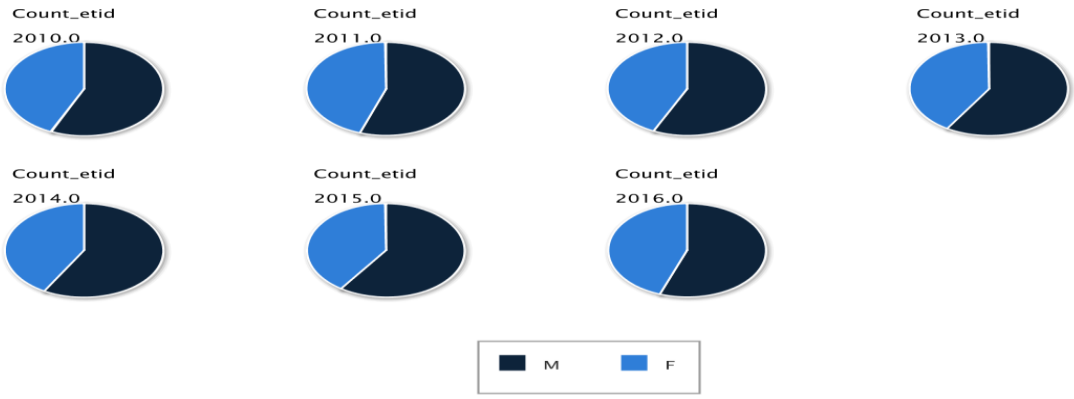


Figure 51Number of males and females by year

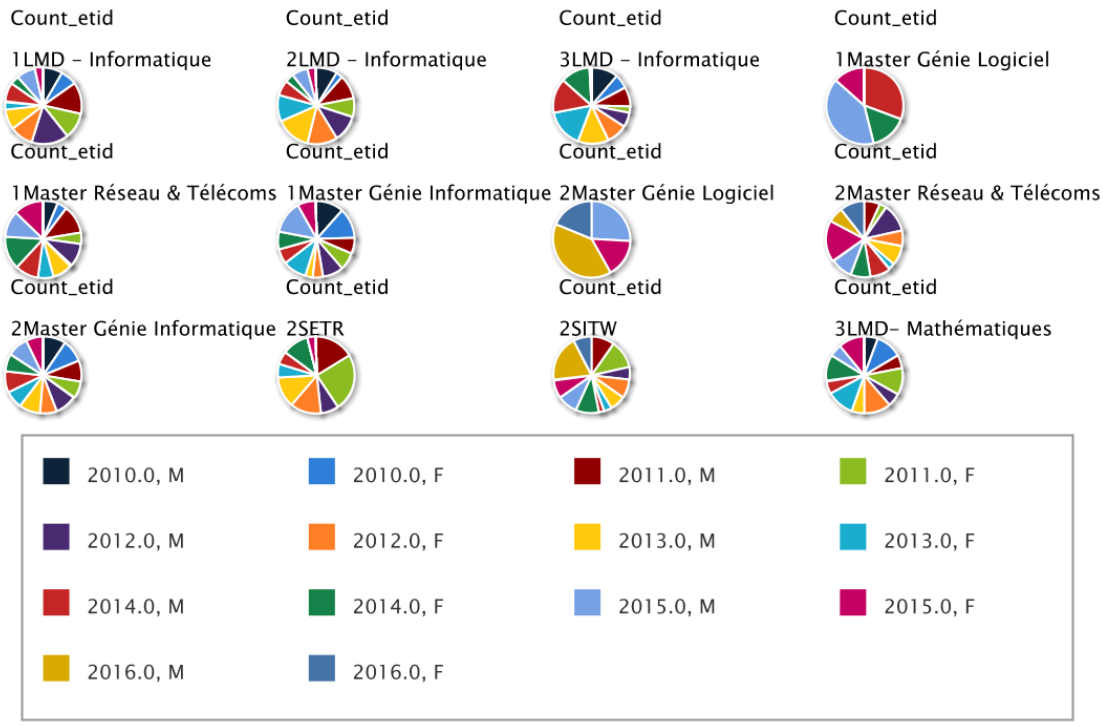


Figure 52Number of males and females by year and cycel

[Figure 53] shows us which devision of baccalaureate did the students from each year graduate from. For example: In the year 2011, we see that students enrolled in this university graduated their baccalaureate from all around the devisions, and the highest count graduating from a foreign school. And for that, we suggest and increase in the number of students sent to study abroad.

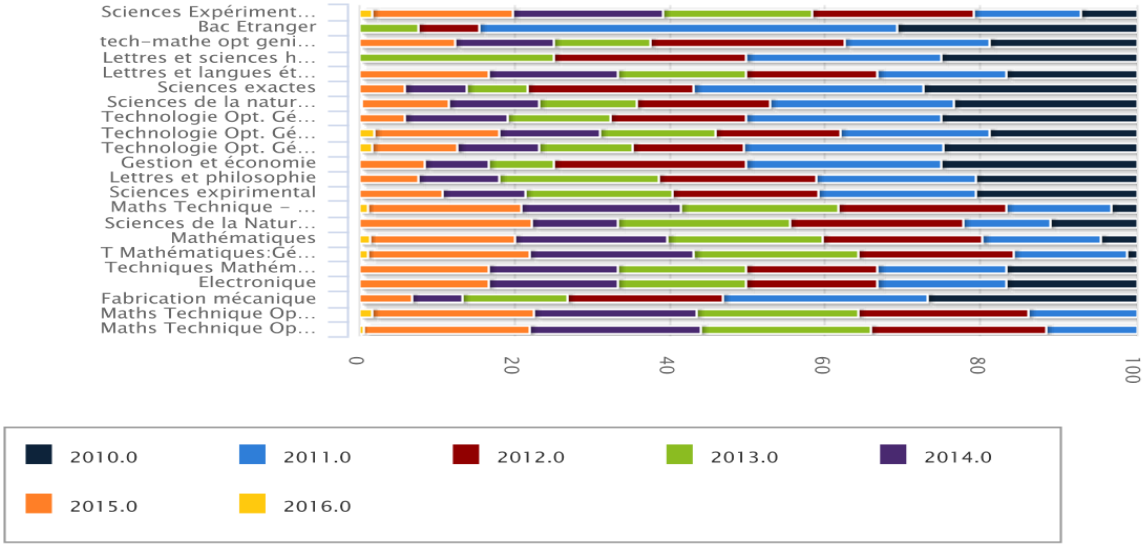


Figure 53Number of university baccalaureate disciplines by year

[Figure 54] and [Figure 55] show us the total number of students per cycle, and their grades. For example: We see that almost all 2nd year LMD students fail the year. And for that, we suggest helping them graduate more by making the subjects easier to understand or having more thought-provoking in-class questions.

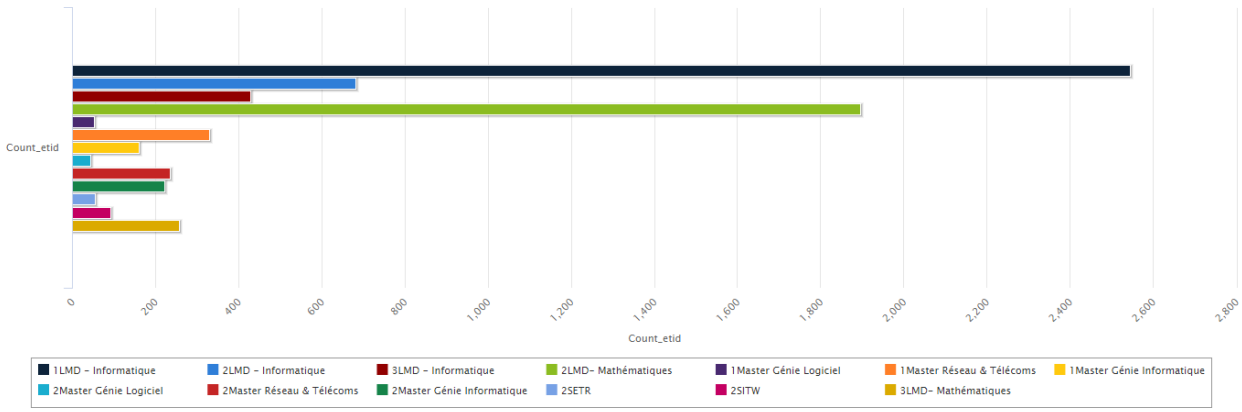


Figure 54 Total number of students by cycle

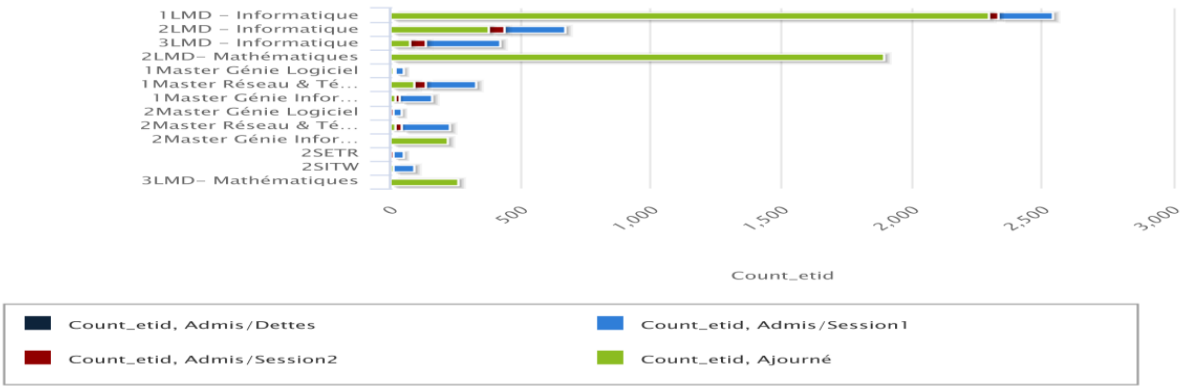


Figure 55 Students Degrees by university years Specialisation.

[Figure 56] helps us understand more by showing us the results of the education, while [Figure 57] shows us which subjects were taught each year. And that helps us know which subjects were hard to teach more than others, and which ones were too hard to teach and therefore discontinued.

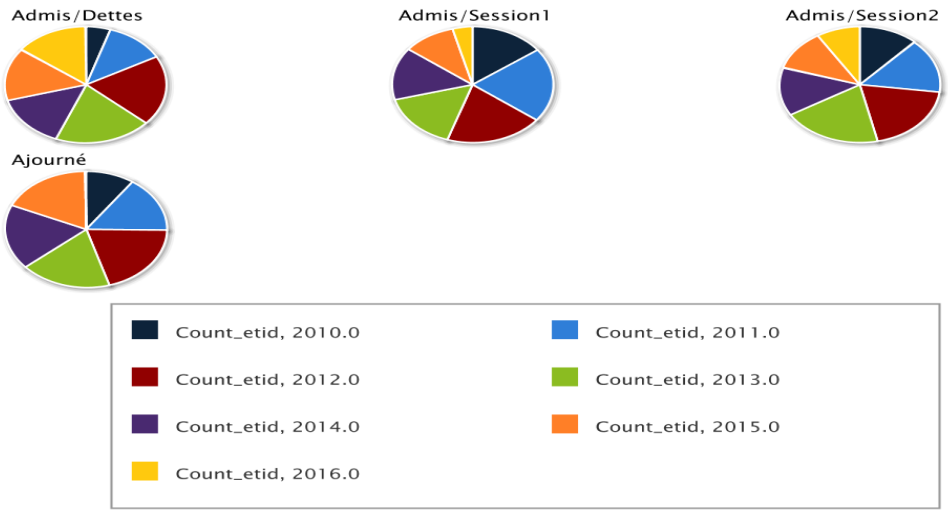


Figure 56 Total grades for each year.

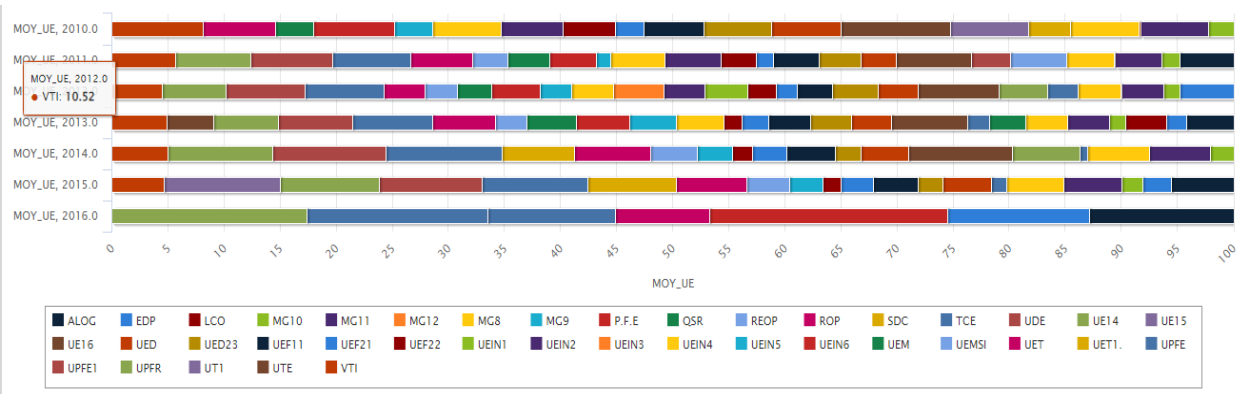


Figure 57 Unit rate by year

Chapter 3

Implement website

1. Our Brand Definition

1.1. The naming logo:

The name NULL just came by accident when we were arguing about what should we make the name of our group. After 2 days of simply going back and forth about which name sounded better or was written better we came to a stalemate. Then, out of nowhere, Mokhtari said that we should make it "null" because we were arguing too much, and it would make sense, since we couldn't decide, to make it "Group_Name = null".

1.2. The logo:

After deciding on "null", we agreed that I would make it, and so I did. I used a Black background and White text to represent the binary system; the 1 and 0 of colors. I made the letters all CAPITAL to attract attention and to look nice. I also used the brackets "[" , "]" because I like programming and it makes it look like we're professionals. But it still felt incomplete somehow. At that time, I asked advice from friends and family about what should I do to make it better, and one of my friends said "why dont you mirror half of the letters like some big corporations?", so I did as they said and it turned out perfect. And that's how it looks now.

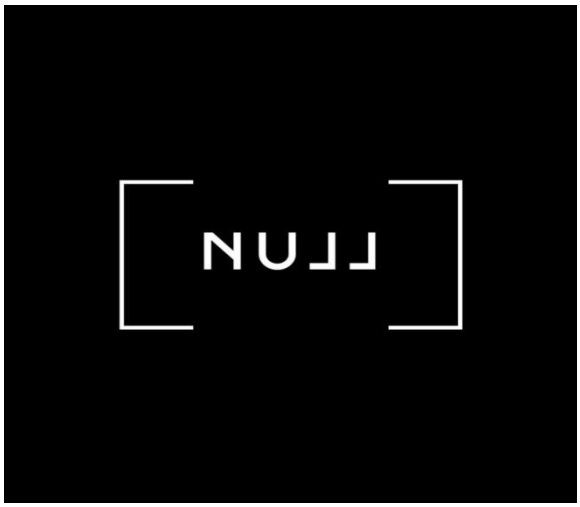


Figure 58 our logo

2. Introduction :

In this chapter we will present the conceptual studies of the data and processing of our system, we present in this chapter the implementation of the management website of the university federation.

3. Environment and development tools and modelization:

3.1. Programming languages:

3.1.1. HTML:

HTML (HyperText Markup Language) is a markup language used to represent the structure of a page: titles, subtitles, paragraphs, images, input forms, hypertext links, etc. It is the basis of a web page.

3.1.2. CSS :

CSS(Cascading Style Sheets) for Cascading Style Sheets, is a computer language used on the Internet for formatting files and HTML pages.

3.1.3. JS :

JS (JavaScript) is a complete dynamic programming language which, applied to an HTML document, can provide dynamic interactivity on websites.

3.1.4. PHP :

PHP (HyperText Preprocessor) is a programming language free, mainly used toproduce dynamic web pages via an HTTP server (The HyperText Transfer Protocol), but can also function as any local interpreter language. PHP is an imperative object-oriented language.

3.1.5. Bootstrap :

Bootstrap is a Framework developed by the Twitter social network team. Proposes in open source, this Framework using the languages HTML, CSS and JavaScript provide developers with tools to create a site easily. This Framework is designed to develop sites with a responsive design.

3.1.6. SQL :

SQL (Structured Query Language) is a computer language des-tine to query or control a database.

3.2. Working environment:

3.2.1. Visual Studio :

Visual Studio Code is an extensible code editor developed by Microsoft for Windows, Linux and Marcos. Features include debugging support, syntax highlighting, smart code completion, code snippets, code refactoring.

3.2.2. XAMPP :

XAMPP is a set of software allowing to set up a local web server, FTP(File Transfer Protocol) is a server for email. This is a free software distribution offering good flexibility of use, renowned for its simple and quick installation.

4. App Description:

We will present the database tables and the main features of our website.

	Table	Action	Lignes	Type	Interclassement	Taille	Perte
☐	admin	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	2,8 kio	-
☐	document	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	2,2 kio	212 o
☐	signetud	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
☐	skills	Parcourir Structure Rechercher Insérer Vider Supprimer	13	MyISAM	latin1_swedish_ci	3,3 kio	-
☐	talnd	Parcourir Structure Rechercher Insérer Vider Supprimer	14	MyISAM	latin1_swedish_ci	2,2 kio	-
☐	worke	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	3,2 kio	-
	6 tables	Somme	39	InnoDB	utf8mb4_general_ci	29,7 kio	212 o

Figure 59Data bases of our website

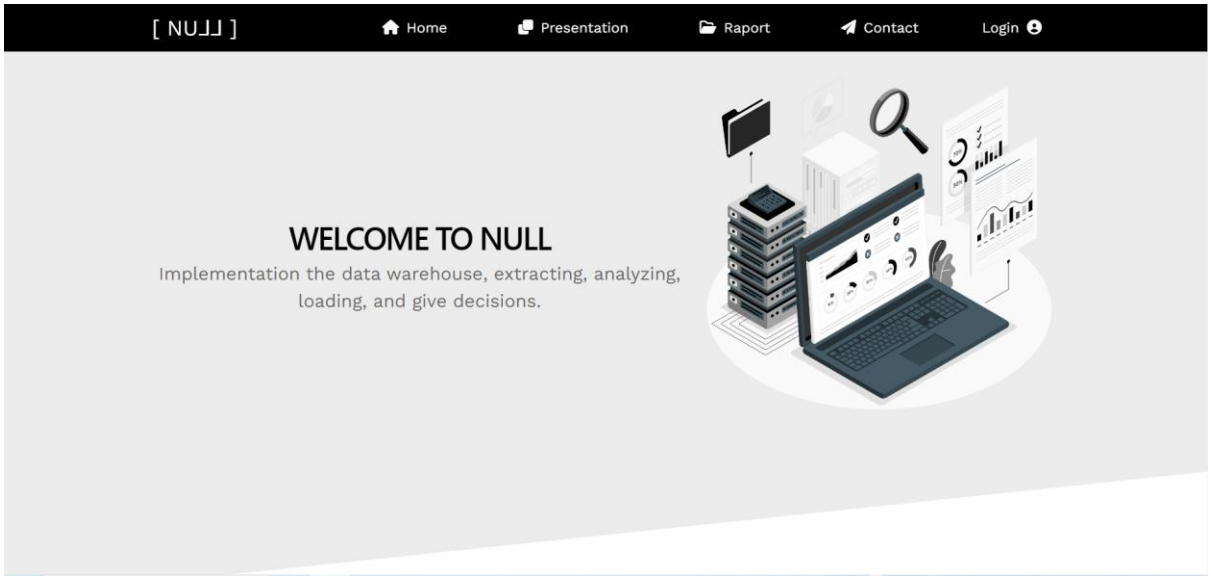


Figure 60 Home page

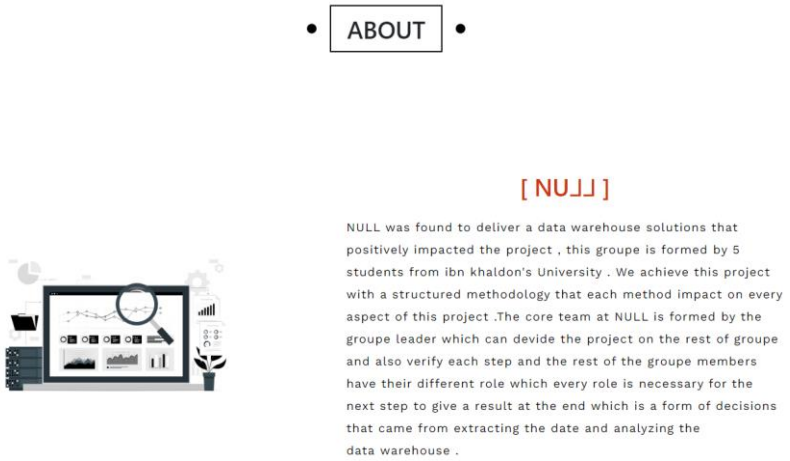


Figure 61 About page

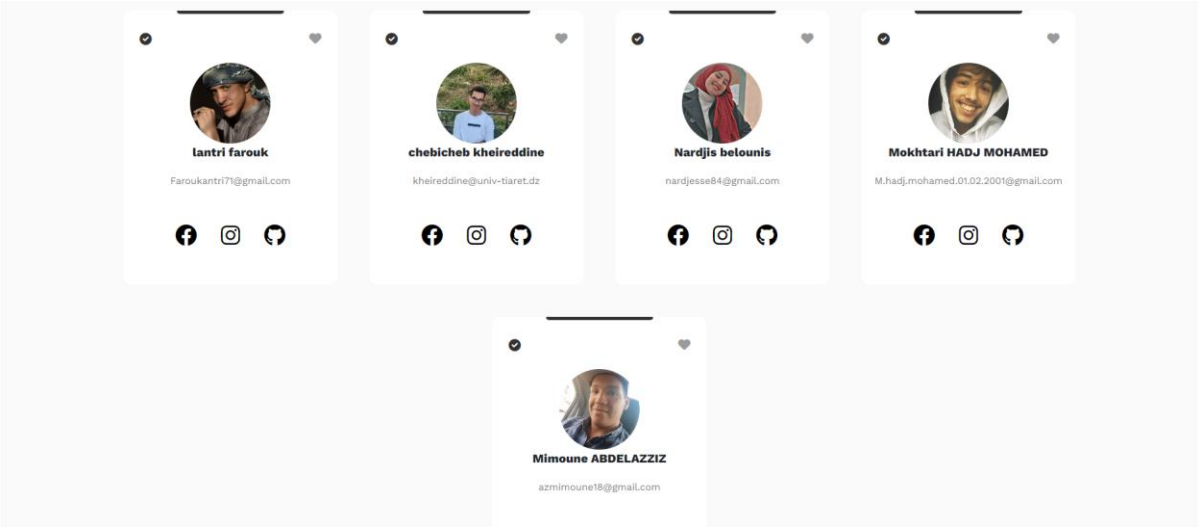


Figure 62 Our team

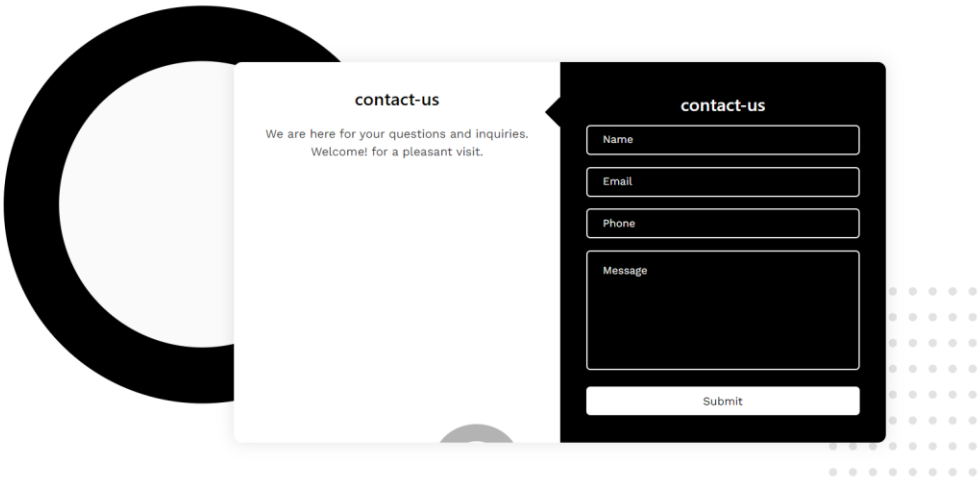


Figure 63 Contact Page

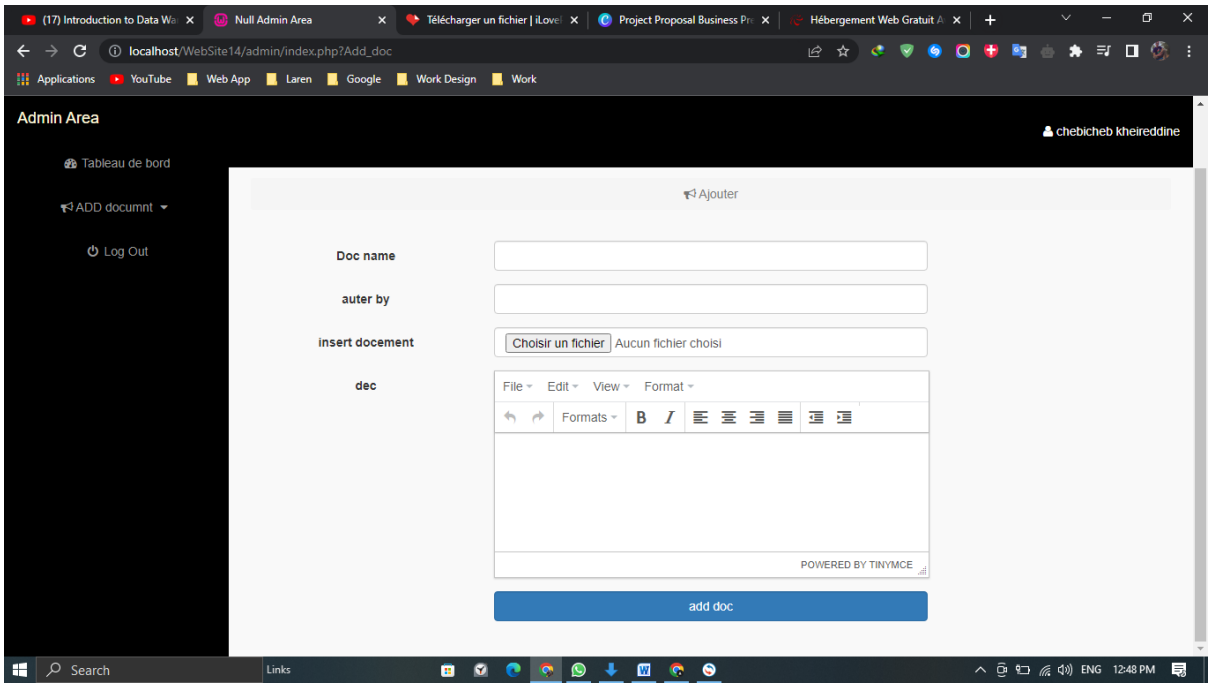


Figure 64 add files

Conclusion

Data warehouses are an essential part of any organization's data infrastructure. They provide organizations with a centralized repository for data from multiple sources, enabling them to analyze data from different perspectives and gain insights into their operations. Data warehouses also enable organizations to store large amounts of data in a secure and efficient way.