

# **Data Pipelines for Stellantis Carflow Project (Digital Marketing Dashboards)**

**-STELLANTIS-  
-African Technical Center-**

**Mémoire de fin d'études présenté pour l'obtention  
Du diplôme d'Ingénieur d'Etat  
De l'Ecole des Sciences de l'Information**

**Salma OUARDI**

**Sous la direction de : Siham YOUSFI**

**Membres de jury :**

**Président : Anass MAMOUNY**

**Encadrante : Siham YOUSFI**

**Co-encadrante : Maryem RHANOUI**

**Tuteur : Amine BENMOUSSA**

**Promotion 2022**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَقُلْ إِنِّي مَوْلَاكُمْ وَأَنَا الْمَوْلَى الَّذِي عَلَيْكُمْ وَأَنَا الْمَوْلَى الَّذِي عَلَيْكُمْ وَأَنَا الْمَوْلَى الَّذِي عَلَيْكُمْ

اللَّهُ  
صَلَّى  
الْعَظِيمِ

# Dedication

To Allah the almighty and the merciful

To my parents, my reason for living, my love, my life.

Thank you for your support, your trust, your encouragement, and your sacrifices.

I owe you what I am today and what I will be tomorrow, and I will always do my best to  
remain your pride.

Thank you for your support and prayers

I love you all.

To Kawtar my little sister for her encouragement and help.

To my baby brothers for existing in my life and making it better.

To my friends: Ahmed, Kenza, Meryem, Yassine, Nada, Soundous, Younes, Ayman, and Jing  
for their unconditional love and support, I'm lucky to have you all in my life.

To my classmates and my seniors who helped a lot during this period especially: Ayman,  
Reda, and Nour Eddine.

To those who believe in me,

To those who love me,

To those I love.

I dedicate this work to you

# Acknowledgment

First of all, I would like to thank Allah the almighty and merciful, who gave me the energy and the patience to accomplish this modest work.

I would like to express my deepest gratitude and thanks to my teacher and supervisor, Mrs. Siham YOUSFI, for the help she had given me during the whole period of my internship, it's stunning how she can explain complex concepts that I struggle to understand in the simplest ways, I love how she encouraged me to finish my report on time so I graduate the same time as my classmates, and above all, whenever I have a problem that I think can't be solved and I get into a negative zone of thinking, I call her and she gets out of there easily, she eases my stress. I would like to express my respect to you for your efforts.

I would like to thank especially Mrs. Maryem RHANOUI for her precious advice and her collaboration with us to improve the quality of this deliverable.

I sincerely thank my tutor Mr. Amine BENMOUSSA in Stellantis, with whom I had the chance to work, for his confidence and his support throughout the internship period, I appreciate his hard work and seriousness.

I would also like to thank Anass LAHMAMSI for his help and guidance.

I also want to thank Mohamed EL KANOUNI, and Omar AAKI from the business team for their help and support all the time. Also, I would like to express my gratitude to Gerard ROUGIER, Christophe SERWINSKI, and Asma JOUINI from the Data team for their guidance and help.

I would like to express my sincere thanks to all the professors at the School of Information Sciences who have taught me and who have supported me in pursuing my studies with their expertise.

And finally, I would like to thank all the people who have contributed in any way to the realization of this work

---

# Abstract

The present report summarizes the outcome of my end-of-study project within ATC Stellantis. This project aims to leverage Data Engineering and ETL workflows to create data pipelines that enable the flow of data from several applications to a Datalab, the data at the end of the pipe is used to create Power BI dashboards for the Carflow Project which helps the Direction of Sales and Marketing of Stellantis to have an in-depth view of the overall operations of Carflow in the region of the Middle East and Africa (MEA).

The success of this project depends on 5 main steps: the project context, the literature review, the analysis of the existing solution, the conception and design of the new solution, and finally the implementation and result of said solution. The following report goes in the same order.

Keywords: Data Engineering, ETL workflows, Data, Datalab, Data Pipeline, Power BI dashboards.

---

---

# Résumé

Le présent rapport résume les résultats de mon projet de fin d'études au sein d'ATC Stellantis. Ce projet vise à exploiter le Data Engineering et les flux ETL pour créer des data pipelines qui permettent le flux de données de plusieurs applications vers un Datalab, les données à la fin du pipeline sont utilisées pour créer des tableaux de bord Power BI pour le projet Carflow qui aide la Direction des Ventes et du Marketing de Stellantis à avoir une vue approfondie des opérations globales de Carflow dans la région du Moyen-Orient et de l'Afrique (MEA).

Le succès de ce projet dépend de 5 étapes principales : Le contexte du projet, La revue de littérature, L'analyse de la solution existante, la conception et le design de la nouvelle solution et enfin l'implémentation et le résultat de cette dernière. Le rapport suivant suit le même ordre.

Mots-clés : Data Engineering, flux ETL, données, Datalab, Data Pipeline, tableaux de bord Power BI.

---

# Table of Figures

Figure 1: Logo of Stellantis	21
Figure 2: Logo of Fiat Chrysler Automobiles	22
Figure 3: Logo of PSA (Peugeot S.A)	22
Figure 4: Image for Stellantis brands	25
Figure 5: Stellantis plant in Kenitra	28
Figure 6: ICT MEA Organigram	29
Figure 7: ICT MEA Organigram	29
Figure 8: CARFLOW Entity Organigram	30
Figure 9: Gantt Diagram	32
Figure 10: Cycle Programme Explanation	35
Figure 11: The Benefits of Business Intelligence	37
Figure 12: The Characteristics of Data Warehouse	38
Figure 13: ETL Process	39
Figure 14: Batch vs Streaming Processes	40
Figure 15: DAG Example	41
Figure 16: Dag code example	42
Figure 17: Cron Syntax	42
Figure 18: Carflow Project Steps	45
Figure 19: Technical Architecture of Step 3	46
Figure 20: Current Solution Architecture	47
Figure 21: Project Needs	48
Figure 22: Conception Diagram	50
Figure 23: The Data Model	62
Figure 24: Functional Architecture of the Data Pipeline	63
Figure 25: Oracle SQL Developer Icon	68
Figure 26: Apache Airflow Icon	69
Figure 27: Putty Icon	70
Figure 28: Visual Studio Code Icon	71
Figure 29: WinSCP Icon	71
Figure 30: iCubeData Pipeline Code Snippet 1	72
Figure 31: iCubeData Pipeline Code Snippet 2	72
Figure 32: iCubeData Pipeline Code Snippet 3	73
Figure 33: iCubeData Pipeline Code Snippet 4	73
Figure 34: iCubeData Pipeline Code Snippet 5	73
Figure 35: iCubeData Pipeline Code Snippet 6	74
Figure 36: Data Output after transformation	74
Figure 37: iCubeData Pipeline Code Snippet 7	74
Figure 38: iCubeData Pipeline Code Snippet 8	75
Figure 39: write_df_to_oracle function	75
Figure 40: iCubeData Pipeline Code Snippet 9	76
Figure 41: iCube Shell Script	76
Figure 42: iCube Airflow Imports	76
Figure 43: iCube Airflow Credentials	77
Figure 44: iCube Airflow DAG Arguments	77
Figure 45: iCube Airflow DAG Definition	78
Figure 46: iCube Airflow BashOperator	78



Figure 47: The SQL script to Extract the Production demands from Eprog part(1)	80
Figure 48: The SQL script to Extract the Production demands from Eprog part(2)	81
Figure 49: Eprog Pipeline Code Snippet 1	82
Figure 50:Eprog Pipeline Code Snippet 2	82
Figure 51: Eprog Pipeline Code Snippet 3	83
Figure 52: Eprog Pipeline Code Snippet 4	83
Figure 53: Eprog Pipeline Code Snippet 5	84
Figure 54: Eprog Pipeline Code Snippet 6	84
Figure 55: Eprog Pipeline Code Snippet 7	85
Figure 56: Eprog Shell Script	85
Figure 57: Eprog Airflow DAG	86
Figure 58: Eprog Airflow BashOperator	86
Figure 59: Madax SQL Script 1	87
Figure 60: Madax SQL Script 2	88
Figure 61: Madax SQL Script 3	88
Figure 62: Madax Pipeline Code Snippet 1	89
Figure 63:Madax Pipeline Code Snippet 2	90
Figure 64: Madax Pipeline Code Snippet 3	90
Figure 65: Shell Script Wholesales	91
Figure 66: Shell Script Retails	91
Figure 67: Madax Airflow DAG	91
Figure 68: Madax Airflow BashOperator	92

# List of Tables

Table 1: Stellantis Brands	24
Table 2: Data in BRC and DLK	53
Table 3: KPIs Structure	55
Table 4: Type of KPI	55
Table 5: Nature of KPIs	55
Table 6: Axes of Analysis	56
Table 7: Update Frequency	56
Table 8: Mapping of Data(Tables)	57
Table 9: Data availability Matrix	58
Table 10: CSF-Carflow Mapping	58
Table 11: EProg-Carflow Mapping	59
Table 12: Madax-Carflow Mapping	60
Table 13: iCube-Carflow Mapping	61
Table 14: Source Table Format	64
Table 15: Destination Data Format	64
Table 16: Data in the source	65
Table 17: Data in the destination	66
Table 18: Source Data	73
Table 19: Data output after transformation 2	75
Table 20: iCube Data Pipeline Result	79
Table 21: Data Output Eprog	84
Table 22: Eprog Data Pipeline Result	86
Table 23: Data output Madax	90
Table 24: Madax Data Pipeline Result	92

# Business Glossary

<i>Terms</i>	<i>Definition</i>
<i>NV</i>	New vehicles
<i>NSC</i>	Commercial subsidiaries in countries
<i>Registrations</i>	Total Number of vehicles registered in a market/country/region in a given period for a specific Brand
<i>Market</i>	Total Number of vehicles registered in a market/country/ region in a given period for all competitors
<i>Delivery</i>	Time of hand-over of the vehicle to the final customer (private, fleet, internal company car, or any other)
<i>Invoice</i>	NSC's invoicing to dealers/importers or direct sales customer
<i>Order</i>	Vehicle order that has a final customer identified
<i>Channel</i>	The channel through which the vehicle was sold: B2C (Vehicles registered by Private Customers for Passenger Cars), B2B (Vehicles registered by Fleet and Business Customers for Passenger Cars), RENTAL (LCD) (Sales to International Rental companies for Passenger Cars) ...
<i>Type ("genre" in French)</i>	Passenger cars (PC in English, VP in French) and light commercial vehicles (LCV in English, VU in French)
<i>PSV</i>	« Point de Structure de Vente ». An entity that has a trade relationship with Stellantis Group. Identified by 7 digits (RRDI code)
<i>CP</i>	Cycle Programme
<i>PRE-REF</i>	Pre Reference: The objective of the pre-Reference process is to compare the forecasts, to generate a list of scenarios. Pre Reference defines, builds, and valorizes scenarios. At the end of the Pre Reference, one or multiple valorized scenarios will be ready to be presented for the Reference meeting.
<i>REF</i>	The Reference Meeting validates sales forecasts aggregated at the Family / Sub Family level. At the end of the Reference meeting, the validated scenario is sent to the workflow to be displayed.
<i>Registrations</i> <i>Market share</i>	The number of vehicles registered divided by the market in registrations (all brands) in the same time frame in the same geographical area/country Calculation method: Sales in the market divided by total market Sales at each time If registration is not available, then the calculation is Deliveries/Market

# List of Abbreviations

The Abbreviation	The Significance
<b>MEA</b>	Middle East and Africa
<b>ETL</b>	Extract, Transform, Load
<b>ELT</b>	Extract, Load, Transform
<b>BI</b>	Business Intelligence
<b>PSA</b>	Peugeot S.A
<b>FCA</b>	Fiat Chrysler Automobiles
<b>AC</b>	Automobile Citroen
<b>AP</b>	Automobile Peugeot
<b>OV</b>	Opel / Vauxhall
<b>DMOA</b>	Direction Moyen-Orient Afrique
<b>DLK</b>	Datalake
<b>BRC</b>	Bigdata Ressource Classificator
<b>KPI</b>	Key Performance Indicator
<b>ARC</b>	Stellantis platform dedicated for Data Science
<b>PCOM</b>	Pays Commercial
<b>PPROG</b>	Pays Programme
<b>OLAP</b>	Online Analytical Processing
<b>SQL</b>	Structured Query Language
<b>CRM</b>	Customer Relationship Management
<b>DAG</b>	Directed Acyclic Graph
<b>HDFS</b>	Hadoop Distributed File System
<b>XLS</b>	Excel Spreadsheet
<b>B-B</b>	Business-to-Business service
<b>B-C</b>	Business-to-consumer
<b>CSF</b>	Country Sales Forecast
<b>PC</b>	Passenger Car
<b>LCV</b>	Light Commercial Vehicles

<b>LCDV</b>	Langage Commun de Définition Véhicule » Stellantis Product Code
<b>IDE</b>	Integrated Development Environment
<b>DBA</b>	Database Administrator
<b>SSH</b>	Secure Shell Protocol
<b>FTP</b>	File Transfer Protocol
<b>SFTP</b>	SSH File Transfer Protocol
<b>WS</b>	Wholesales
<b>ACT</b>	Actuals

# General Introduction

Today, Business Intelligence has become a real stepping-stone for the decision-making process of companies. It brings together tools and processes that allow companies to collect information and analyze it to improve their decision-making. In practical terms, this means collecting an important source of data, analyzing it, and transforming it into usable information.

Through data analysis, the company can better identify market trends. It has a broader view of its environment and its various activities. This allows it to detect any possible errors and make improvements. Business intelligence thus represents a factor of productivity and efficiency. The collected data allows us to take not only tactical but also strategic decisions<sup>1</sup>.

ETL (Extract, transform and load) is an important part of today's business intelligence (BI) processes and systems. It is the IT process by which data from various disparate sources can be placed in one place to analyze and discover business information programmatically<sup>2</sup>.

Realizing the importance of Business Intelligence, Stellantis decided to be proactive and integrate this world to take advantage of its various technologies and innovations to take better advantage of their data instead of archiving it and leaving it useless and unexploitable. For their project Carflow, they decided to build several dashboards to keep track of the operations in the region of MEA. During their first try to create the said dashboards, they went with a manual approach where they extract the data, transform it, and load it manually into their Datalab.

As they wanted to go with a more automatic approach, we decided to build ETL pipelines to get the data from various heterogeneous sources transform it, and load it to our Datalab Oracle Exadata automatically.

---

<sup>1</sup> Groupe Calliope. (2022). *Qu'est-ce que la Business Intelligence dans l'entreprise ?* [online] Available at: <https://www.groupe-calliope.com/business-intelligence/qu-est-ce-que-business-intelligence/#:~:text=Quelle%20est%20l> [Accessed 6 Jul. 2022].

<sup>2</sup> Lambert, N. (2020). *Pourquoi utiliser un ETL ? Avantages, enjeux et cas d'usage*. [online] Axysweb. Available at: <https://www.axysweb.com/pourquoi-utiliser-un-etl/> [Accessed 6 Jul. 2022].

This report describes the different steps followed for the implementation of our project:

- Chapter 1: Project Context: This chapter provides a general presentation and overview of the project environment. In the first part we are going to present the host company, its Carflow entity and their missions, on the second part of it we will present the project, expose the problem to be solved and the objectives and the project planification, and finally a conclusion.
- Chapter 2: Literature Review: In this chapter, we will discuss the theoretical background where we will present and define the different concepts be it business or technical ones related to the project, to have a clear understanding of the rest.
- Chapter 3: Analysis of the existent: After defining the theoretical scope of our project and identifying the different concepts addressed. In this chapter, we will start with an analysis of the existing situation. Indeed, this phase is essential for the conception and design of the project and to fulfill its objectives, which allows us to realize our ETL pipeline.
- Chapter 4: Conception and Design: After having specified the needs, in this chapter, we will define the general conception adapted to the needs to frame functionally and technically our project and to detail each stage of the realization of our solution.
- Chapter 5: Implementation and Results: In this chapter, we'll witness the implementation and the result of our project.

# Problem Definition

As mentioned in the introduction, Stellantis has already implemented a solution to create its Carflow dashboards with the data provided from different resources.

The problem with the existing system is that the whole ETL process is manual, where an employee in the carflow entity had to extract the data manually from the sources, send it as an attachment in an email or share it on SharePoint, so that the BI consultant or data analyst has to download it and perform the transformations manually using Excel's power query, and then load it again manually in the correspondent table in the Oracle Exadata Datalab.

This whole manual process demands a lot of energy and precious time and above all, it is repetitive and therefore becomes a burden to the person performing the refresh each month.

Our mission is to create ETL pipelines to automate the whole process mainly using python and SQL scripts and Airflow dags as automation jobs.



# Table of Contents

<b>Dedication</b>	<b>3</b>
<b>Acknowledgement</b>	<b>4</b>
<b>Abstract</b>	<b>6</b>
<b>Résumé</b>	<b>7</b>
<b>Table of Figures</b>	<b>8</b>
<b>List of Tables</b>	<b>10</b>
<b>Business Glossary</b>	<b>11</b>
<b>List of Abbreviations</b>	<b>12</b>
<b>General Introduction</b>	<b>14</b>
<b>Problem Definition</b>	<b>16</b>
<b>Chapter One: Project Context</b>	<b>20</b>
<b>1. Introduction</b>	<b>21</b>
<b>2. Company Presentation</b>	<b>21</b>
2.1. General introduction	21
2.1.1. History and highlights of the Group Stellantis	22
2.1.2. The Group's brands	24
2.2. The group's sectors of activity	26
2.3. The group 2021 sales	26
<b>3. Stellantis in Morocco</b>	<b>27</b>
3.1. Stellantis Plant in Kenitra	27
3.2. Organigram of <i>ICT MEA</i>	29
<b>4. Carflow entity presentation</b>	<b>29</b>
4.1. Introduction of the entity	29
4.2. Organigram of Carflow	30
<b>5. Presentation of the project</b>	<b>31</b>
5.1. Main Problematic	31
5.2. Objectives	31
5.3. The Method of implementation	31
5.4. Project Planning	32
<b>6. Conclusion</b>	<b>32</b>
<b>Chapter Two: Literature Review</b>	<b>33</b>
<b>1. Introduction</b>	<b>34</b>
<b>2. Business Concepts</b>	<b>34</b>
<b>3. Technical Concepts</b>	<b>36</b>
3.1. Business Intelligence	36

3.2.	Data Warehouse	37
3.3.	ETL (Extract, Transform and Load)	38
3.4.	Data Pipeline	40
3.5.	ETL vs Data Pipeline	41
3.6.	ETL Pipelines	41
3.7.	Airflow Dags	41
<b>4.</b>	<b>Conclusion</b>	<b>43</b>
<b>Chapter Three: Analysis of the existent</b>		<b>44</b>
<b>1.</b>	<b>Introduction</b>	<b>45</b>
<b>2.</b>	<b>Analysis of the existent</b>	<b>45</b>
2.1.	The Carflow MEA Dashboards project steps	45
2.2.	The Current Solution	46
2.3.	Problems with the current solution	47
<b>3.</b>	<b>Project Requirements and needs</b>	<b>48</b>
<b>4.</b>	<b>Conclusion</b>	<b>48</b>
<b>Chapter Four: Conception and Design</b>		<b>49</b>
<b>1.</b>	<b>Introduction</b>	<b>50</b>
<b>2.</b>	<b>General Conception</b>	<b>50</b>
<b>3.</b>	<b>Environment Setting</b>	<b>51</b>
3.1.	Access Services	51
3.2.	Technical toolkit	51
<b>4.</b>	<b>Business Study</b>	<b>52</b>
4.1.	Study the Carflow Business	52
4.2.	Meetings with the business team	52
<b>5.</b>	<b>System Analysis</b>	<b>52</b>
5.1.	Study the BRC and the DLK	52
5.1.1.	The BRC Concept	52
5.1.2.	The DLK (Datalake)	53
5.1.3.	The Data in the BRC and DLK	53
5.2.	Identify and Analyze the Data Sources	53
5.3.	Analyze the Tables and the KPIs	55
5.3.1.	KPIs Analysis	55
<b>6.</b>	<b>Solution Design</b>	<b>57</b>
6.1.	Data Validation	57
6.2.	Data Mapping	57
6.2.1.	CSF - Carflow Mapping	58
6.2.2.	EProg-Carflow Mapping	59
6.2.3.	Madax – Carflow Mapping	60
6.2.4.	iCube – Carflow Mapping	61
<b>7.</b>	<b>The Data Model</b>	<b>62</b>
<b>8.</b>	<b>Solution Implementation</b>	<b>63</b>

8.1.	Building the Data Pipelines	63
8.1.1.	iCube Pipelines	64
8.1.2.	Eprogramme and Madax pipelines	65
8.1.3.	CSF pipelines	65
8.2.	Scheduling	66
<b>9.</b>	<b>Conclusion</b>	<b>66</b>
<b>Chapter Five: Implementation and Results</b>		<b>67</b>
<b>1.</b>	<b>Introduction</b>	<b>68</b>
<b>2.</b>	<b>Tools and Softwares</b>	<b>68</b>
<b>3.</b>	<b>iCube Pipelines and Automation jobs</b>	<b>72</b>
3.1.	Pipeline code	72
3.2.	Shell Script	76
3.3.	Airflow dags	76
3.4.	The Result	78
<b>4.</b>	<b>EProg Pipelines and Automation jobs</b>	<b>80</b>
4.1.	SQL Query	80
4.2.	Pipeline code	82
4.3.	Shell Script	85
4.4.	Airflow dags	85
4.5.	The Result	86
<b>5.</b>	<b>Madax Pipelines and Automation jobs</b>	<b>87</b>
5.1.	SQL Query	87
5.2.	Pipeline code	89
5.3.	Shell Script	90
5.4.	Airflow dags	91
5.5.	The Result	92
<b>6.</b>	<b>Conclusion</b>	<b>92</b>
<b>General Conclusion</b>		<b>93</b>
<b>References</b>		<b>94</b>

## Chapter One: Project Context



# 1. Introduction

This chapter provides a general presentation and overview of the project environment.

In the first part we are going to present the host company, its Carflow entity and their missions, on the second part of it we will present the project, expose the problem to be solved and the objectives and the project planification, and finally a conclusion.

## 2. Company Presentation

### 2.1. General introduction



*Figure 1: Logo of Stellantis*

Stellantis N.V. is a multinational automotive manufacturing corporation formed in 2021 based on a 50-50 cross-border merger between the Italian-American conglomerate Fiat Chrysler Automobiles (FCA) and the French PSA Group. The company is headquartered in Amsterdam. In terms of global vehicle sales in 2021, Stellantis was the world's fifth largest automaker behind Toyota, Volkswagen, Hyundai, and General Motors.

The primary listings for the company's stock are on Milan's Borsa Italiana and Euronext Paris. The principal activity of Stellantis is the design, development, manufacture, and sale of automobiles bearing its 16 brands of Abarth, Alfa Romeo, Chrysler, Citroën, Dodge, DS, Fiat, Fiat Professional, Jeep, Lancia, Maserati, Mopar, Opel, Peugeot, Ram, and Vauxhall. At the time of the merger, Stellantis had approximately 300,000 employees, a presence in more than 130 countries with manufacturing facilities in 30 countries.<sup>3</sup>

---

<sup>3</sup> Smith, E. (2021). *World's fourth-largest carmaker rallies on first day of trade after \$52 billion merger*. [online] CNBC. Available at: <https://www.cnbc.com/2021/01/18/stellantis-rallies-on-first-day-of-trade-after-52-billion-merger.html> .

### 2.1.1. History and highlights of the Group Stellantis



Figure 2: Logo of Fiat Chrysler Automobiles

Fiat Chrysler Automobiles N.V. is an Italian and American multinational corporation and is the world's eighth-largest automaker. The group was established in October 2014 by merging Fiat and Chrysler into a new holding company. Fiat Chrysler Automobiles' main headquarters are in the Netherlands, and the financial headquarters are in London for tax purposes. The holding company is listed on the New York Stock Exchange and Borsa Italiana in Milan. Exor N.V., an Italian investment group controlled by the Agnelli family, owns 29.19% of FCA and controls 44.31% through a loyalty voting mechanism. FCA's mass-market brands operate through two principal subsidiaries: FCA Italy (previously Fiat Group Automobiles SpA) and FCA US (formerly Chrysler Group LLC). The company's portfolio includes Abarth, Alfa Romeo, Chrysler, Dodge, Fiat, Fiat Professional, Jeep, Lancia, Maserati, and Ram Trucks. Ferrari was spun off from the group in 2016. Today, FCA operates in four global markets: NAFTA, LATAM, APAC, and EMEA. FCA also owns industrial subsidiaries Comau, Magneti Marelli, Mopar, and Teksid<sup>4</sup>.

Groupe PSA, legally known as Peugeot S.A., was a French multinational manufacturer of automobiles and motorcycles. It was founded on April 9, 1976, when Peugeot S.A. acquired



Figure 3: Logo of PSA (Peugeot S.A)

---

<sup>4</sup> Scherfner, E., Смирнов, Д., Nisay, A.A., Gayle, A.T., Otieno, W., Павлова, К., Alaimo, K.-K., Tramm, K., Jhaveri, N. and Gustafson, M. (n.d.). *Stellantis - Wiki*. [online] Golden. Available at: <https://golden.com/wiki/Stellantis-JY9E5E> [Accessed 6 Jul. 2022].

Citroën and created the PSA Group (Peugeot Société Anonyme). The group was formerly known as PSA Peugeot Citroën from 1991 to 2016<sup>5</sup>.

In July of 2020, FCA announced it would be merging with Peugeot S.A. (PSA Groupe) and formally changing its name to Stellantis. The merger resulted in Stellantis becoming the fourth-largest automaker in the world. It was motivated by both automakers, FCA and PSA, to expand consumer bases, develop better electric vehicle technologies, and build more electric vehicles. Stellantis unveiled its new logo in November of 2020 to reflect the company's new name more closely, which is derived from the Latin word "Stello," meaning "to brighten with stars." The names and logos of individual brands like Jeep, Fiat, Dodge, Ram, etc., will remain unchanged, while newly developed vehicle brands will fall under Stellantis<sup>6</sup>.

#### The Timeline:

- 1810: The Peugeot company began working in the metal industry.
- 1890: The first Peugeot-branded gasoline car.
- 1919: Launch of Type A, the first Citroën car.
- June 6, 1925: Chrysler is founded by Walter Chrysler in Detroit, Michigan.
- 1976: Creation of the PSA Peugeot Citroën Group.
- 1978: PSA Peugeot Citroën takes control of Chrysler Europe.
- 1998: Chrysler changes its name to DaimlerChrysler after the Daimler-Benz merger.
- 1978: PSA Peugeot Citroën takes control of Chrysler Europe.
- 2000: Presentation and launch of the particulate filter system (DPF).
- 2014: DS officially becomes the Group's third brand.
- January 21, 2014: Chrysler is acquired by Fiat and is reorganized as Fiat Chrysler Automobiles.
- 2015: Agreement with the Kingdom of Morocco to build a plant in Kenitra.
- 2016 : PSA Peugeot Citroën becomes the PSA Group.

---





<sup>5</sup> chcom (2021). *Stellantis N.V.* [online] CompaniesHistory.com - The largest companies and brands in the world. Available at: <https://www.companieshistory.com/stellantis-n-v/> [Accessed 6 Jul. 2022].

<sup>6</sup> Howard, P.W. (n.d.). *FCA to change name to Stellantis after merger with PSA in 2021.* [online] Detroit Free Press. Available at: <https://eu.freep.com/story/money/cars/chrysler/2020/07/15/stellantis-fca-fiat-chrysler-peugeot-sa/5444259002/> [Accessed 6 Jul. 2022].

- 2017: Acquisition of the Opel and Vauxhall automotive brands, with the purchase of the European division of General Motors
- 2019: PSA and Fiat Chrysler (Fiat Chrysler) proposed a 50/50 merger plan to form the world's fourth-largest automotive group.
- July 31, 2020: Fiat Chrysler is named Stellantis after the merger with PSA is finally complete<sup>7</sup>.
- January 2021: French carmaker group PSA and Italian-American company Fiat Chrysler Automobiles announced a merger agreement, thus creating the world's fourth-largest car company— Stellantis<sup>8</sup>.
- May 18, 2021: Stellantis and Foxconn team up to make cars more connected<sup>9</sup>.

### 2.1.2. The Group's brands

Table 1: Stellantis Brands

Origin	Brand	Established	Brand CEO
 <b>Italy</b>	Abarth	1949	Olivier François
 <b>Italy</b>	Alfa Romeo	1910	Jean-Philippe Imparato
 <b>United States</b>	Chrysler	1925	Christine Feuell
 <b>France</b>	Citroën	1919	Vincent Cobée

<sup>7</sup> www.nowcar.com. (n.d.). *NowCar | Fiat Chrysler Is Named Stellantis After Merger Is Finally Complete*. [online] Available at: <https://www.nowcar.com/blog/archive/fiat-chrysler-is-named-stellantis-after-merger-is-finally-complete/> [Accessed 6 Jul. 2022].

<sup>8</sup> www.leadersleague.com. (n.d.). *Peugeot and Fiat Chrysler Create Fourth Largest Automaker*. [online] Available at: <https://www.leadersleague.com/en/news/peugeot-and-fiat-chrysler-create-fourth-largest-automaker> [Accessed 6 Jul. 2022].

<sup>9</sup> AP NEWS. (2021). *Stellantis, Foxconn team up to make cars more connected*. [online] Available at: <https://apnews.com/article/europe-technology-business-3b076c99f3a70ee9bff1b423a0803b1a> [Accessed 6 Jul. 2022].














Origin	Brand	Established	Brand CEO
 United States	Dodge	1900	Timothy Kuniskis
 France	DS Automobiles	2014	Béatrice Foucher
 Italy	Fiat	1899	Olivier François
 Italy	Fiat Professional	2007	
 United States	Jeep[note 2]	1943	Christian Meunier
 Italy	Lancia	1906	Luca Napolitano
 Italy	Maserati	1914	Davide Grasso
 United States	Mopar	1937	
 Germany	Opel	1862	Uwe Hochgeschurtz
 France	Peugeot	1810	Linda Jackson
 United States	Ram	2010	Michael Koval
 United Kingdom	Vauxhall	1857 [42]	Uwe Hochgeschurtz



Figure 4: Image for Stellantis brands

## 2.2. The group's sectors of activity

The group's activity is essentially organized around 4 divisions:

- Sales of passenger cars and commercial vehicles: Abarth, Alfa Romeo, Chrysler, Dodge, Fiat, Peugeot, Citroën, Opel, Vauxhall, Fiat Professional, Jeep, Lancia, Ram, and SRT brands.
- Sale of luxury vehicles: Maserati and DS Automobiles brands.
- Sale of automotive equipment: interior systems, car seats, car exteriors, emission control systems, etc.
- Other activities: sales financing services (purchase, rental, leasing, etc.), after-sales services, etc<sup>10</sup>.

## 2.3. The group 2021 sales

In 2021, Stellantis launched more than 10 new models, including the Citroën C4, Fiat Pulse, DS 4, Jeep Grand Cherokee, Wagoneer, Maserati MC20, Opel Mokka, Opel Rocks-e, and Peugeot 308. The Company accelerated its low emission vehicles (LEV) commercial momentum leveraging the portfolio of 34 LEV models in the market including hydrogen fuel cell medium vans. Global LEV sales reached 388,000 units, up 160% year-on-year with a number one position for battery electric van sales in EU30. Stellantis confirmed its strong position in the global commercial vehicles market with leadership in both EU30 and South America markets and achieved its highest ever worldwide pickup sales with approximately 1 million vehicles sold.

- ✓ In North America, the Jeep Wrangler 4xe was the bestselling plug-in hybrid electric vehicle in U.S. retail for 2021.
- ✓ In South America, Stellantis was the market leader in 2021 with a 22.9% share and was also the leader in commercial vehicles with a 30.9% market share.

---

<sup>10</sup> www.abcbourse.com. (n.d.). *Stellantis, secteur d'activité en bourse et sociétés comparables*. [online] Available at: <https://www.abcbourse.com/marches/secteur/STLAp> [Accessed 6 Jul. 2022].

- ✓ In Enlarged Europe, Stellantis was the EU30 market leader in commercial vehicles with a 33.7% market share for 2021. The Peugeot 208 was the number one selling vehicle in the EU30 and 2008 was number one in the EU30 B-SUV segment for 2021.
- ✓ In the Middle East & Africa, consolidated shipments were up 6%, while market share grew in most major markets year-on-year.
- ✓ In India & Asia Pacific, the Company is preparing to launch the all-new Citroën C3, developed and produced in India.
- ✓ In China, Dongfeng Peugeot Citroën Automobile Co. Ltd (DPCA), more than doubled its annual sales volume of 2020 with 100,000 units sold and Stellantis became the fourth largest Independent After Market (IAM) parts distributor in China with sales growth of approximately 30% year on year
- ✓ Maserati's global market share grew to 2.4%, with North America and China's market share at 2.9% and 2.7%, respectively, for 2021<sup>11</sup>.

### 3. Stellantis in Morocco

#### 3.1. Stellantis Plant in Kenitra

The Stellantis Kenitra plant is a Moroccan car plant belonging to the French-Italian-American automobile manufacturing group Stellantis. It is in Ameer Seflia, Kénitra Province. The plant started its operations in June 2019 with an annual output of approximately 200,000 cars.

The plant is designed by Still Industrial and is considered one of the most important industrial plants in Morocco<sup>12</sup>.

The project began with an agreement signed between Stellantis Group and the Kingdom of Morocco on June 19, 2015. Four years later, this event also marked the deployment of the entire Stellantis Group ecosystem in Morocco, just as it has been implemented in other strategic areas of the group. Its decision-making center is in Casablanca, covering R&D centers in Africa and the Middle East. And since then, has created a factory with the best standards of the company.

---

<sup>11</sup> [www.stellantis.com](https://www.stellantis.com/en/news/press-releases/2022/february/full-year-2021-results). (n.d.). *Full Year 2021 Results / Stellantis*. [online] Available at: <https://www.stellantis.com/en/news/press-releases/2022/february/full-year-2021-results>.

<sup>12</sup> [JeuneAfrique.com](https://www.jeuneafrique.com/555292/economie/maroc-psa-debutera-sa-production-le-2-juillet-dans-un-secteur-automobile-en-plein-boom/). (n.d.). *Maroc : PSA débutera sa production le 2 juillet dans un secteur automobile en plein boom – Jeune Afrique*. [online] Available at: <https://www.jeuneafrique.com/555292/economie/maroc-psa-debutera-sa-production-le-2-juillet-dans-un-secteur-automobile-en-plein-boom/> [Accessed 14 Jul. 2022].

Stellantis Group is the only car manufacturer to cover the entire value chain in Africa. The ecosystem has integrated a network of 62 local suppliers in Morocco and has created 27 new supplier sites to provide products for the Kenitra site.

The Kenitra plant showcases the group's excellence and proprietary technology. Automotive production based on the CMP platform begins with the new 208 parallel at the Trnava Slovakia plant.

The performance of the Kenitra site makes it possible to imagine the production of vehicles that meet the expectations of private and professional customers in terms of quality and price.



*Figure 5: Stellantis plant in Kenitra*

## 3.2. Organigram of ICT MEA



Figure 6: ICT MEA Organigram



Figure 7: ICT MEA Organigram

## 4. Carflow entity presentation

### 4.1. Introduction of the entity

The Carflow entity of the Sales and Marketing department is mainly responsible for sales synthesis within the Middle East and Africa region. It must therefore monitor the activity continuously, by comparing the actual to what is planned, and adjusting the forecasts as things progress.

The Middle East and Africa region include 6 zones:

- MAGHREB



- DEPARTMENT OVERSEAS AND ISRAEL
- SUB-SAHARAN AFRICA
- EGYPT AND MASHREQ
- GULF COOPERATION COUNCIL
- PSA TURKEY

Each zone is managed by a zone manager, who has under his direction two zone chiefs, one for the Opel brand and the other for the grouping of PCD brands (Peugeot, Citroën, and DS). Then we find the supply managers' business planners.

## 4.2. Organigram of Carflow

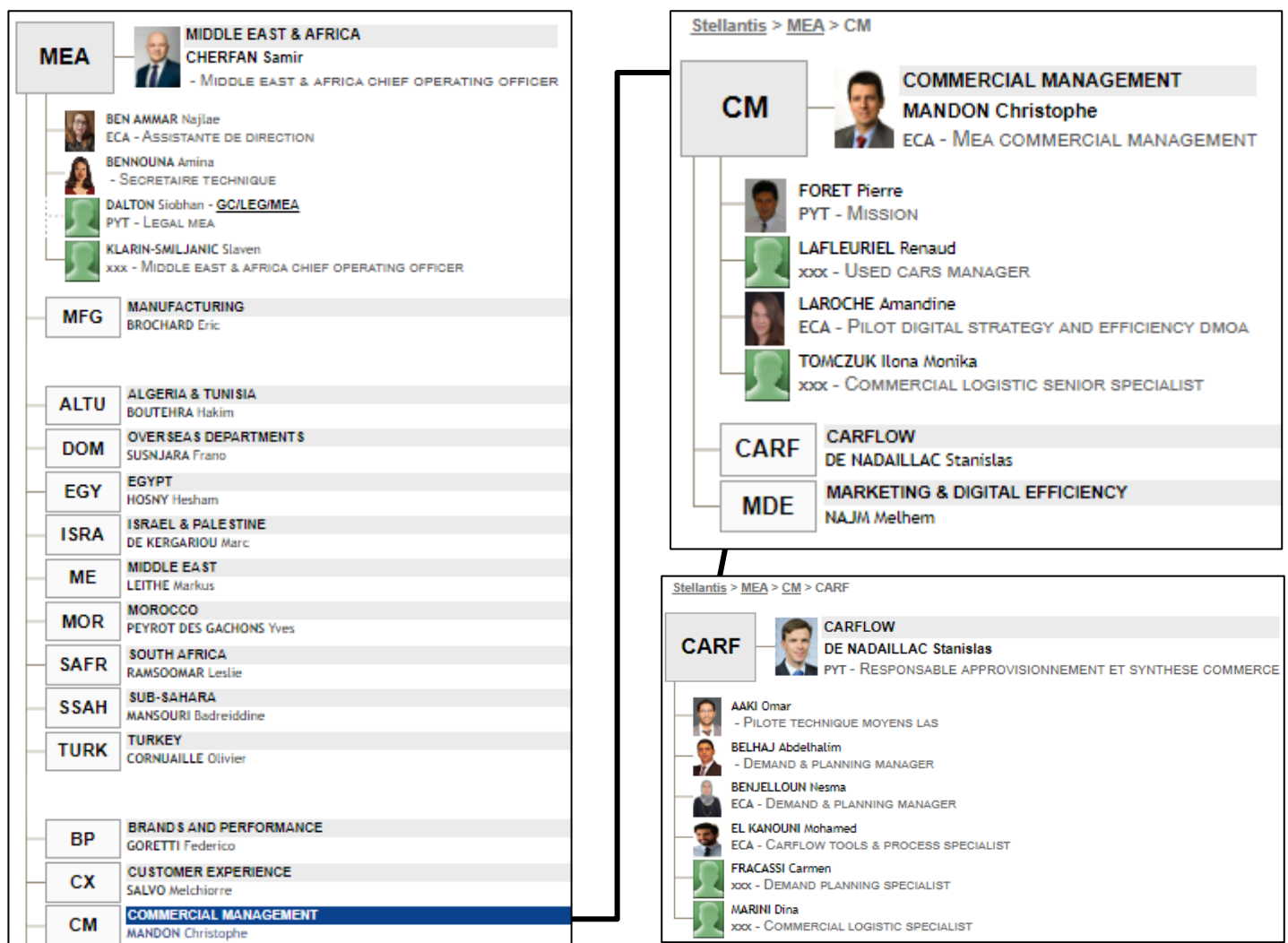


Figure 8: CARFLOW Entity Organigram

## 5. Presentation of the project

### 5.1. Main Problematic

As stated in the previous problematic section, Stellantis has a working Power BI dashboard called “MEA Carflow Dashboard”. This dashboard serves as the decision-making system for the Middle East and Africa - DMOA branch, which will allow users and business analysts to have an in-depth view of the overall operation of CARFLOW. The problem with this Business Intelligence solution is that the process of extraction, transformation, and loading is 100% manual, which a lot of energy and precious time.

### 5.2. Objectives

The main objective of this project is to create data pipelines to serve as the ETL workflows between the root source and our data warehouse<sup>13</sup> which is the Oracle Exadata, it is also referred to as the “Datalab”.

### 5.3. The Method of implementation

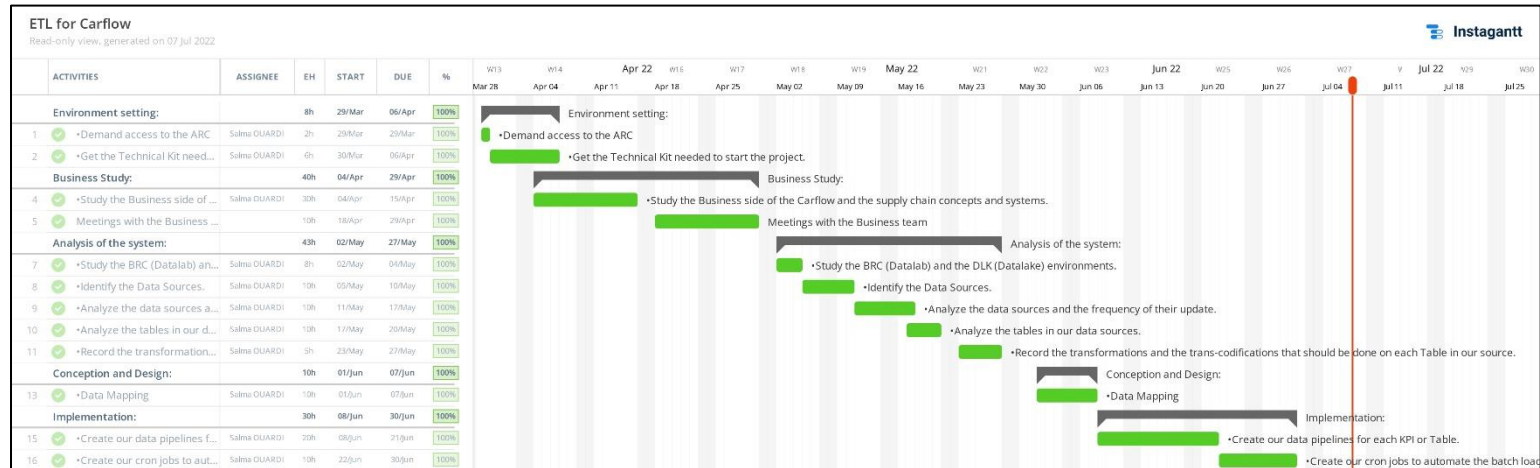
We can obtain our objective by following this approach:

- Study the Business side of Carflow and the supply chain concepts and systems.
- Demand access to the ARC which is the platform dedicated to Data Science.
- Get the Technical Kit needed to start the project.
- Study the BRC (Datalab) and the DLK (Datalake) environments.
- Identify the Data Sources.
- Analyze the data sources and the frequency of their update.
- Analyze the tables in our data sources.
- Record the transformations and the trans-codifications that should be done on each Table in our source.
- Data Mapping between the source tables and the destination tables in our Datalab.
- Create our data pipelines for each KPI or Table.
- Create our cron jobs to automate the batch loads.

---

<sup>13</sup> For the rest of this document the « Datalab » is our Datawarehouse.

## 5.4. Project Planning



## 6. Conclusion

In this chapter, we have presented the general context of the project. In the first phase, we presented the host company, the scope of the end-of-study project, then we elaborated on the objectives, the method and procedure to be followed for the realization of the project, and finally the schedule established for our project.

In the next chapter, we will detail the theoretical concepts used to meet the needs of the host organization.



## Chapter Two: Literature Review



## 1. Introduction

In this chapter, we will discuss the theoretical background where we will present and define the different concepts be it business or technical ones related to the project, to have a clear understanding of the rest.

## 2. Business Concepts

**Budget:** The zone begins the fiscal construction exercise in several stages:

- H1 in June, for Year A (based on achieving at the end of May) and A+1
- H2 in September
- H3 in November

This consists in preparing the forecasts of volumes, in terms of production, invoices, sales, and inventory, in the detailed version and month.

Thus, the annual budget Hx is built and will serve as a reference for the rest. This annual budget will be re-adjusted on two occasions, one in June and the other in November, and will give way to Ax versions of the budget (A1 in February/March, A2 in June, and A3 in November).

**Ax:** annual financial re-forecast (usually 3 per year: realized in late February + FY, realized in late May + FY, real in late August +FY) that relies on the data set production, market, sales, invoices of the completed cycle.

**Invoices:** what Stellantis charges to importers/representatives Stellantis (B to B)

**Deliveries:** what importers charge to final customers (B to C)

### **Programming Cycles:**

Each month, Stellantis begins a new cycle (CPxx), which is spread over 5 rolling months, and consists of establishing allocations and accommodating the estimated budget (in terms of production, invoices, sales, and stock).

Two variants of the programming cycle can be distinguished:

- CP Importer
- Central CP

The “Cycle Programme” is made up of periods over which is defined the production demand and programs. There are 11 CP in a year (no cycle in August).

The importers' programming cycle is triggered by importers who submit their production requests on iCube (importers' version). The central receives requests and returns allowances to importers after study and validation.

The production allocations of  $M + 1$ ,  $M + 2$ ,  $M + 3$  and  $M + 4$  become the new reference, or "planned" for the next cycle. The following management rule is therefore established:

"The allocation over a current cycle will become the forecast for the next cycle"

The first month of production is called a firm month because its allocation is confirmed. At each CP, two exercises are carried out to readjust the commercial forecasts with a redefinition of invoices and sales.

- ✓ 3 days before the end of M (current month): Preref CP0x based on the CP0x allocations, revision of the objectives for month M, the definition of the objectives  $M + 1$ ,  $M + 2$ , revision of the total invoices/sales, and market
- ✓ 2nd working day of M: Ref CP0x, the input of objectives for month M, definition of M objectives + 1 revision of the total volume invoices, retails, market.

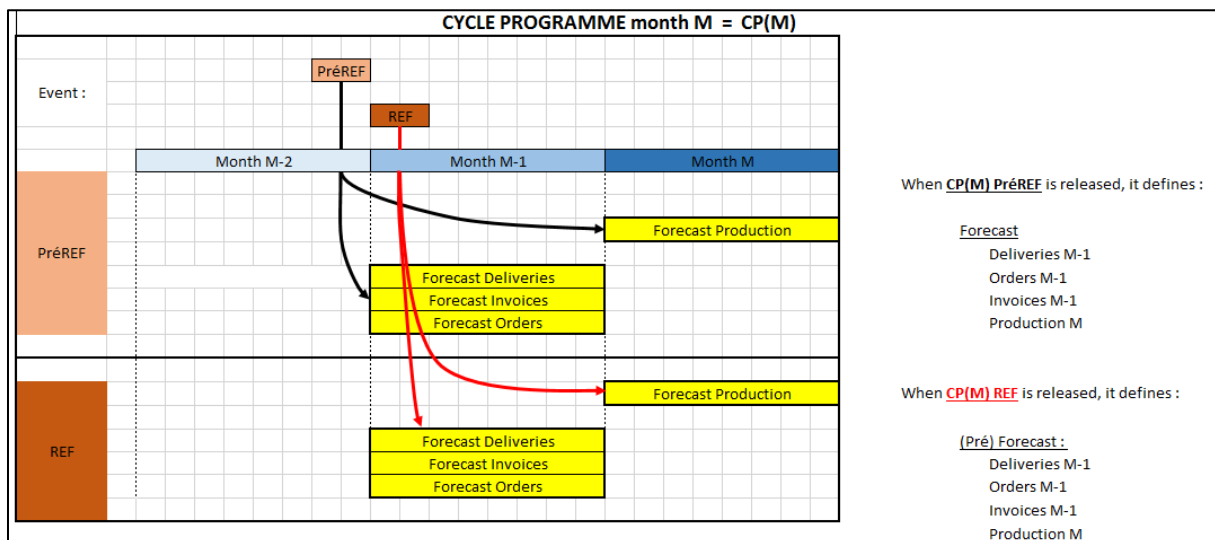


Figure 10: Cycle Programme Explanation

**The (months) in progress:** the months range from the current month, mainly the month M,  $M+1$  (  $M+2$  For IMP cycle) to the month before the cycle.

**Country of marketing:** A country of commercialization is the smallest country. It may coincide with a subsidiary (e.g. Turkey) or a given importer in a geographical country.

Information for production requests is received by PCOM mesh, but all our reports are in PPROG mesh

**Program Country:** may group one or more countries of marketing.

If the PPROG contains a single PCOM, the binding is said to be of type 1:1

If the PPROG contains several PCOMs, the binding is said to be of type 1:N

## 3. Technical Concepts

### 3.1. Business Intelligence

Business Intelligence (or BI) is the set of technologies, tools, and methods that enable better use of data in companies. Business Intelligence is used to understand one's business and environment to make more effective decisions.

Technology plays a key role in Business Intelligence. And for good reason: IT tools allow for better centralization, presentation, and sharing of information to make it an aid to decision making.

The first use of the term Business Intelligence dates back to the 1960s: it referred to the way information was shared between departments and organizations. But modern BI takes on its full meaning with the rise of new technologies and the data revolution: companies finally can collect a new volume of data. Data that can help them understand their market, anticipate problems, and ultimately be more productive.

The only problem is that they still need to learn how to take advantage of this data, and decipher it to use it at the right time and in the right place. This is where business intelligence comes in. We talk a lot about the importance of data, and the need to collect and manage it. But raw data

is not very useful in companies. We need to make it intelligible and transform it into concrete decisions. Without business intelligence, we can't take advantage of data.<sup>14</sup>

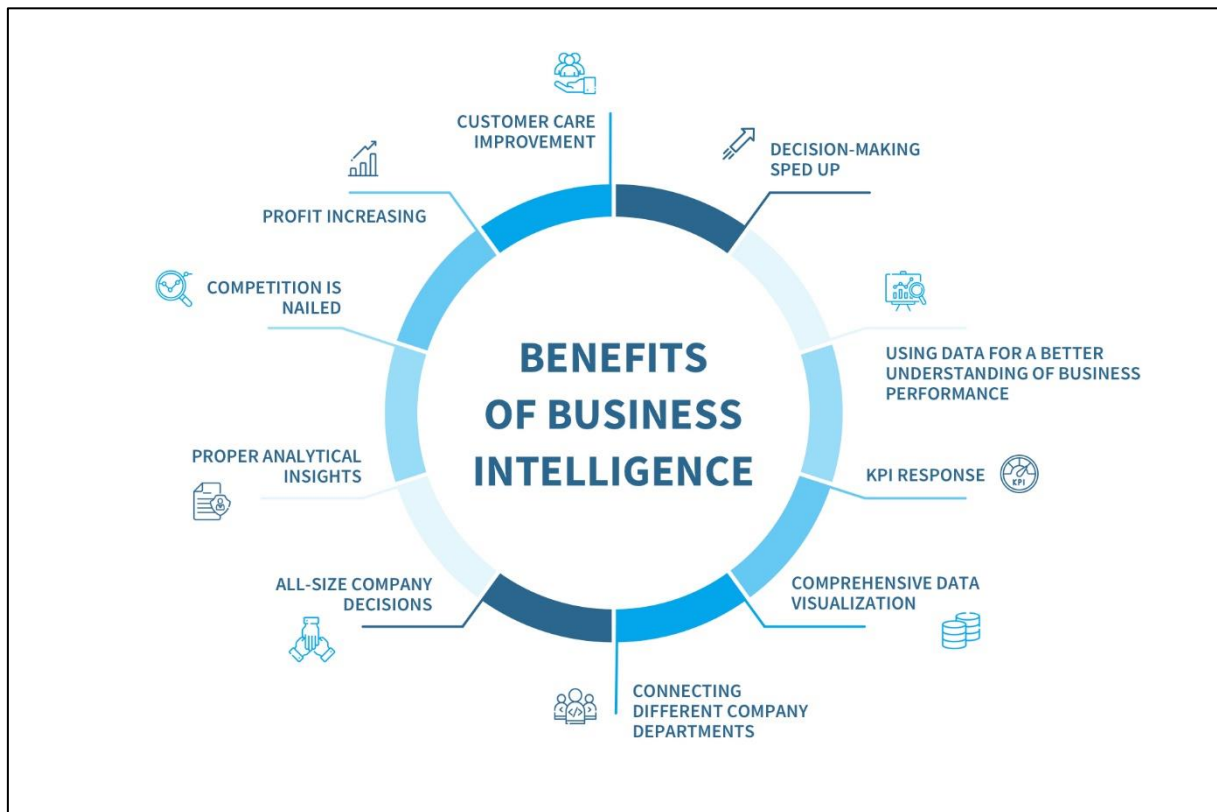


Figure 11: The Benefits of Business Intelligence

### 3.2. Data Warehouse

A data warehouse is a relational database that is designed for data queries and analysis, decision making, and business intelligence activities rather than for transaction processing or other traditional database uses.

The information stored in the data warehouse is historical, providing an overview of the various transactions that have taken place over time. Redundant data is often included in data warehouses to provide users with multiple views of the information. For this reason, the data stored in the Warehouse is often aggregated to allow users to access it more easily.

<sup>14</sup> Cegid. (n.d.). *Business Intelligence*. [online] Available at: <https://www.cegid.com/fr/glossaire/bi-business-intelligence/> [Accessed 7 Jul. 2022].

In addition to a relational database, a data warehouse environment includes data extraction, transformation, and loading (ETL) tool. There is also an analytical processing engine (OLAP), customer analysis tools, and other applications to manage the processing of collected data.

One of the main characteristics of a data warehouse is that the information is classified by subject (customers, products, etc.). What defines a data warehouse is the type of data it contains and the people who use it.<sup>15</sup>

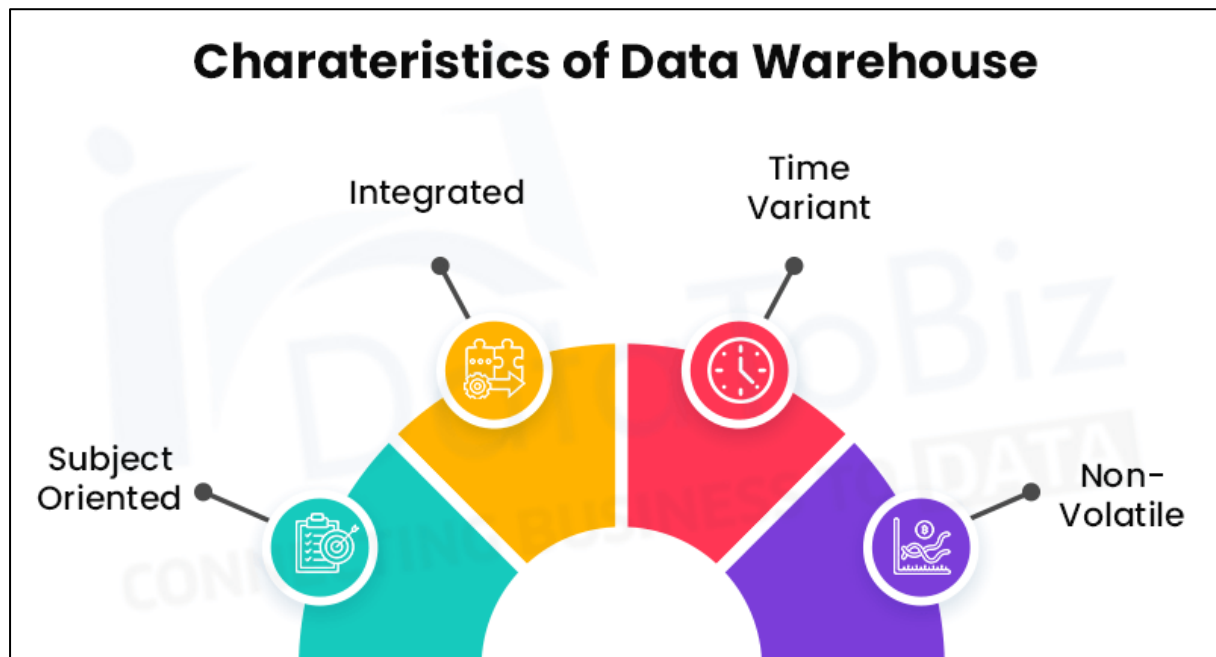


Figure 12: The Characteristics of Data Warehouse

### 3.3. ETL (Extract, Transform and Load)

The term Extract-Transform-Load, better known as ETL, refers to an IT process developed in the 1970s when large companies began to aggregate and store large volumes of disparate data from multiple sources. Since then, ETL has grown along with data warehouses to become an essential process as the amount of data processed worldwide increases.

ETL process carries out the corresponding process, i.e. it extracts raw data from a database (Extract), and restructures it (Transform) into a format suitable for the data warehouse to which it transfers the data in fine (Load).

<sup>15</sup> L, B. (2018). *Data Warehouse (entrepôt de données) définition : qu'est-ce que c'est ?* [online] LeBigData.fr. Available at: <https://www.lebigdata.fr/data-warehouse-entrepot-donnees-definition>.

It is therefore the fundamental link in data management, operating at the same time with data sources, data warehouses, and data lakes and without which the data collected would be unusable.

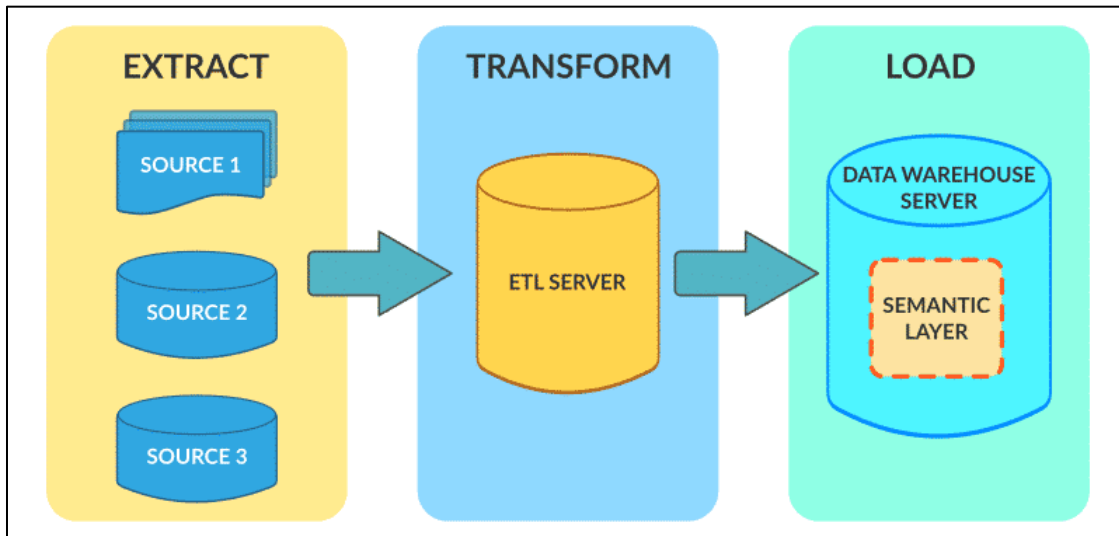


Figure 13: ETL Process

- **Extraction:** The goal of ETL is to produce clean, easily accessible data that can be effectively used by analytics, business intelligence, and/or business operations. Raw data can be extracted from various sources, in particular:
  - Existing databases.
  - Activity logs (network traffic, error reports, etc.).
  - Application behavior, performance, and anomalies.
  - Security events.

The extracted data is sometimes stored in a location such as a data lake or data warehouse.<sup>16</sup>

- **Transformation:** Transformation is the key step in the ETL process. After extraction, where raw data is aggregated and stored, it is cleaned and converted to the company's report format. The cleaning process facilitates compliance with internal company standards. These operations, without which the reports would be unusable, are based on predefined rules:
  - Standardization, determines, among other things, the format and storage mode.
  - Deduplication, i.e. the monitoring and deletion of duplicates.

---

<sup>16</sup> Softwares, L. (2014). *What is Web Data Extraction*. [online] Loginworks Softwares Pvt. Ltd. Available at: <https://www.loginworks.com/blogs/209-web-data-extraction/> .

- Verification to monitor anomalies and remove unusable data.
  - Sorting or grouping of data to maximize the efficiency of Datawarehouse queries.<sup>17</sup>
- **Loading:** The last step of the standard ETL process is to load the extracted and transformed data into its new location. In general, data warehouses support two modes for loading data: full loading (all data is loaded) and incremental loading (only the last data is loaded).<sup>18</sup>

### 3.4. Data Pipeline

A data pipeline is a set of steps that moves data from one system to another. Different types of data pipelines perform different operations through the data transit.

Standard data pipelines include:

- **Batch data pipeline:** A batch data pipeline periodically transfers bulk data from source to destination. It's also a common choice for ELT or ETL processing.
- **Streaming data pipeline:** A streaming data pipeline continually flows data from source to destination while translating the data into a receivable format in real-time.

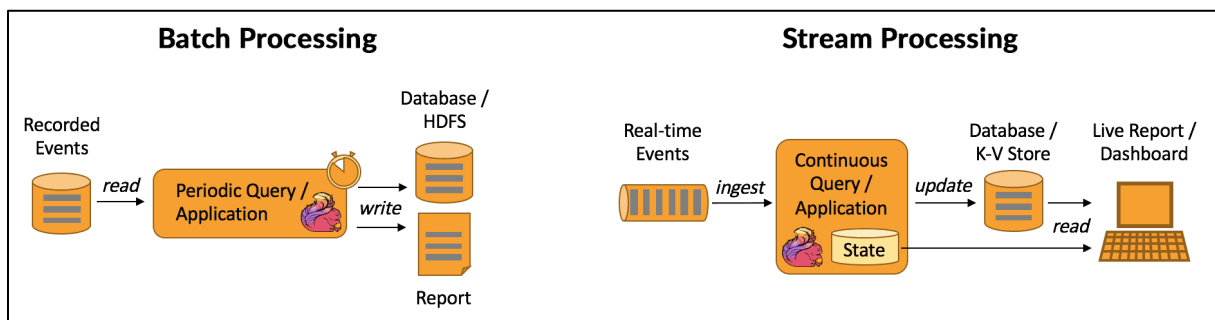


Figure 14: Batch vs Streaming Processes

<sup>17</sup> Garnier, A. (n.d.). *Qu'est-ce que le processus ETL ?* [online] blog.hubspot.fr. Available at: <https://blog.hubspot.fr/marketing/extract-transform-load> [Accessed 7 Jul. 2022].

<sup>18</sup> Softwares, L. (2014). *What is Web Data Extraction*. [online] Loginworks Softwares Pvt. Ltd. Available at: <https://www.loginworks.com/blogs/209-web-data-extraction/>.



### 3.5. ETL vs Data Pipeline

ETL refers to a set of processes that extract data from a system, transform it, and load it into a target system. A data pipeline is a more general term; it refers to any kind of processing that shifts data from one system to another and may or may not transform it.

### 3.6. ETL Pipelines

An ETL pipeline is just a data pipeline that uses ETL strategies to extract, transform, and load data. Here, data is usually extracted from various data sources, such as SQL or NoSQL databases, CRM, CSV files, etc.

The data is then transformed into a staging area whose sole purpose is to prepare the data in a format best suited to the data destination (usually a data warehouse or database).<sup>19</sup>

### 3.7. Airflow Dags

Airflow is a task scheduler with the following main functionalities:

- dependencies between tasks
- automatic retry in case of failure
- manage a job from a web interface (run job or specific task, visualize log)

Airflow is organized around two main principles:

- a **DAG**: a representation of the dependencies between multiple tasks
- an **Operator**: a task defined by an id and a command

Here is an example of a DAG with dependencies between operators (tasks).

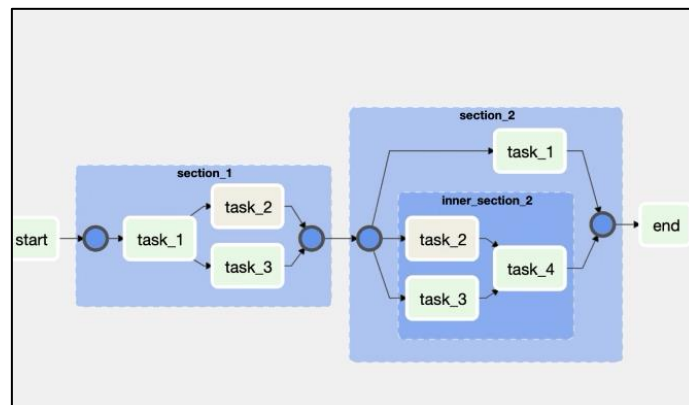


Figure 15: DAG Example

<sup>19</sup> StreamSets. (2022). *The Relationship Between ETL and Data Pipelines*. [online] Available at: <https://streamsets.com/blog/relationship-between-etl-and-data-pipelines/#:~:text=ETL%20refers%20to%20a%20set> [Accessed 7 Jul. 2022].

- A DAG is described by a Python script.
- Here is an example of a DAG with two tasks: t1 and t2 which runs sequentially. This DAG runs every day.

```
with DAG(
    'My DAG',
    description='A simple DAG',
    schedule_interval=timedelta(days=1),
    start_date=days_ago(2),
) as dag:

    t1 = BashOperator(
        task_id='print_date',
        bash_command='date',
    )

    t2 = BashOperator(
        task_id='sleep',
        bash_command='echo Hello there',
        retries=3,
    )

    t1 >> t2
```

Figure 16: Dag code example

- A DAG is defined by:
  - **an id**: the name of the python file without the extension
  - a description
  - **a start date** that corresponds to the first launch
  - **a periodicity** at which it must be launched. In this case, we use the cron syntax to schedule our tasks:

```
# Cron syntax

# minute    hour    day of month    month    day of week    command
# (0-59)    (0-23)    (1-31)    (1-12 or Jan-Dec)    (0-6 or Sun-Sat)

# Execution every day at 10am
# 0 10 * * *
```

Figure 17: Cron Syntax

## 4. Conclusion

In this chapter, we have defined the key theoretical concepts and notions that we will use in the following chapters of the graduation project.

In the next chapter, we will proceed to the analysis of the existing situation and the requirements analysis.

## Chapter Three: Analysis of the existent



## 1. Introduction

After defining the theoretical scope of our project and identifying the different concepts addressed. In this chapter, we will start with an analysis of the existing situation. Indeed, this phase is essential for the conception and design of the project and to fulfill its objectives, which allows us to realize our ETL pipeline.

## 2. Analysis of the existent

### 2.1. The Carflow MEA Dashboards project steps

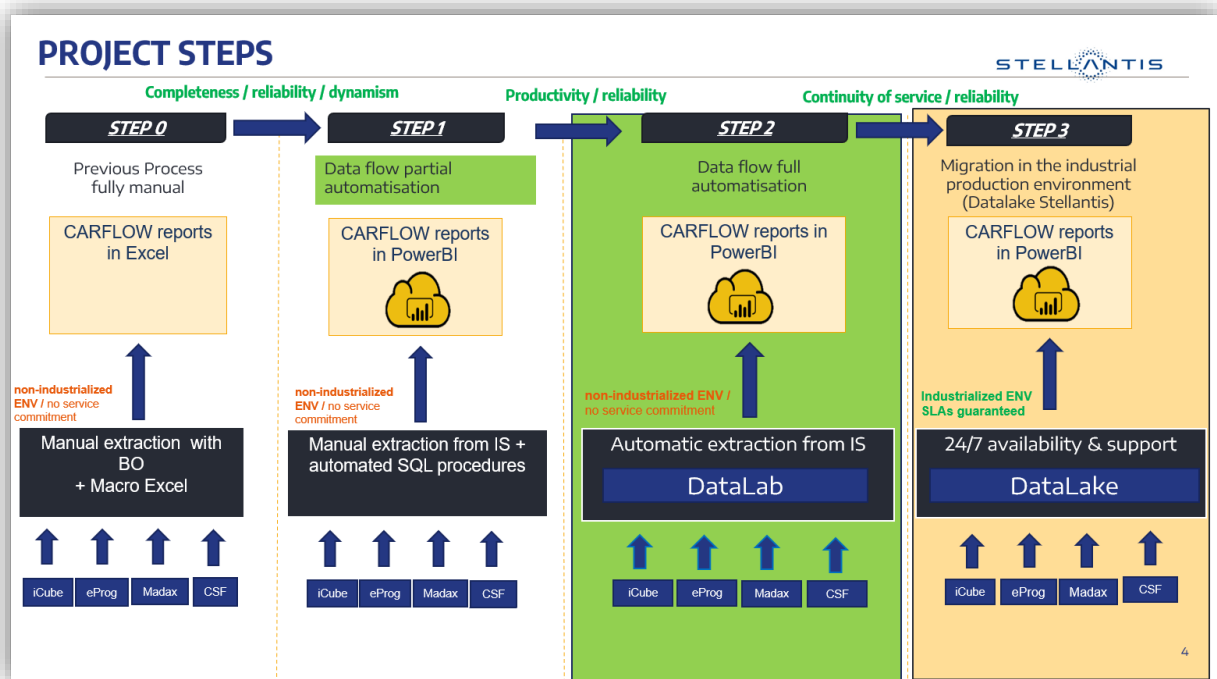


Figure 18: Carflow Project Steps

Stellantis has a BI project called “MEA Carflow Dashboards”, this project has 4 steps and currently they’re in the 3<sup>rd</sup> step of it.

- Step 0: In this step, they created their reports in Microsoft Excel and extracted the data needed from the Business Objects manually (it’s a fully manual process).
- Step 1: In this step, they started using Power BI dashboards and they did the whole extraction and transformation using manual extractions from the source systems via business objects and automated SQL procedures.
- Step 2: This step aims to use Oracle Exadata Datalab as their Datawarehouse and start extracting the data manually from the source systems and create a getaway

between the Datalab and the Power BI dashboards so that the refreshes can be automatic without the manual processes. (Fully Automatic)

→ Step 3: Here we start to do our analytics on the HDFS on premises with the help of Apache Spark and then load our refined, standardized data in the Datalab and the same way as in the previous step, we'll schedule our refreshes through a Gateway.

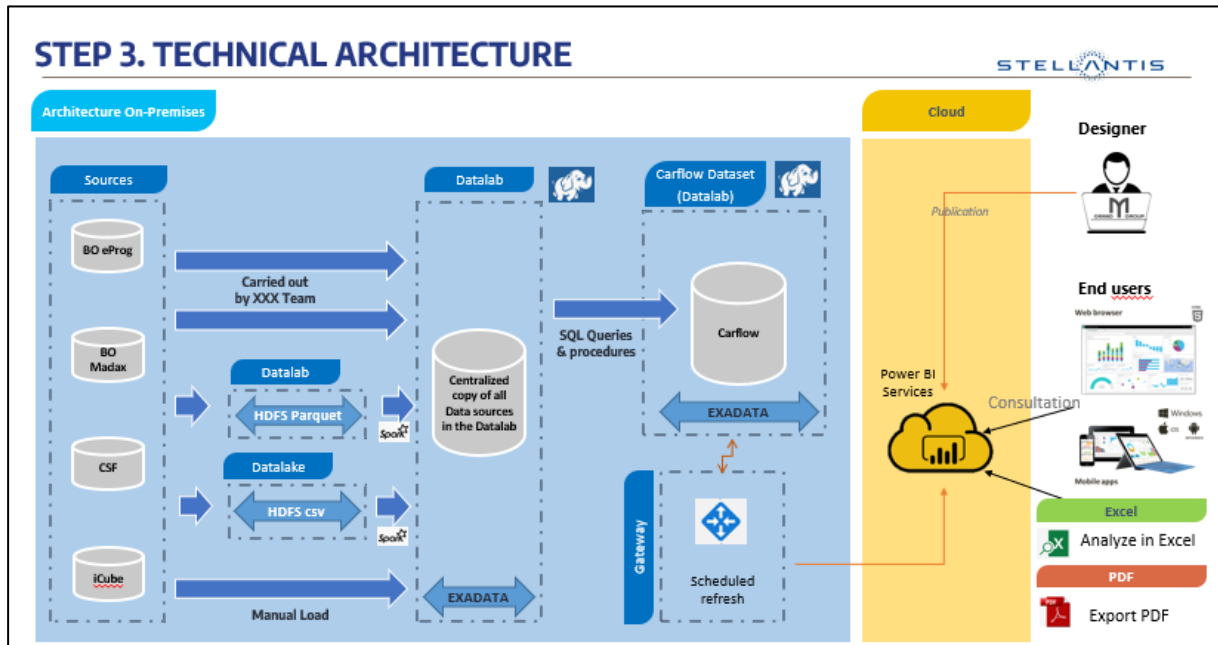


Figure 19: Technical Architecture of Step 3

## 2.2. The Current Solution

Even though the BI team succeeded in implementing half of Step 2 which is creating the Carflow Dataset, the tables, the views, and the procedures in the Datalab, they did not build the most important part of it, which is the automation of the ETL workflows. They kept extracting the data manually from the sources through the business objects in the format of CSV or XLS, sending the extracted files to the employee charged with the refresh process either through an email or uploading them in a Microsoft Teams channel. The person then needs to download the files, do the transformation cited in the manual and upload the data manually to the tables in

the Oracle Exadata Datalab, run the procedures and then refresh the Power BI dashboard. The schema below will help us understand the current solution.

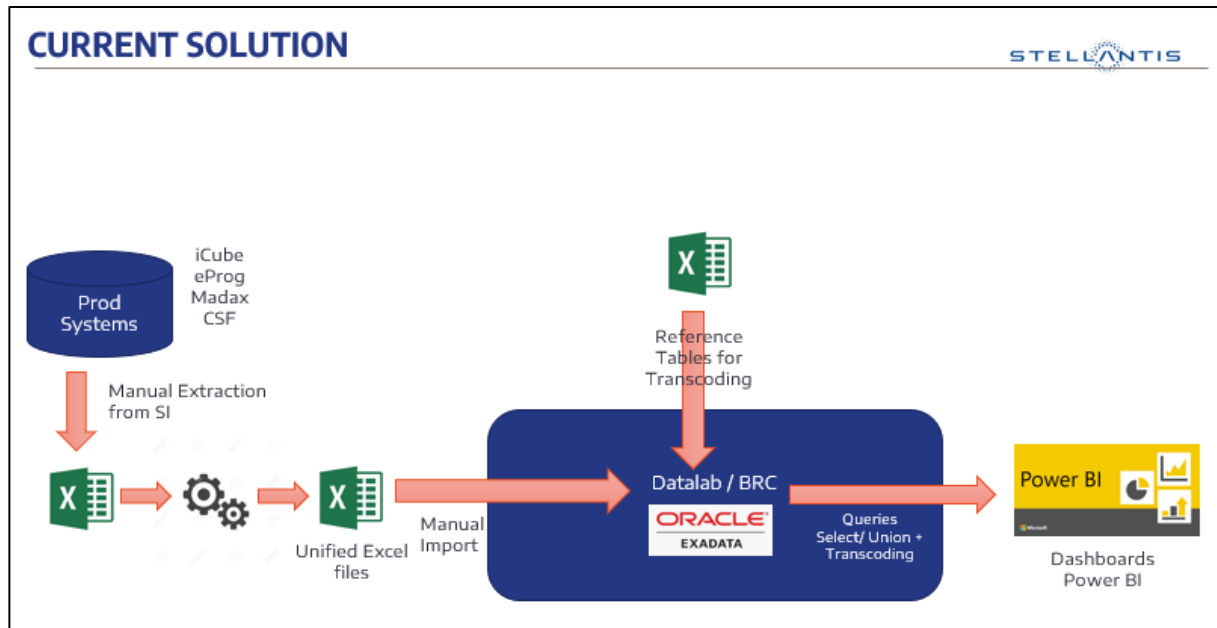


Figure 20: Current Solution Architecture

## 2.3. Problems with the current solution

The main problems with the current solution are:

- ❖ The manual process of the extraction, transformation, and loading represents a high risk of errors as the responsibility for it is a human, and as accurate as they can be, they're going to make an error sooner or later. And this obviously could falsify the calculations and induce the decision makers into false assumptions therefore business loss.
- ❖ The whole manual process is long and repetitive, and it is most likely to be assigned to a worker who parallels additional work to do besides this. This overload, therefore, overworks our employee and waste precious time and energy.
- ❖ The manual process doesn't allow automatic refresh of the dashboard, it must be also manually refreshed, as we can't schedule the refresh through gateways.

### 3. Project Requirements and needs

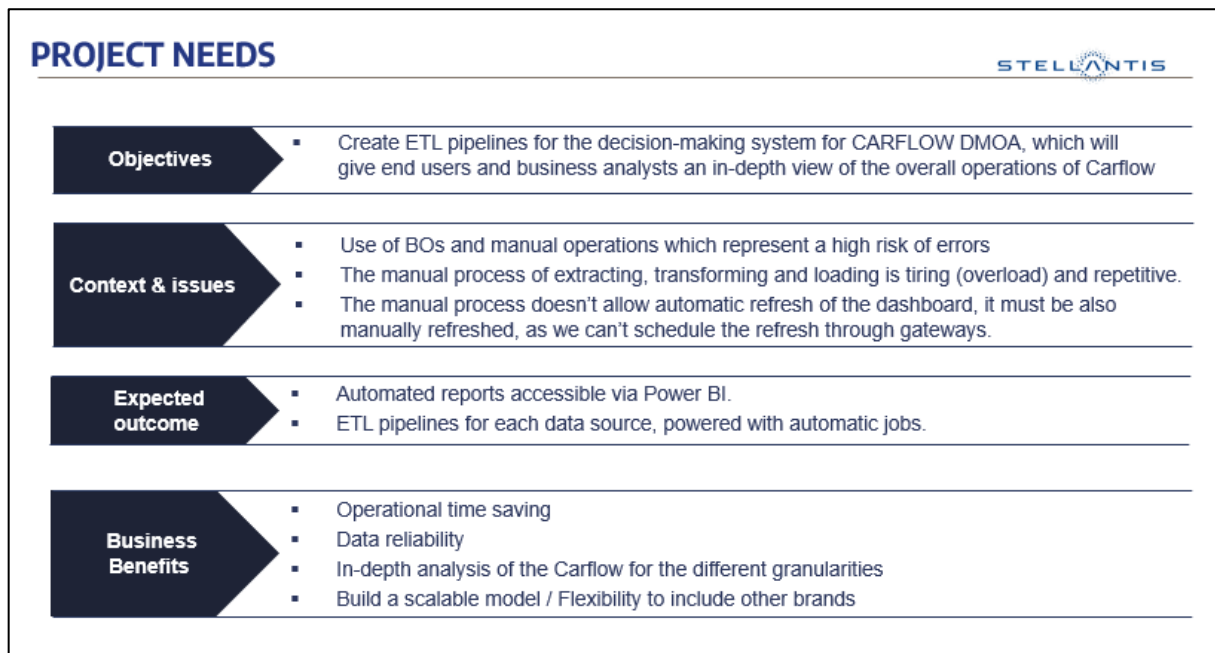


Figure 21: Project Needs

Perimeter:

- All new vehicles
- All Brands: AP, AC, DS et OV
- All DMOA Countries (program country level)

### 4. Conclusion

In this chapter, we were able to have an eagle eye view of the MEA Carflow Dashboards project, describe the different steps of the project, and pinpoint the last achieved step, the current situation, and its problems finally we were able to present the project needs.

In the next chapter, we will delve into the practical aspect of the project and do a conception of our solution.



## Chapter Four: Conception and Design



## 1. Introduction

After having specified the needs, in this chapter, we will define the general conception adapted to the collected needs to frame functionally and technically our project and to detail each stage of the realization of our solution.

## 2. General Conception

The following general conception presents the steps undertaken to give birth to our project.

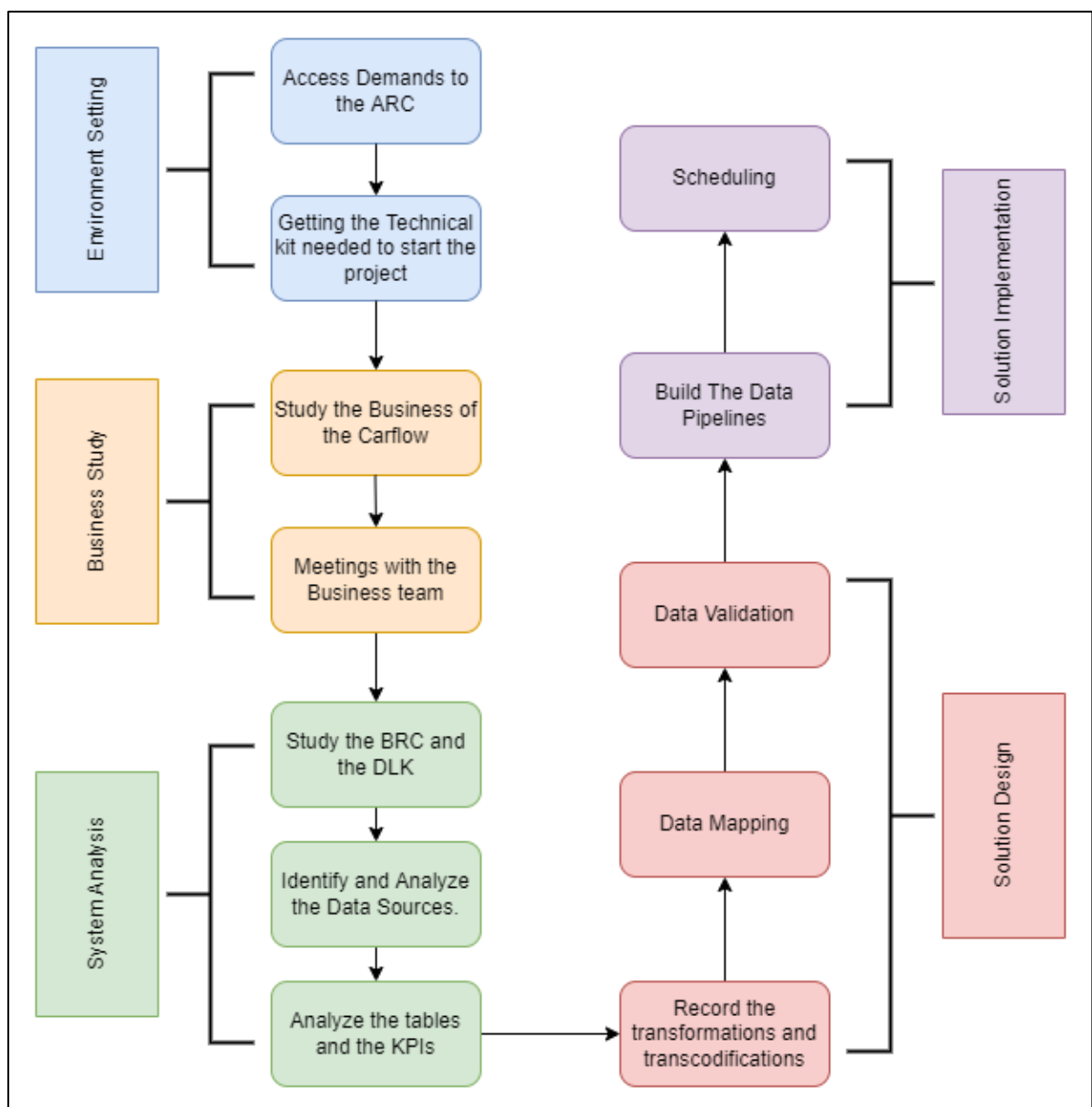


Figure 22: Conception Diagram

Each step plays a vital role in the success of the project. Therefore, in this report, we will focus on all the main parts.

### 3. Environment Setting

#### 3.1. Access Services

The Access Services (“Services d'Accès”= SA) allow users to access Stellantis applications.

To have the access to the Data lake, the Datalab, and the servers I had to follow the steps below:

- ✓ Sign the Commitment letter for Confidentiality & Privacy policy.
- ✓ Join the signed commitment letter in an email. Precise the motivation of the Datalab / Data lake access with the following elements:
  - The function / position.
  - The Direction and department.
  - The reporting line.
  - Description of the project(s) for which we need to access the Datalab / Data lake.
  - The role played for each of the project(s).
- ✓ Make a request to The ASL (by email) for the following Access Services (SAS):
  - BRCU :
    - Datalab technical access
  - BRCH/BRCC
    - Reading all C1 / C2 data of all the BRCs
    - Access to the CIQ (GitHub, TeamCity ...) environments of all the BRCs
  - BRCB: write access for BRC
    - Write to an Oracle table or Hadoop
    - Automate the job when moving the project to the pilot phase
  - BRCA : BRC Administrator
    - CIQ platform management
  - BRCI: write access to C3 data
    - Gives you access in TAM to the technical account BRCxx\_PRDCy

#### 3.2. Technical toolkit

List of the software needed:

- **Putty**: ssh connection to ARC
- **WinSCP**: file transfer between ARC & the local machine
- **VPN-Cisco**: connection to the Stellantis environment from an external network
- **Oracle-SQL-Developer**: querying Oracle or other databases (DB2, SQL server ...) with JDBC drivers

- **PowerBI:** data visualization tool -> take the 64-bit version, but pay attention that you choose the Oracle client version 64-bit (below)
- **Visual Studio Code:** IDE

## 4. Business Study

### 4.1. Study the Carflow Business

During this step, I had to learn the Business basics with which they work in the Supply Chain department, mostly the Carflow entity of the said department.

In this part, I had the chance to discover a lot of Business concepts mainly in the supply chain area, I got to know how Stellantis plan its production streams using the Production cycles I talked about in the business concept's introduction of the Literature Review (*Chapter 3*).

I also learned how each system ( iCube, Madax, CSF, E-Programme) facilitates the Supply Chain process from the Importer's demand of production to the production allocation to the Central production demands to their plants to the Sales (B-B) to the Wholesales (B-C).

### 4.2. Meetings with the business team

The meetings with the Business team had a great impact on my ability to understand the business, the KPIs, and their needs as the Dashboards are mainly directed to them (The Supply Chain Department).

## 5. System Analysis

### 5.1. Study the BRC and the DLK

#### 5.1.1. The BRC Concept

The concept of *BRC* was introduced in April 2018 to standardize how Data Scientists access data.

A BRC (BigData Ressource Classification) is an entity that brings together:

- People belonging to a business management
- The projects carried out for this business direction
- Data Scientists working on these projects
- The data generated by these business entities

### 5.1.2. The DLK (Datalake)

The Datalake is a central directory that collects all of the company's data, based on the Hadoop Framework.

### 5.1.3. The Data in the BRC and DLK

Table 2: Data in BRC and DLK

The BRC	The DLK	The label
BRC00	DLK00	Templates / Test
BRC01	DLK01	Data Connected Vehicles
BRC02	DLK02	Data Test and Validation Vehicles
BRC03	DLK03	Data Sales and Marketing
BRC04	DLK04	Data Referential Vehicle
BRC05	DLK05	Data Manufacturing
BRC06	DLK06	Data Supply Chain
BRC07	DLK07	Data Quality Vehicle
BRC08	DLK08	Data Purchase
BRC09	DLK09	Data RH COM SG
BRC10	DLK10	Data Finance
BRC11	DLK11	Data Centralized OV
BRC12	DLK12	Data After Sales
BRC13	DLK13	Data Stellantis Bank
BRC14	DLK14	Data DDIG
BRC15	DLK15	Open and External Data
BRC16	DLK16	IT for IT

The data for the dashboard (Destination Tables) exists in the BRC03 under the schema BRC03\_CRF0A. And the source data exists in the BRC06 and the DLK06.

## 5.2. Identify and Analyze the Data Sources

### ❖ iCube (Importers Information Interface)

Application allowing:

- Centrally, exchanges with importers, order/production forecasts; the follow-up of importing orders.
- Importers consult their data (customer file, open range), to contribute to the development of forecasts, and to take their orders.

### ❖ eProgramme

As part of the VN&ICP renovation, e-Programme receives from e-Market the version level sales forecasts and establishes a stock sequence based on:

- From the stock of the subsidiary and the network at the beginning of the month,
- Sales made the previous month
- A stock standard determined by the Central for the Country / Family / Subfamily
- Sales forecasts established in e-Market. The result makes it possible to obtain the country's production demand for the cycle which includes 5 months starting at m+2.

Upon receipt of production requests from countries, the Central consolidates the requests, adjusts the consolidated demand according to the commercial policy, and forwards this request to DTI (Manufacturing) for analysis. The demand made by the trade is confronted with the production possibilities expressed by DTI and the arbitration is then made at the DG level (General Demand). Upon receipt of the DG (document framing the production), the Central proceeds to a distribution of the quantities allocated according to the stocks in each country by version

#### ❖ **CSF (Country Sales Forecast)**

CSF “Country Sales Forecast” is the target solution for forecasted Sales Demand by countries.

The implementation and deployment of CSF are ongoing for PCD and OV, to replace current E-Market, E-MarketMix systems, the CAFE toolbox, and the OV GM legacy system.

CSF covers Countries' Sales (Deliveries, Registrations, Orders, Invoices, Mix LCDV14) and CO2 forecasts, Production Demand Simulation.

CSF stores data from:

- The last reference
- The last annual forecast
- The last budget
- The current cycle --> it will become the next reference or budget ...
- Current weekly outlook (forecast for the current month)

#### ❖ **Madax**

MADAX is the monitoring tool of commercial activity for all brands. MADAX feeds BO RVN (BO RESULTAT VN).

VN business activity dashboards for Brands, subsidiaries, and points of sale. Portfolio of orders, sales orders, invoicing, deliveries, ...

The MADAx application processes the monitoring of the distribution of new vehicles in the world's commercial subsidiaries, using dashboards of the VN commercial activity for Peugeot and Citroën.

## 5.3. Analyze the Tables and the KPIs

### 5.3.1. KPIs Analysis

The Carflow Reports will permit to analysis and compare Actuals / Forecasts / Budgets of production demands (importers and central)/ production allocation / Wholesales (or invoices) / retails or (deliveries) / market and market share, for DMOA perimeter. These KPIs can be analyzed according to several axes of analysis.

#### ➤ KPIs Structure

Table 3: KPIs Structure

Type of KPI	Nature of KPI	Axis of analysis
Central production demands	Actual	Brand
Importers' production demands	Forecast / CP	Country
Production allocation	Objectives (REF and	Carline / family
Retails or deliveries	PreREF)	Type of vehicle ("Genre" in French)
Wholesales or invoices	Budget (Hx and Ax)	Cycle
Stock OEM		Date / period
Stock network		
Market		

#### ➤ Type of KPI

Table 4: Type of KPI

<b>Central production demands</b>	The volume of vehicles demanded to be produced from the plants
<b>Importers' production demands</b>	The volume of vehicles demanded from the dealer
<b>Production allocation</b>	The volume of vehicles allocated to the dealer/importers
<b>Retails or deliveries</b>	Volumes of vehicles delivered by Stellantis group to its customers
<b>Wholesales or invoices</b>	Volumes of vehicles invoiced by Stellantis group to its customers
<b>Stock OEM</b>	Order placed by an NSC or dealer without being assigned to a dealer or final customer
<b>Stock network</b>	The stock of the dealer/importer
<b>Market/TIV</b>	Volumes of new vehicles registered by all car manufacturers (not only Stellantis).

#### ➤ Nature of KPI

- KPIs previously listed are analyzed and compared according to several natures:

Table 5: Nature of KPIs

<b>Actual</b>	Volumes done of registrations, deliveries, invoices, ...
<b>Forecast</b>	Monthly objectives defined within « Cycles Programmes »
<b>Budget</b>	Budget volumes are defined within budgetary cycles

➤ **Axis of Analysis**

KPI previously listed are analyzed according to 6 axes:

*Table 6: Axes of Analysis*

<b>Brand</b>	A brand that registered, delivered, invoiced... a vehicle. Stellantis brands for its activity, as well as brands outside Stellantis group concerning market data.
<b>Country</b>	Geographical division. Each country belongs to a region, or a representative group of countries (ex: G5)
<b>Carline / family</b>	Model registered, delivered ... Stellantis models and models from other brands concerning market data
<b>Type of vehicle ("Genre" in French)</b>	Passenger cars (PC) and light commercial vehicles (LCV)
<b>Cycle</b>	CP01 - CP02 ...
<b>Date / period</b>	Year, month, daily (current month), and cumulated (Month to Date, Year to Date, Qx, Hx, ...)

**Country**

The country indicates the geographical data division with the finest possible mesh.

In Stellantis language: PAYS DE COMMERCIALISATION (PCOM) (different from PAYS PROGRAMME (PPROG) – not the same granularity, a PPROG can contain several PCOM). this project will only evolve DMOA countries. All reports will be in PPROG granularity.

**Carline / Family**

In PCD data (from MADAX), it is the family, identified by its LCDV4 code (4 first digit of LCDV). The family permits to distinguish the generations from the same model.

➤ **Update Frequency**

*Table 7: Update Frequency*

Nature of KPI	Frequency of updating
<b>Actual</b>	Four times a Month
<b>CP Objective</b>	Twice a month, REF and PRE REFF
<b>Budget</b>	Twice a year (or three times a year): Hx, Ax



## 6. Solution Design

### 6.1. Data Validation

In this step we organize meetings with the Business teams, to check if the data we found is the data they are looking for.

This step is vital for the whole process, it prevents us from transporting the wrong data and having to redo the whole analysis and mapping for the new valid data.

### 6.2. Data Mapping

Data mapping is critical to the success of many data processing. Mistakes in data mapping can seep into our ETL projects, leading to repeated errors and ultimately inaccurate analysis. Data mapping is the process of matching fields from one database to another. This is the first step in simplifying the ETL workflow.

Data mapping bridges the gap between two systems or data models so that when data moves from one source, it is accurate and available at the destination.

- In the Table below I did a mapping between the Source Tables in the Source Systems and the Destination Tables in the BRC03\_CRFOA.

Table 8: Mapping of Data(Tables)

SI	KPIs	Source Table Name	FACT/DIM	Frequency of update	Source Table Location	Destination Table Name	Destination Table Location
eProg	Central production demands Production allocation	BPG.RBVQTVPN	Fact	At 01:00 every day	BRC06_BPG00.RBVQTVPN	EPS_AC_2, pour la première mâj / EPS_AC_4, pour la 2e & 3e mâj	BRC03_CRFOA.EPS_AC_2/4 BRC03_CRFOA.EPS_OV_2/4 BRC03_CRFOA.EPS_AP_2/4
		BPG.RBVQTCA1	Dimension	At 01:00 every day	BRC06_BPG00.RBVQTCA1		
		BPG.RBVQTPE1	Dimension	At 01:00 every day	BRC06_BPG00.RBVQTPE1		
		BPG.RBVQTF1	Dimension	At 01:00 every day	BRC06_BPG00.RBVQTF1		
		BPG.RBVQTPP	Dimension	At 01:00 every day	BRC06_BPG00.RBVQTPP		
Madax	Stock OEM/Constructeur/NCS Stock Dealer/Network	BDS.RBVQTDNE	Fact	At 00:00 every day	BRC06_BDS00.RBVQTDNE	STOCK_EXCEL STOCK_DEALER	BRC03_CRFOA.STOCK_EXCEL BRC03_CRFOA.STOCK_DEALER
		BDS.RBVQTFLL	Dimension	At 00:00 every day	BRC06_BDS00.RBVQTFLL		
		BDS.RBVQTTFF	Dimension	At 00:00 every day	BRC06_BDS00.RBVQTTFF		
		BDS.RBVQTFAM	Dimension	At 02:00 every day	BRC06_BDS00.RBVQTFAM		
Icube	Impoters Demand	PROREV_202107IMP_ACDS			Extracted from Icube Portal	DP_IMP_AC_T	BRC03_CRFOA.DP_IMP_AC_T
		PROREV_202107IMP_AC - TR					
		PROREV_202107IMP_DS - TR					
		PROREV_202107IMP_AP				DP_IMP_AP_T	BRC03_CRFOA.DP_IMP_AP_T
		PROREV_202107IMP_AP - TR					
		PROREV_202107IMP_OV				DP_IMP_OV_T	BRC03_CRFOA.DP_IMP_OV_T
CSF	Retails or deliveries	BCSQT_CSF_CL_DELIVERY_LOAD			/gpfs/users/dlk06/data/raw/bcs 00/00	RS_CP_PREREF / RS_CP_REF	BRC03_CRFOA.Tables
	Wholesales or invoices	BCSQT_CSF_CL_INVOICE_LOAD			/gpfs/users/dlk06/data/raw/bcs 00/00	WS_CP_PREREF / WS_CP_REF	BRC03_CRFOA.Tables
	Market	BCSQT_CSF_CL_MARKET_LOAD			/gpfs/users/dlk06/data/raw/bcs 00/00	MARKET_PREREF / MARKET_REF	BRC03_CRFOA.Tables

- In the table below I created a matrix to explain where we can find each KPI.

Table 9: Data availability Matrix

	Production						Factures/Invoices/WS				Ventes/Sales/ Deleveries				Stock		Market	
	DP IMP		DP Central		Allocation		Réalisé		Prévision		Réalisé		Prévision				Prévision	
	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL	PCD	OPEL
iCube	x	x																
eProgramme			x	x	x	x												
Madax							x	x			x	x			x	x		
CSF									x	x			x	x			x	x

### 6.2.1. CSF - Carflow Mapping

In the table below we will find the data mapping of the source tables and the destination tables with an exhaustive description of each table and column in our source as well as in our destination. I also included the transformations to be made on the source data of each column, and finally the calculation to be done on the data.

Table 10: CSF-Carflow Mapping

Dataset Carflow - Destination						
Tables	Description	Colonnes	Type de données	Description		
TF_Carflow	Fact table Carflow	Brand_Id	Bigint	surrogate key, linked with teh Brand dimension		
		Country_Id	Bigint	surrogate key, linked with the Country dimension		
		Carline_Id	Bigint	surrogate key, linked with the Carline dimension		
		TYPE_Id	Bigint	surrogate key, linked with the Type dimension		
		MonthYear_Id	Bigint	surrogate key, linked with the Date dimension		
		Cycle_Id	Bigint	surrogate key, linked with the Cycle dimension		
		Measure	varchar(50)	Describes the nature of the measure (Wholesales, market ...)		
		Value	float	The value of the measure, the unit depends on the nature of the measure		
		Inserted_Date	Date	Date of insertion		
Source						
Tables / Files		Colonnes	Type de données	Description	Règles de transformation	Règles de calul
BCSQT_CSF_CL_DELIVERY_LOAD		BRAND	text(10)	Brand identifier (AC/AP/OV)	No Transformation	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		PSV_PPROG	text(10)	Programming Country Code	No Transformation	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		FAMILY	text(4)	Family ID	No Transformation	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		TYPE	text(10)	PC/LCV identifier	No Transformation	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		MONTH	Date	Month date stamp in DD-MM-YYYY format	Fist day of month	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		CYCLE_ID	text(10)	PSA Planning Cycle	No Transformation	No calculation
>>		>>	>>	>>	Fixed value, depends on the source below	No calculation
BCSQT_CSF_CL_DELIVERY_LOAD		CURRENT_VOL	float	Related Quantity	if measure = Retails	No calculation
BCSQT_CSF_CL_INVOICE_LOAD		CURRENT_VOL	float	Related Quantity	if measure = Wholesales	No calculation
BCSQT_CSF_CL_MARKET_LOAD		CURRENT_VOL	float	Related Quantity	if measure = Market	No calculation
>>		>>	>>	Date of insertion	No Transformation	SYSTEM DATE

## 6.2.2. EProg-Carflow Mapping

Same for the data extracted from Eprogramme, the data mapping includes descriptions, examples, transformation rules, and finally calculations.

Table 11: EProg-Carflow Mapping

Dataset Carflow – Destination				Source		
Tables	Colonnes	Type de données	Description	Tables / Files	Table functional name	Colonnes
Carflow Tables	Brand	Bigint	surrogate key, linked with teh Brand dimension	BPG.RBVQTVPN	Brand label	VPN_CODE_MARQUE
	Country	Bigint	surrogate key, linked with the Country dimension	BPG.RBVQTGREP	Programming country	GREP_LIBELLE_PP
	Family	Bigint	surrogate key, linked with the Carline dimension	BPG.RBVQTFAT	Family decomposition	FAT_CODE_FAMILLE,
	MonthYear	Bigint	surrogate key, linked with the Date dimension	BPG.RBVQTPE1	Periods	Year = BPG.RBVQTPE1.PE1_ANNEE month = BPG.RBVQTPE1.PE1_NUMERO
	Cycle_Id	Bigint	surrogate key, linked with the Cycle dimension	BPG.RBVQTCA1	Cycles	CAT_CYCLE
	Measure	varchar(50)	Describes the nature of the measure (Wholesales, market ...)	>>		
	Value	float	The value of the measure, the unit depends on the nature of the measure	BPG.RBVQTVPN	Volume per Cycle / period	VPN_VOLUME_TOTAL
	Inserted_Date	Date		BPG.RBVQTVPN	Volume per Cycle / period	VPN_VOLUME_ALLOUE
Exemple	Type de données	Description	Règles de transformation	Règles de calul		
AP	char(2)	AC/AP/OV				
FRT530	varchar(50)	Code pays Programme PSV				
1SD3	varchar(50)	Code de famille (2PA9)				
20214	INT	Date of all/demand		Year = BPG.RBVQTPE1.PE1_ANNEE & month = BPG.RBVQTPE1.PE1_NUMERO		
202010	INT	Cycle				
	varchar(50)	Central demand/ Allocation	Fixed value, depends on the source			
3463	float	Volume of vehicles requested (demanded) by the subsidiary / importer for a type of its range.	if measure = central demand			
368	float	Volume of vehicles allocated by management for a type in the range of a subsidiary.	if measure = Allocation			
	Date	Date of insertion		system date		

### 6.2.3. Madax – Carflow Mapping

The same applies to data extracted from EPROG, the data mapping includes description, examples, transformation rules, and finally the calculations to be done.

Table 12: Madax-Carflow Mapping

Dataset Carflow - Destination				Source				
Tables	Colonnes	Type de données	Description	Tables / Files	Table functional name	Colonnes	Type de données	Description
TF_Carflow	Brand_Id	Bigint	surrogate key, linked with the Brand dimension	BDS.RBVQTFLL / BDS.RBVQTFAM	Madax NSC / Importers	MARCA/Marque	varchar(50)	APIAC/OV
	Country_Id	Bigint	surrogate key, linked with the Country dimension	BDS.RBVQTTFF	Madax NSC / Importers labels	LIBELLE_FRANCAIS	varchar(50)	Libellé des filiales en français
	Carline_Id	Bigint	surrogate key, linked with the Carline dimension	BDS.RBVQTDNE	Fictive family label	FAMILLE (Fictive family code)	varchar(50)	Code de Famille
	TYPE_Id	Bigint	surrogate key, linked with the Type dimension	BDS.RBVQTFAM	Fictive family label	VPVU	varchar(50)	Indicates if the vehicle is a VP (Véhicule Particulier) or a VU (Véhicule Utilitaire)
	MonthYear_Id	Bigint	surrogate key, linked with the Date dimension	BDS.RBVQTDNE	Actuals KPI	DATE_PHOTO (Date of Picture)	INT	Date
	Cycle_Id	Bigint	surrogate key, linked with the Cycle dimension	BDS.RBVQTDNE	Actuals KPI	from DATE_PHOTO	INT	Le Cycle
	Measure	varchar(50)	Describes the nature of the measure (wholesales, market ...)				varchar(50)	Stock Importers/ Stock Constructeur
	Value	Bigint	The value of the measure, the unit depends on the nature of the measure	BDS.RBVQTDNE	Actuals KPI	COM_M(CAF- Prod Mois)	INT	Number of Vehicles produced in the current month
				BDS.RBVQTDNE	Actuals KPI	TFAC_M(FAC - Fact PDV du Mois)	INT	Balance of network invoices of the month (takes into account cancellations)
				BDS.RBVQTDNE	Actuals KPI	TENTR_M(VEN - Stockout Mois)	INT	Stockout balance of the month
				BDS.RBVQTDNE	Actuals KPI	RED(STK Veh en Réseau)	INT	Vehicle in network stock, i.e.: Vehicle not sold and billed to the network
				BDS.RBVQTDNE	Actuals KPI		INT	Stock Constructeur
	Inserted_Date	Date	Date d'insertion					Date d'insertion

Description	Règles de transformation	Règles de calcul
APIAC/OV	P >> AP C >> AC	
Libellé des filiales en français	Transcodage de Libellé pays to code pays PSV	
Code de Famille		
Indicates if the vehicle is a VP (Véhicule Particulier) or a VU (Véhicule Utilitaire)	VP >> PV VU >> LCV	
Date	Extraction de la date seulement (Les 8 premiers caractères)	
Le Cycle	the first day of the month	
Stock Importers/ Stock Constructeur	Fixed value, depends on the source	
Number of Vehicles produced in the current month	if measure = Actual production	
Balance of network invoices of the month (takes into account cancellations)	if measure = Actual invoices	
Stockout balance of the month	if measure = Actual retails	
Vehicle in network stock, i.e.: Vehicle not sold and billed to the network	if measure = stock importer	
Stock Constructeur	if measure = stock constructor	BDS.RBVQTDNE.ENC_PAR C_USINE+BDS.RBVQTDNE.E NC_FACT+BDS.RBVQTDNE. PARQFIL
Date d'insertion		system date

## 6.2.4. iCube – Carflow Mapping

And finally, the data mapping for iCube is the same as the other sources with a little bit more transformation.

Table 13: iCube-Carflow Mapping

Dataset Carflow - Destination						
Tables	Description	Colonnes	Type de données	Description		
TF_Carflow	Fact table Carflow	Brand_Id	Bigint	surrogate key, linked with teh Brand dimension		
		Country_Id	Bigint	surrogate key, linked with the Country dimension		
		Carline_Id	Bigint	surrogate key, linked with the Carline dimension		
		TYPE_Id	Bigint	surrogate key, linked with the Type dimension		
		MonthYear_Id	Bigint	surrogate key, linked with the Date dimension		
		Cycle_Id	Bigint	surrogate key, linked with the Cycle dimension		
		Measure	varchar(50)	Describes the nature of the measure (Wholesales, market ...)		
		Value	float	The value of the measure, the unit depends on the nature of the measure		
		Inserted_Date	Date	Insertion Date		
Source						
Tables / Files		Colonnes	Type de données	Description	Règles de transformation	Règles de calcul
7 excel files PROREV_202107IMP_ACDS PROREV_202107IMP_AC - TR PROREV_202107IMP_DS - TR PROREV_202107IMP_AP PROREV_202107IMP_AP - TR PROREV_202107IMP_OV PROREV_202107IMP_OV - TR		Marque	text(10)	Brand identifier (AC/AP/OV)	C >> AC P >> AP	
		Pays	text(10)	Programming Country Code		
		Code version	text(4)	Family ID	Extraction of the first 4 characters	
		TYPE	text(10)	PC/LCV identifier, deduced from the country and the family	Transcodage	
		MONTH	Date		Fist day of month	
		Date	text(10)	From the file name " XXX CP01 XXX.xlsx"	From the file name " XXX CP01 XXX.xlsx"	
					fixed value = "importer demand"	
		excel cell values	float	Related Quantity		
				SYSTEM DATE		SYSTEM DATE

## 7. The Data Model

Below is the data model for the Carflow Fact table designed for the Carflow MEA Dashboard.

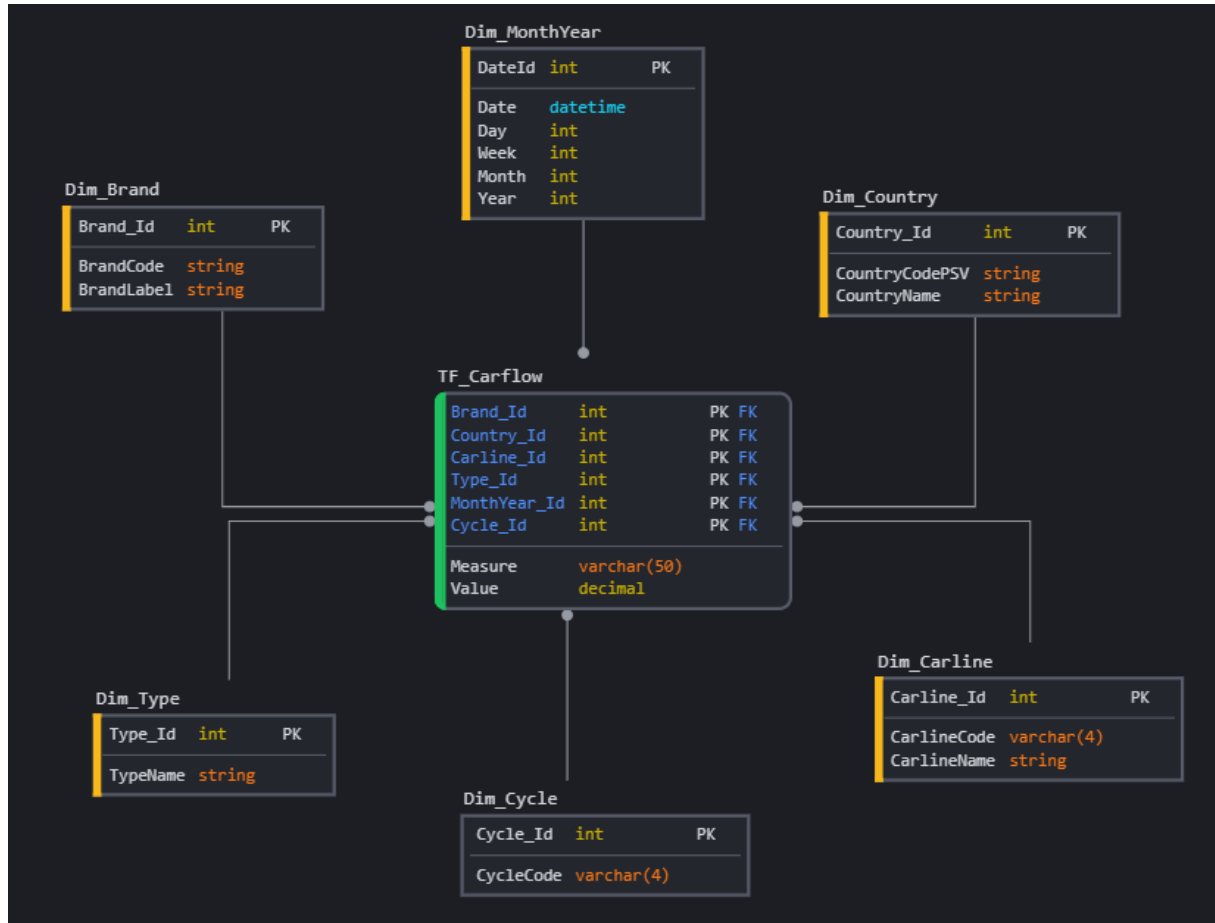


Figure 23: The Data Model

## 8. Solution Implementation

### 8.1. Building the Data Pipelines

In this step we are going to build our data pipeline based on the following architecture:

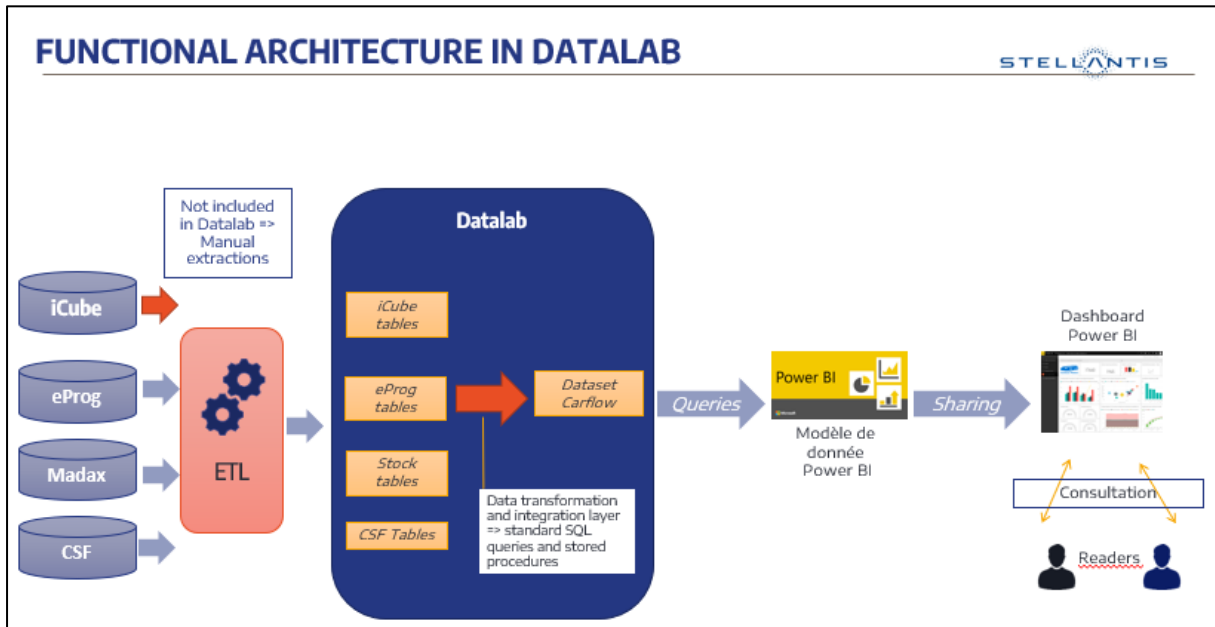


Figure 24: Functional Architecture of the Data Pipeline

With the help of the data sources analysis, the KPI analysis, and finally our data mapping, we can build our ETL pipelines:

- 3 pipelines for iCube ( AP, ACDS, OV) → Importer's production demands for each.
- 2 pipelines for Madax ( Retails, Wholesales Actuals) all brands.
- 3 pipelines for Eprogramme (AP, AC, OV) → Production Demands, Allocations for each pipeline.
- 3 pipelines for CSF (Delivery/Invoices/Market(Forecast)) → All brands.

### 8.1.1. iCube Pipelines

- **Extraction:** we're going to extract our data from a CSV file in our HDFS (for each brand there is a file identified by the cycle code), and we need to make the file path dynamic so it could change without manual help.
- **Transformations:**
  - **The source data format:**

Table 14: Source Table Format

	Marque	Pays	Libelle pays	Code version	Date	Volume 01	Volume 02	Volume 03	Volume 04	Volume 05	Volume 06	Volume 07	Volume 08	Volume 09	Volume 10	Volume 11	Volume 12
0	C	027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA050	202201	0	0	4	1	3	0	12	10	5	2	3	0
1	C	027987W	SOUTH AFRICA AC VT New	1CCESYVKNMURA050	202201	0	0	6	2	4	0	17	14	7	2	5	0
2	C	027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA046	202201	13	8	0	0	0	0	0	0	0	0	0	0
3	C	027987W	SOUTH AFRICA AC VT	1CCESYVKNMURA046	202201	19	11	0	0	0	0	0	0	0	0	0	0

- **The Destination Data format:**

Table 15: Destination Data Format

	Marque	Pays	LIBELLE_PAYS	FAMILLE	MONTHYEAR	Measure	Value	Cycle	INSERTED_DATE
0	C	027987W	SOUTH AFRICA AC VT New	1CCE	2022-01-01	DP IMP	0	2022-01-01	2022-07-13
1	C	027987W	SOUTH AFRICA AC VT New	1CCE	2022-01-01	DP IMP	0	2022-01-01	2022-07-13
2	C	027987W	SOUTH AFRICA AC VT New	1CCE	2022-01-01	DP IMP	13	2022-01-01	2022-07-13
3	C	027987W	SOUTH AFRICA AC VT New	1CCE	2022-01-01	DP IMP	19	2022-01-01	2022-07-13
4	C	027987W	SOUTH AFRICA AC VT New	1CW8	2022-01-01	DP IMP	17	2022-01-01	2022-07-13

To match the Destination Tables, we need to do some transformations:

- Change the data type of the Date column which contains our cycle (INT → Date)
- Change the names of the columns (Volumes 01/02...) to dates starting from the cycle date up to the start of each next month.
- Pivot the volumes columns, so we can get a column for the values (Production Demands/ Allocations) and a column for the date of each value.
- Keep only the first 4 characters of the Version Code (LCDV4).
- Add some columns : [Inserted\_date] to keep track of the insertion operation and [Measure] to identify if the value is a Production Demand or an Allocation.
- Rename the columns to match the destination table ( Code Version → Famille ...)



- **Loading:** Connect to the Oracle Database following the data mapping above and insert the data.

### 8.1.2. Eprogramme and Madax pipelines

- **Extraction:**
  - As the data of Eprogramme and Madax is available in the Oracle Database BRC06\_BPG00 and BRC06\_BDS00 respectively, we will need an SQL query to extract the data needed.
  - The issue with the extraction is that it cannot be done from a single table as it's available in different tables in the BRC06\_BPG00 and BRC06\_BDS00 databases. To solve this, we need to join these tables into our target table using different common keys.
  - We will also need to filter our data as we only need the data for certain brands, and countries...
  - We need also to filter by the cycle needed.
- **Loading:** Connect to the Oracle Database following the data mapping above and insert the data.

### 8.1.3. CSF pipelines

- **Extraction:**
  - As our data for the CSF KPIs are in our HDFS in a parquet format or CSV, we will import it using the file path and treat it like iCube.
- **Transformation:**

The data in our source is a little bit different from the destination one, so to match the two we will need to transform the source data.

Table 16: Data in the source

CYCLE_ID	CYCLE_TYP	OUTLOOK	REPREF	TYPE	BRAND	FAMILY	SUB_FAM	FICTIVE_F	SALES_CO	PSV_PCO	PSV_PPR	INVOICE	MONTH	YEAR	CURRENT_VOL
202208	PRE-REFE	0	0	PC	CITROEN	1CLE	1CLE-XXX	1CLE	YT	029973E	080032S		01-06-202	2022	0
202208	PRE-REFE	0	0	PC	DS	1SD4	1SD4-XXX	1SD4	BE	028831N	028831N		01-01-202	2023	85
202208	PRE-REFE	0	0	LCV	CITROEN	2CK0	2CK0-XXX	2CK0	GF	027139Z	080064B		01-03-202	2022	0
202208	PRE-REFE	0	0	LCV	OPEL	2GU9	2GU9-FRG	2GU9	GB	004538H	004538H		01-09-202	2023	259,742
202208	PRE-REFE	0	0	LCV	CITROEN	2CK0	2CK0-XXX	2CK0	YT	012698C	080033T		01-02-202	2022	0
202208	PRE-REFE	0	0	PC	PEUGEOT	1PP6	1PP6-641	1PP6	MG	006526Q	839894A		01-12-202	2023	0
202208	PRE-REFE	0	0	PC	CITROEN	1CK9	1CK9-XXX	1CK9	BE	028831N	028831N		01-10-202	2022	132,5036
202208	PRE-REFE	0	0	PC	PEUGEOT	KPPV	KPPV-300	KPPV	MM	870226U	839896V		01-07-202	2022	0
202208	PRE-REFE	0	0	PC	CITROEN	1CSC	1C51-XXX	1C51	SK	028541Y	080504E		01-10-202	2023	0

Table 17: Data in the destination

	PROGRAM_COUNTRY	FAMILY	MONTHYEAR	MEASURE	VALUE	INSERTED_DATE	CYCLE
1	006866X	1PM3	01/01/2021	Wholesales CP REF	1	05/05/21	01/07/21
2	006866X	1PM3	01/02/2021	Wholesales CP REF	1	05/05/21	01/07/21
3	006866X	1PM3	01/03/2021	Wholesales CP REF	4	05/05/21	01/07/21
4	006866X	1PM3	01/04/2021	Wholesales CP REF	0	05/05/21	01/07/21
5	006866X	1PM3	01/05/2021	Wholesales CP REF	7	05/05/21	01/07/21
6	006866X	1PM3	01/06/2021	Wholesales CP REF	0	05/05/21	01/07/21
7	006866X	1PM3	01/07/2021	Wholesales CP REF	0	05/05/21	01/07/21
8	006866X	1PM3	01/08/2021	Wholesales CP REF	0	05/05/21	01/07/21
9	006866X	1PM3	01/09/2021	Wholesales CP REF	0	05/05/21	01/07/21
10	006866X	1PM3	01/10/2021	Wholesales CP REF	0	05/05/21	01/07/21
11	006866X	1PM3	01/11/2021	Wholesales CP REF	0	05/05/21	01/07/21
12	006866X	1PM3	01/12/2021	Wholesales CP REF	0	05/05/21	01/07/21

- As we can see we need to delete some columns like Cycle\_type, Outlook, Brand, Invoice\_chanel...
- We need to rename some columns also.
- We need to add a new column called 'Measure' which tells us which KPI we're looking at.
- **Loading:** Connect to the Oracle Database following the data mapping above and insert the data.

## 8.2. Scheduling

We are going to schedule our jobs with the help of the ETL orchestrator Apache Airflow. The schedule is to be defined by the Business teams, but we will try to export the data as soon as it reaches its source.

## 9. Conclusion

In this chapter, we were able to discover the Conception Process that will lead us eventually to the implementation of our solution. In the next chapter we will discover how the solution was implemented and the tools used to help, we will also see the Result as filled Tables and Dashboards.

## Chapter Five: Implementation and Results



## 1. Introduction

- In this chapter, we will delve into the implementation of our solution and its results as well as the tools and software we work with to succeed in our implementation.

## 2. Tools and Softwares

- **Oracle SQL Developer:**



*Figure 25: Oracle SQL Developer  
Icon*

Oracle SQL Developer is a multi-platform integrated development environment (IDE) provided free of charge by Oracle Corporation and using Java technology (Java Development Kit). It is a graphical tool for querying Oracle databases using the SQL language.<sup>20</sup>

Oracle SQL Developer allows for the development of applications from scratch in PL/SQL, the provision of worksheets for executing queries and scripts, a console for database administration (DBA), an interface for reporting, a complete solution for data model design, and a migration interface for migrating third-party databases to Oracle2.

Oracle SQL Developer supports Oracle products as well as plugins that allow connection to non-Oracle databases. Oracle SQL Developer works with IBM DB2,

---

<sup>20</sup> Oracle.com. (2022). *SQL Developer*. [online] Available at: <https://www.oracle.com/database/sqldeveloper/> [Accessed 14 Jul. 2022].

Microsoft Access, Microsoft SQL Server, MySQL, Sybase Adaptive Server, and Teradata databases.<sup>21</sup>

This tool will help us explore our Oracle Databases, visualize, and create the Tables, and Query data from different tables.

- **Apache Airflow:**



*Figure 26: Apache Airflow Icon*

Apache Airflow is an ETL (Extraction, Transformation, Loading) workflow management system widely used for data transformation. It is coded in python and workflows are written via python scripts.

The use of python allows the developer to easily create workflows by importing python classes and libraries.

Workflows are organized and expressed as Directed Acyclic Graphs (DAG). In reality, it is the method chosen to perform a task.

It all started in 2014 at Airbnb. The company was growing and thus had a considerable volume of data to process. The Californian company hires Data Scientists, Data Analysts, and Data Engineers who must work closely together to process all this data.

---

<sup>21</sup> Oracle.com. (2018). *SQL Developer History*. [online] Available at: <https://www.oracle.com/database/technologies/appdev/sqldev/history.html> [Accessed 14 Jul. 2022].

To automate everything, they write scheduled batch jobs, so they can improve the quality of work. With this in mind, the data engineer, Maxime Beauchemin, created an open-source tool called Airflow. In April 2016, the Apache Foundation incubates the project and receives the status of "top-level" project in January 2019. It has over 1400 contributors, 11,230 contributions, and 19,800 stars on GitHub by the end of 2020.

There are many relevant use cases for Airflow, you can for example:

- Gather daily sales team updates from Salesforce to deliver a daily message to company executives.
- Organize and launch certain Machine Learning jobs that run on an external Spark cluster.
- Or load and analyze hourly, application, or website analytics data into a data warehouse.<sup>22</sup>

This tool will help us schedule our data pipeline executions.

- **Putty:**

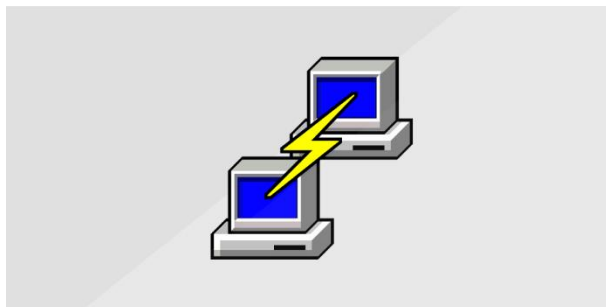


Figure 27:Putty Icon

PuTTY is a terminal emulator and client for SSH, Telnet, rlogin, and raw TCP protocols. It also allows direct connections via the RS-232 serial link. Originally available only for Windows, it is now ported to various Unix platforms (and unofficially to other platforms). PuTTY is written and maintained primarily by Simon Tatham.

It is free software distributed under the terms of the MIT license.<sup>23</sup>

---

<sup>22</sup> airflow.apache.org. (n.d.). *Apache Airflow Documentation — Airflow Documentation*. [online] Available at: <https://airflow.apache.org/docs/apache-airflow/stable/index.html> .

<sup>23</sup> the.earth.li. (n.d.). *Using PuTTY*. [online] Available at: <https://the.earth.li/~sgtatham/putty/0.54/html/doc/Chapter3.html> [Accessed 14 Jul. 2022].

This tool will help us to connect to our Stellantis Server to execute our scripts and check the data.

- **Visual Studio Code**



*Figure 28: Visual Studio Code Icon*

Visual studio code or VS Code is a code editor developed by Microsoft in 2015. Unlike what Microsoft used to get us used to for years, it is one of those first open source and free products and is especially available on Windows, Linux, and Mac operating systems. The code is developed with the Electron framework and is designed mainly to develop projects with JavaScript, Node.js, or TypeScript. This IDE will help us write our Python, SQL, and Shell Scripts.

- **WinSCP**



*Figure 29: WinSCP Icon*

WinSCP is a popular free SFTP and FTP client for Windows, a powerful file manager that will improve your productivity. It offers an easy-to-use GUI to copy files between a local and remote computer using multiple protocols: Amazon S3, FTP, FTPS, SCP, SFTP, or WebDAV. Power users can automate WinSCP using .NET assembly. WinSCP is available in English and many other languages.

### 3. iCube Pipelines and Automation jobs

#### 3.1. Pipeline code

In this section I will explain how I implemented my solution and because the code is similar for all the 3 brands (AC, AP, and OV), I'll take AC as an example:

- We need to import the libraries we will need in our code:

```
4  # ### Imports
5
6  get_ipython().run_line_magic('load_ext', 'autoreload')
7  get_ipython().run_line_magic('autoreload', '2')
8  import os
9  import datetime
10 import pandas as pd
11 pd.set_option("display.max_rows", 500)
12 pd.set_option("display.max_columns", 100)
13 from pyspark.sql import functions as F
14
15 from crf0a_app.configuration import spark_config
16 from crf0a_app.utils import system
17
```

Figure 30: iCubeData Pipeline Code Snippet 1

- And then we need to create a spark session to help with our analytics and transformations as well as our loading in the Oracle Datalab:

```
19 # ### Spark session
20
21
22 spark_context, spark_session = spark_config.get_spark(
23     app_name="[app00] Test_Read_Write_Data",
24     driver_cores=1,
25     driver_mem="4g",
26     max_executors=8,
27     executor_cores=4,
28     executor_mem="4g"
29 )
30
```

Figure 31: iCubeData Pipeline Code Snippet 2

- Next, I created a variable called cycle that changes our cycle dynamically using the today method from the DateTime library as a reference, this will help us change the directory of the file we look for in the HDFS automatically based on the current date.

For example: if the extraction is done on the 7th of October 2022 we will get the cycle variable (202210) with the help of the [\*DateTime.today\*](#) method. This will automatically generate the new file name which will be: [\*PROREV 202210IMP ACDS.csv\*](#)



```

37 # Data filepath
38 today = str(datetime.date.today())
39 cycle = today[0:4]+today[5:7]
40 flow_filepath = "/user/sd01865/cr-f0a/data/iCube/PROREV_%sIMP_ACDs.csv" %cycle
41 print(flow_filepath)

```

Figure 32: iCubeData Pipeline Code Snippet 3

- Then we will need to read the data from the CSV file, we also need to be precise that we have headers in our original file and also that the delimiter to look for is an «; ». We will do that with the help of the `spark_session.read.options(our options).csv(file path)` method.

```

43
44 # Read data from HDFS
45 df_flow = spark_session.read.options(header='True', inferSchema='True', delimiter=';').csv(flow_filepath)
46

```

Figure 33: iCubeData Pipeline Code Snippet 4

The output:

Table 18: Source Data

Marque	Pays	Libelle pays	Code version	Date	Volume 01	Volume 02	Volume 03	Volume 04	Volume 05	Volume 06	Volume 07	Volume 08	Volume 09	Volume 10	Volume 11	Volume 12
0	C 027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA050	202201	0	0	4	1	3	0	12	10	5	2	3	0
1	C 027987W	SOUTH AFRICA AC VT New	1CCESYVKNMURA050	202201	0	0	6	2	4	0	17	14	7	2	5	0
2	C 027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA046	202201	13	8	0	0	0	0	0	0	0	0	0	0
3	C 027987W	SOUTH AFRICA AC VT	1CCESYVKNMURA046	202201	19	11	0	0	0	0	0	0	0	0	0	0

- Next comes the transformations that need to be done so the source data matches the destination data.
- We're going to start by changing the data type of the Date column from integer to date type using the `pyspark.SQL.functions.to_date` method.
- Next, we will need a variable called `the_date`, this variable contains the date of the first-row fourth column which is the cycle date.

```

48 # ### Transformations
49
50 #change the data type of the Date column
51 df_flow = df_flow.withColumn("Date", F.to_date(F.col("Date").cast("string"), 'yyyyMMdd'))
52
53
54 #get the date value we will need to change the volume columns names
55 the_date = df_flow.collect()[0][4]
56 print(the_date)

```

Figure 34: iCubeData Pipeline Code Snippet 5

- Next, as mentioned in the conception part of this report, we will need to rename the Volume columns, and we will be able to do that with the help of [the\\_date](#) variable that we mentioned above, also with the help of the [pandas.withColumnRenamed](#) method that helps us rename our columns and a bunch of [DateTime](#) library methods such as [timedelta](#) and [replace](#), this will help us to dynamically change the column names to the correct dates with are the first of each month starting from the month of the cycle: for example, if the cycle of our extraction is 202203 then the first date of the volume 01 column will have to be 01/03/2022 and volume 02 will have to be 01/04/2022, etc.

```

58 #Rename the Volume cols with the date of the first of every month starting from the_date
59 df_flow = df_flow.withColumnRenamed("Volume 01",str(the_date)) /
60     .withColumnRenamed("Volume 02",str((the_date + datetime.timedelta(days=32)).replace(day=1)))/
61     .withColumnRenamed("Volume 03",str((the_date + datetime.timedelta(days=32*2)).replace(day=1)))/
62     .withColumnRenamed("Volume 04",str((the_date + datetime.timedelta(days=32*3)).replace(day=1)))/
63     .withColumnRenamed("Volume 05",str((the_date + datetime.timedelta(days=32*4)).replace(day=1)))/
64     .withColumnRenamed("Volume 06",str((the_date + datetime.timedelta(days=32*5)).replace(day=1)))/
65     .withColumnRenamed("Volume 07",str((the_date + datetime.timedelta(days=32*6)).replace(day=1)))/
66     .withColumnRenamed("Volume 08",str((the_date + datetime.timedelta(days=32*7)).replace(day=1)))/
67     .withColumnRenamed("Volume 09",str((the_date + datetime.timedelta(days=32*8)).replace(day=1)))/
68     .withColumnRenamed("Volume 10",str((the_date + datetime.timedelta(days=32*9)).replace(day=1)))/
69     .withColumnRenamed("Volume 11",str((the_date + datetime.timedelta(days=32*10)).replace(day=1)))/
70     .withColumnRenamed("Volume 12",str((the_date + datetime.timedelta(days=32*11)).replace(day=1)))/
71     .withColumnRenamed("Date","Cycle")
72

```

Figure 35: iCubeData Pipeline Code Snippet 6

Output:

```

Index(['Marque', 'Pays', 'Libelle pays', 'Code version', 'Cycle', '2022-01-01',
      '2022-02-01', '2022-03-01', '2022-04-01', '2022-05-01', '2022-06-01',
      '2022-07-01', '2022-08-01', '2022-09-01', '2022-10-01', '2022-11-01',
      '2022-12-01'],
      dtype='object')

```

Figure 36: Data Output after transformation

- We will next need to pivot the dates columns, so we can get a column for the values (Production Demands/ Allocations) and a column for the date of each value. We will do that with the help of the [pyspark.pandas.DataFrame.melt](#) method which helps to unpivot a DataFrame from wide format to long format.

```

76 #pivot the table
77 v pivoted_df = pd_df.melt(id_vars = ['Marque', 'Pays','Libelle pays','Code version','Cycle'], value_vars = ['2022-01-01',
78     '2022-02-01', '2022-03-01', '2022-04-01', '2022-05-01', '2022-06-01',
79     '2022-07-01', '2022-08-01', '2022-09-01', '2022-10-01', '2022-11-01',
80     '2022-12-01'], var_name = 'Date', value_name = 'Value')

```

Figure 37:iCubeData Pipeline Code Snippet 7

Output:

Table 19: Data output after transformation 2

	Marque	Pays	Libelle pays	Code version	Cycle	Date	Value
0	C	027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA050	2022-01-01	2022-01-01	0
1	C	027987W	SOUTH AFRICA AC VT New	1CCESYVKNMURA050	2022-01-01	2022-01-01	0
2	C	027987W	SOUTH AFRICA AC VT New	1CCESYTKNMURA046	2022-01-01	2022-01-01	13
3	C	027987W	SOUTH AFRICA AC VT New	1CCESYVKNMURA046	2022-01-01	2022-01-01	19
4	C	027987W	SOUTH AFRICA AC VT New	1CW8AFKCZMURA052	2022-01-01	2022-01-01	17

- As we only need the LCDV4 and not the whole Version Code we have to slice the values of the corresponding column to leave only the first 4 characters :

```
83 #slice the Code version values to leave only the first 4 characters
84 pivoted_df['Code version']=pivoted_df['Code version'].str[:4]
85
```

Figure 38: iCubeData Pipeline Code Snippet 8

- Finally, we need to write our data in the right table. To do that we need the [OracleDatabase](#) object from the [crf0a\\_app.infra.oracle\\_database\\_library](#). And we will need the write\_df\_to\_oracle method defined below, this method and the whole library are provided by the Stellantis Data Engineering team to help with such data projects.

```
95 def write_df_to_oracle(self, df, table_name, mode="overwrite", dict_type=None):
96     """Write dataframe in an Oracle table.
97
98     Parameters
99     -----
100     df : pd.DataFrame
101         Dataframe to write.
102     table_name : str
103         Oracle SQL table name.
104     mode : str
105         How to behave if the table already exists: 'append', 'truncate', 'overwrite', 'error'.
106     dict_type : dict, optional
107         Columns types.
108
109     """
```

Figure 39: write\_df\_to\_oracle function

- We will use the method explained above and precise our Dataframe, Target Table in the Oracle Datalab, and the writing mode be it 'append's 'overwrite'. Our target table in the Oracle Datalab, in this case, is BRC\_SD01865.DP\_IMP\_ACDS for the brands AC and DS importer's production demands.

```

114 # ### Write
115 from crf0a_app.infra.oracle_database import OracleDatabase
116
117
118 #Instantiate OracleDatabase object
119 oracle_db = OracleDatabase(dialect="jdbc", spark_session=spark_session)
120
121
122 # Write data to Oracle
123 oracle_db.write_df_to_oracle(
124     sparkDF,
125     "BRC_SD01865.DP_IMP_ACDS",
126     mode="append"
127 )

```

Figure 40: iCubeData Pipeline Code Snippet 9

### 3.2. Shell Script

- We will need the following Shell Script to automatically trigger the execution of our data pipeline using the airflow dags.

```

1  #!/usr/bin/sh
2  set -e
3  script_dirpath=$(dirname $0)
4  cd $script_dirpath/..
5  make install
6  source script/app_profile.sh
7
8  python "$UNXPACKAGE/pipeline/dp_imp_acds_data_pipeline.py"
9

```

Figure 41: iCube Shell Script

### 3.3. Airflow dags

- As explained in the Technical Concepts part, Airflow dags as basically python scripts that help us schedule our tasks using dags and operators such as bash\_operator or python\_operator. So we need to import the necessary libraries.

```

1  import os
2  import airflow
3  import pendulum
4  from datetime import datetime, timedelta
5  from textwrap import dedent
6  from airflow.models import DAG
7  from airflow.models import Variable
8  from airflow.operators.bash_operator import BashOperator
9  from airflow.operators.dummy_operator import DummyOperator
10 from airflow.operators.python_operator import PythonOperator, BranchPythonOperator
11 from airflow.utils.trigger_rule import TriggerRule

```

Figure 42: iCube Airflow Imports

- In this section, we need to define the DAG credentials, basically who is the person or the technical account responsible for this Airflow DAG.

```

14 # -----
15 #                               DAG credentials
16 # -----
17 # The user who runs the tasks must be a technical account NOT a personal account
18 # and has access to the corresponding BRC
19 V_USER = 'sd01865'
20

```

Figure 43: iCube Airflow Credentials

- Here, we need to set our Default arguments which will be passed in each operator, these arguments are the owner already mentioned, their email, and if they should receive the email when the task fails or is retried, the number of retries if the task fails once, the retry delay and other arguments.

```

33 v default_args = {
34     # DO NOT MODIFY THE OWNER
35     'owner': V_USER,
36     # People who receive emails
37     'email': ['salma.ouardi@external.stellantis.com'],
38     # Email when task fails
39     'email_on_failure': True,
40     # Email when task retries
41     'email_on_retry': False,
42     # Number of retries when a task failed
43     'retries': 0,
44     # Delay between two retries
45     'retry_delay': timedelta(minutes=5),
46     # The task will be executed ONLY if this task ran successfully in previous launch

```

Figure 44: iCube Airflow DAG Arguments

- Next, we need to define our DAG, we need to set the timezone, the start date of the task which is the first date at which the DAG must be launched, get the description for the dag, and the schedule interval using the cron syntax explained before.

```

70 local_tz = pendulum.timezone('Europe/Paris')
71 START_DATE = datetime(2022, 7, 11, tzinfo=local_tz)
72
73 # Set only description, schedule_interval, start_date
74 # Example: the DAG is executed every day at 8am
75 dag = DAG(
76     V_DAG_ID,
77     default_args=default_args,
78     description='DP IMP ACDS Data Pipeline (Airflow test)',
79     schedule_interval='0 1 * * *',
80     start_date=START_DATE,
81     catchup=False
82 )

```

Figure 45: iCube Airflow DAG Definition

- Then we need to specify the Shell script path so it can be then run by the BashOperator.

```

88 UNXAPPLI = '/gpfs/user/sd01865/crf0a'
89 CMD_DP_IMP_ACDS = 'cd ' + UNXAPPLI + ';script/dp_imp_acds_data_pipeline.sh;'
90
91 with dag:
92     # Only BashOperator can be used
93     t1 = BashOperator(
94         # Task id must be unique in the DAG
95         task_id='dp_imp_acds',
96         # To use variables in command, the command must be defined outside of the operator
97         bash_command=str(CMD_DP_IMP_ACDS),
98     )
99
100
101 # Organize dependencies between tasks
102 t1

```

Figure 46: iCube Airflow BashOperator

### 3.4. The Result

- Here below are the values inserted in our BRC\_SD01865.DP\_IMP\_ACDS table.

Table 20: iCube Data Pipeline Result

Marque	Pays	LIBELLE_PAYS	FAMILLE	MONTHYEAR	Measure	Value	Cycle	INSERTED_DATE
1 C	028271E	CYPRUS ACDS GPA VISION	1CB6	2022-04-01	DP IMP	0	01/01/22	13/07/22
2 C	029648B	COSTA RICA AC VEINSA	1CM5	2022-04-01	DP IMP	0	01/01/22	13/07/22
3 C	029648B	COSTA RICA AC VEINSA	2CU9	2022-04-01	DP IMP	0	01/01/22	13/07/22
4 C	029648B	COSTA RICA AC VEINSA	2CU9	2022-04-01	DP IMP	2	01/01/22	13/07/22
5 C	029648B	COSTA RICA AC VEINSA	1CEA	2022-04-01	DP IMP	0	01/01/22	13/07/22
6 C	029648B	COSTA RICA AC VEINSA	1CEA	2022-04-01	DP IMP	0	01/01/22	13/07/22
7 C	029648B	COSTA RICA AC VEINSA	1CEA	2022-04-01	DP IMP	0	01/01/22	13/07/22
8 C	029648B	COSTA RICA AC VEINSA	1CEA	2022-04-01	DP IMP	0	01/01/22	13/07/22
9 C	029648B	COSTA RICA AC VEINSA	1CCE	2022-04-01	DP IMP	0	01/01/22	13/07/22
10 C	029648B	COSTA RICA AC VEINSA	1CCE	2022-04-01	DP IMP	0	01/01/22	13/07/22
11 C	029648B	COSTA RICA AC VEINSA	1CCE	2022-04-01	DP IMP	0	01/01/22	13/07/22
12 C	029648B	COSTA RICA AC VEINSA	1CW8	2022-04-01	DP IMP	0	01/01/22	13/07/22
13 C	029648B	COSTA RICA AC VEINSA	2CU9	2022-04-01	DP IMP	0	01/01/22	13/07/22
14 C	029648B	COSTA RICA AC VEINSA	2CU9	2022-04-01	DP IMP	0	01/01/22	13/07/22
15 C	029648B	COSTA RICA AC VEINSA	2CU9	2022-04-01	DP IMP	0	01/01/22	13/07/22



## 4. EProg Pipelines and Automation jobs

### 4.1. SQL Query

- To extract the data from the Eprogramme system we will need to use SQL scripts as the data is in an Oracle Database called BRC06\_BPG00.

```
1  /*Created by SALMA GUARDI */
2  /* Query for PROD DEM / AP */
3  SELECT
4      a1.grep_libelle_pp_22      program_country,
5      a1.fal_code_famille_12     family,
6      '01'
7      || '/'
8      || a1.pe1_numero_15
9      || '/'
10     || a1.pe1_annee_14 monthyear,
11     'PROD DEM' measure,
12     SUM(a1.vpn_volume_total) AS valeur,
13     to_char(sysdate) inserted_date,
14     a1.cal_cycle_11             cycle
15 FROM
16     (
17         SELECT
18             a3.vpn_id_pays_unitaire_0      vpn_id_pays_unitaire,
19             a3.vpn_code_marque_1           vpn_code_marque_1,
20             a3.vpn_num_cycle_2             vpn_num_cycle,
21             a3.vpn_annee_cycle_3           vpn_annee_cycle,
22             a3.vpn_periode_4               vpn_periode,
23             a3.vpn_est_basculer_6           vpn_est_basculer_6,
24             a3.cal_code_marque_7           cal_code_marque,
25             a3.cal_annee_cycle_8           cal_annee_cycle,
26             a3.cal_numero_cycle_9          cal_numero_cycle,
27             a3.cal_code_zone_production_10 cal_code_zone_production_10,
28             a3.cal_cycle_11               cal_cycle_11,
29             a3.fal_code_famille_12         fal_code_famille_12,
30             a3.fal_code_marque_13         fal_code_marque_13,
31             a3.pe1_annee_14               pe1_annee_14,
32             a3.pe1_numero_15              pe1_numero_15,
33             a3.pe1_code_zone_production_16 pe1_code_zone_production_16,
34             a3.pe1_periode_17             pe1_periode,
35             a3.pu1_id_pays_unitaire_18     pu1_id_pays_unitaire,
36             a3.pu1_code_psv_pays_programme_19 pu1_code_psv_pays_programme,
37             a2.grep_libelle_groupe_reporting grep_libelle_groupe_reporting_20,
38             a2.grep_code_psv_pays_programme grep_code_psv_pays_programme,
39             a2.grep_libelle_pp            grep_libelle_pp_22,
40             a3.vpn_volume_total
41         FROM
42             (
43                 SELECT
44                     a5.vpn_id_pays_unitaire_0      vpn_id_pays_unitaire_0,
45                     a5.vpn_code_marque_1           vpn_code_marque_1,
46                     a5.vpn_num_cycle_2             vpn_num_cycle_2,
47                     a5.vpn_annee_cycle_3           vpn_annee_cycle_3,
48                     a5.vpn_periode_4               vpn_periode_4,
49                     a5.vpn_est_basculer_6           vpn_est_basculer_6,
50                     a5.cal_code_marque_7           cal_code_marque_7,
51                     a5.cal_annee_cycle_8           cal_annee_cycle_8,
52                     a5.cal_numero_cycle_9          cal_numero_cycle_9,
53                     a5.cal_code_zone_production_10 cal_code_zone_production_10,
54                     a5.cal_cycle_11               cal_cycle_11,
55                     a5.fal_code_famille_12         fal_code_famille_12,
56                     a5.fal_code_marque_13         fal_code_marque_13,
57                     a5.pe1_annee_14               pe1_annee_14,
58                     a5.pe1_numero_15              pe1_numero_15,
59                     a5.pe1_code_zone_production_16 pe1_code_zone_production_16,
60                     a5.pe1_periode_17             pe1_periode_17,
61                     a4.pu1_id_pays_unitaire        pu1_id_pays_unitaire_18,
62                     a4.pu1_code_psv_pays_programme pu1_code_psv_pays_programme_19,
63                     a5.vpn_volume_total
64                 FROM
65                     (
66                         SELECT
67                             a7.vpn_id_pays_unitaire_0      vpn_id_pays_unitaire_0,
68                             a7.vpn_code_marque_1           vpn_code_marque_1,
69                             a7.vpn_num_cycle_2             vpn_num_cycle_2,
70                             a7.vpn_annee_cycle_3           vpn_annee_cycle_3,
71                             a7.vpn_periode_4               vpn_periode_4,
72                             a7.vpn_est_basculer_6           vpn_est_basculer_6,
73                             a7.cal_code_marque_7           cal_code_marque_7,
74                             a7.cal_annee_cycle_8           cal_annee_cycle_8,
75                             a7.cal_numero_cycle_9          cal_numero_cycle_9,
76                             a7.cal_code_zone_production_10 cal_code_zone_production_10,
77                             a7.cal_cycle_11               cal_cycle_11,
78                             a7.fal_code_famille_12         fal_code_famille_12,
79                             a7.fal_code_marque_13         fal_code_marque_13,
80                             a6.pe1_annee                  pe1_annee_14,
81                             a6.pe1_numero                  pe1_numero_15,
82                             a6.pe1_code_zone_production    pe1_code_zone_production_16,
83                             a6.pe1_periode                  pe1_periode_17,
84                             a7.vpn_volume_total
85                         FROM
86                             (
87                                 SELECT
88                                     a9.vpn_id_pays_unitaire_0      vpn_id_pays_unitaire_0,
89                                     a9.vpn_code_marque_1           vpn_code_marque_1,
90                                     a9.vpn_num_cycle_2             vpn_num_cycle_2,
91                                     a9.vpn_annee_cycle_3           vpn_annee_cycle_3,
92                                     a9.vpn_periode_4               vpn_periode_4,
93                                     a9.vpn_est_basculer_6           vpn_est_basculer_6,
94                                     a9.cal_code_marque_7           cal_code_marque_7,
95                                     a9.cal_annee_cycle_8           cal_annee_cycle_8,
96                                     a9.cal_numero_cycle_9          cal_numero_cycle_9,
97                                     a9.cal_code_zone_production_10 cal_code_zone_production_10,
98                                     a9.cal_cycle_11               cal_cycle_11,
99                                     a9.fal_code_famille_12         fal_code_famille_12,
100                                    a9.fal_code_marque_13         fal_code_marque_13,
101                                    a9.vpn_volume_total
102                                 FROM
103                                     (
104                                         SELECT
105                                             a11.vpn_id_pays_unitaire      vpn_id_pays_unitaire_0,
106                                             a11.vpn_code_marque            vpn_code_marque_1,
107                                             a11.vpn_num_cycle              vpn_num_cycle_2,
108                                             a11.vpn_annee_cycle            vpn_annee_cycle_3,
109                                             a11.vpn_periode                vpn_periode_4,
110                                             a11.vpn_est_basculer            vpn_est_basculer_6,
111                                             a10.cal_code_marque            cal_code_marque_7,
112                                             a10.cal_annee_cycle            cal_annee_cycle_8,
113                                             a10.cal_numero_cycle            cal_numero_cycle_9,
114                                             a10.cal_code_zone_production    cal_code_zone_production_10,
115                                             a10.cal_cycle                  cal_cycle_11,
116                                             a11.vpn_volume_total
117                                         FROM
118                                             brc06_bpg00.rbvqtvpv a11,
119                                             brc06_bpg00.rbvqtcal a10
120                                         WHERE
121                                             a10.cal_code_marque = a11.vpn_code_marque
122                                             AND a10.cal_annee_cycle = a11.vpn_annee_cycle
123                                             AND a10.cal_numero_cycle = a11.vpn_num_cycle
124                                         ) a9,
125                                     brc06_bpg00.rbvqtfa1
126                                 WHERE
127                                     a9.vpn_code_marque_1 = a8.fal_code_marque
128                                     ) a7,
129                                     brc06_bpg00.rbvqtpe1
130                                 WHERE
131                                     a7.vpn_periode_4 = a6.pe1_periode
132                             ) a5,
133                             brc06_bpg00.rbvqtptu1
134                             WHERE
135                                 a5.vpn_id_pays_unitaire_0 = a4.pu1_id_pays_unitaire
136                         ) a3,
137                         brc06_bpg00.rbvqtgrep
138                     WHERE
139                         a3.pu1_code_psv_pays_programme_19 = a2.grep_code_psv_pays_programme
140                     GROUP BY
141                         a3.vpn_id_pays_unitaire_0,
142                         a3.vpn_code_marque_1,
143                         a3.vpn_num_cycle_2,
144                         a3.vpn_annee_cycle_3,
145                         a3.vpn_periode_4,
146                         a3.vpn_est_basculer_6,
147                         a3.cal_code_marque_7,
148                         a3.cal_annee_cycle_8,
149                         a3.cal_numero_cycle_9,
150                         a3.cal_code_zone_production_10,
151                         a3.cal_cycle_11,
152                         a3.fal_code_famille_12,
153                         a3.fal_code_marque_13,
154                         a3.pe1_annee_14,
155                         a3.pe1_numero_15,
156                         a3.pe1_code_zone_production_16,
157                         a3.pe1_periode_17,
158                         a3.pu1_id_pays_unitaire_18,
159                         a3.pu1_code_psv_pays_programme_19,
160                         a2.grep_libelle_groupe_reporting,
161                         a2.grep_code_psv_pays_programme,
162                         a2.grep_libelle_pp,
163                         a3.vpn_volume_total
164                 ) a1
165             )
166         )
167     )
168 )
```

Figure 47: The SQL script to Extract the Production demands from Eprog part(1)



- The script above will help us extract the data needed from different tables in the BRC06\_BPG00 database :
- The Data Mapping helped a lot in this step as the number of tables and columns in each table can be overwhelming, the data mapping step proved to be a crucial part of this project.
- The tables used in our join operation :
  - [BRC06\\_BPG00.RBVQTVPN \(Volumes per cycle\)](#)
  - [BRC06\\_BPG00.RBVQTGREP\(Programming Country\)](#)
  - [BRC06\\_BPG00.RBVQTFAl\(Family Decomposition\)](#)
  - [BRC06\\_BPG00.RBVQTPEI\(Periods\)](#)
  - [BRC06\\_BPG00.RBVQTCAI\(Cycles\)](#)

We need to filter by the cycle, the production zone, the brand, and the group of countries in this case DMOA → MEA region.



There are 6 SQL scripts for the Eprog extraction ([--Allocations \[AC, AP, OV\] – Production demands\[AC, AP, OV\]](#)) But I choose to illustrate only one of them as the only difference is the filter on the Brand(AC, AP, OV) and the type of volume to extract be it Allocations or production demands ( All is well explained in the data mapping above)

```

165  ✓ WHERE
166      a1.ca1_cycle_11 = {cycle}
167      AND a1.pe1_code_zone_production_16 LIKE 'PSA'
168      AND a1.vpn_code_marque_1 LIKE 'AP'
169      AND a1.grep_libelle_groupe_reporting_20 LIKE '500-DMOA'
170  ✓ GROUP BY
171      a1.fa1_code_famille_12,
172  ✓ HAVING
173      SUM(a1.vpn_volume_total) IS NOT NULL

```

Figure 48: The SQL script to Extract the Production demands from Eprog part(2)

## 4.2. Pipeline code

- We need to import the libraries needed just like the iCube pipeline.

```
4  # ### Imports
5
6  get_ipython().run_line_magic('load_ext', 'autoreload')
7  get_ipython().run_line_magic('autoreload', '2')
8  import os
9  import datetime
10 import pandas as pd
11 pd.set_option("display.max_rows", 500)
12 pd.set_option("display.max_columns", 100)
13 from pyspark.sql import functions as F
14
15 from crf0a_app.configuration import spark_config
16 from crf0a_app.utils import system
17
18
19 # ### Spark session
20
21 spark_context, spark_session = spark_config.get_spark(
22     app_name="[crf0A] Write to Exadata",
23     driver_cores=1,
24     driver_mem="4g",
25     max_executors=8,
26     executor_cores=4,
27     executor_mem="4g"
28 )
```

Figure 49: Eprog Pipeline Code Snippet 1

- We need to Instantiate the OracleDatabase object like before to be able to connect to the Oracle Database to read from there and eventually write there too.

```
34
35 from crf0a_app.infra.oracle_database import OracleDatabase
36
37 oracle_db = OracleDatabase(dialect="jdbc", spark_session=spark_session)
38
```

Figure 50: Eprog Pipeline Code Snippet 2

- We will need our cycle variable to render our SQL query dynamic.
- Then we will create a PROD\_DEM variable where we will store our SQL query as a string but we will need to use the « f » at the beginning of our query and curly braces containing

expressions that will be replaced with their values. In this case, the {cycle} value that we will filter on.

```
42 # #### Read
43
44 today = str(date.today())
45 cycle = today[0:4]+today[5:7]
46
47 # SQL query
48 PROD_DEM = f"""
49 /* Query for PROD DEM / AP */
50 SELECT
51     a1.grep_libelle_pp_22    program_country,
52     a1.fa1_code_famille_12  family,
53     '01'
54     || '/'
55     || a1.pe1_numero_15
56     || '/'
57     || a1.pe1_annee_14 monthyear,
58     'PROD DEM' measure,
59     SUM(a1.vpn_volume_total) AS valeur,
60     to_char(sysdate) inserted_date,
61     a1.ca1_cycle_11         cycle
```

Figure 51: Eprog Pipeline Code Snippet 3

- We need to read the data using the read\_df\_from\_query method explained below (it's also a method provided by the Stellantis Data team to facilitate the data projects)

```
41 def read_df_from_query(self,query,fetchsize=20000,partition_column=None,
42     n_partitions=None,lower_bound=None,upper_bound=None,):
43     """Load Oracle SQL query output in a dataframe.
44
45     Parameters
46     -----
47     query : str
48         SQL query (note: multi-line triple-quoted strings work).
49     fetchsize : int, default 20000
50         Number of rows to load per network call.
51     partition_column : str, optional
52         Column on which to partition.
53     n_partitions : int, optional
54         Number of partitions.
55     lower_bound : int, optional
56         Lower bound of the partition column data.
57     upper_bound : int, optional
58         upper bound of the partition column data.
59
60     Returns
61     -----
62     pyspark.sql.dataframe.DataFrame or pd.DataFrame
63         Output of the SQL query.
64
65     """
```

Figure 52: Eprog Pipeline Code Snippet 4

Read the data using our PROD\_DEM query :

```
241 # Read data from Oracle
242 prod_dem_ov = oracle_db.read_df_from_query(PROD_DEM, fetchsize=20000)
243
```

Figure 53: Eprog Pipeline Code Snippet 5

Output:

Table 21: Data Output Eprog

	PROGRAM_COUNTRY	FAMILY	MONTHYEAR	MEASURE	VALEUR	INSERTED_DATE	CYCLE
0	TURQUIE	1PTC	01/9/2022	PROD DEM	7551.0000000000	11-JUL-22	202207
1	TURQUIE	1PS2	01/9/2022	PROD DEM	7551.0000000000	11-JUL-22	202207
2	TURQUIE	1PT3	01/9/2022	PROD DEM	7551.0000000000	11-JUL-22	202207
3	TURQUIE	1P25	01/9/2022	PROD DEM	7551.0000000000	11-JUL-22	202207
4	TURQUIE	2PKP	01/9/2022	PROD DEM	7551.0000000000	11-JUL-22	202207

- This time the PROD\_ALL query that helps us extract our Allocation values :

```
244 # SQL query
245 PROD_ALL = f"""(
246 /* Query for AP / PROD ALL */
247 SELECT
248     a1.grep_libelle_pp_22    program_country,
249     a1.fa1_code_famille_12   family,
250     '01' || '/' || a1.pe1_numero_15 || '/' || a1.pe1_annee_14 monthyear,
251     'PROD ALL' Measure,
252     SUM(a1.vpn_volume_alloue_5) VALEUR,
253     to_char(sysdate) inserted_date,
254     a1.ca1_cycle_11 cycle
255
```

Figure 54: Eprog Pipeline Code Snippet 6

- Same as before, we read our data and then we write everything our target table: BRC\_SD01865.EPS\_AP

```

406 # Read data from Oracle
407 prod_all_ov = oracle_db.read_df_from_query(PROD_ALL, fetchsize=20000)
408
409 # #### Write
410
411 # Write data to Oracle
412 oracle_db.write_df_to_oracle(
413     prod_dem_ov,
414     "BRC_SD01865.EPS_AP",
415     mode="append"
416 )
417
418 # Write data to Oracle
419 oracle_db.write_df_to_oracle(
420     prod_all_ov,
421     "BRC_SD01865.EPS_AP",
422     mode="append"
423 )

```

Figure 55: Eprog Pipeline Code Snippet 7

### 4.3. Shell Script

- We will need the following Shell Script to automatically trigger the execution of our data pipeline using the airflow dags.
- Same as the previous script we only change the data pipeline path.

```

1  #!/usr/bin/sh
2  set -e
3  script_dirpath=$(dirname $0)
4  cd $script_dirpath/..
5  make install
6  source script/app_profile.sh
7
8  python "$UNXPACKAGE/pipeline/eprog_ap_data_pipeline.py"

```

Figure 56: Eprog Shell Script

### 4.4. Airflow dags

- Same as the previous Airflow Dag we only change Dag's description and the Shell Script to trigger in this case.

```

63 # -----
64 #           Define a DAG
65 # -----
66 # The first date at which the DAG must be launched
67 # Tip: if you want to run your job for the first time on 2021/07/02, set START_DATE to 2021/07/01
68 # The default Airflow timezone is UTC, it is then necessary to specify the timezone
69 # in the START_DATE before setting the schedule interval
70 local_tz = pendulum.timezone('Europe/Paris')
71 START_DATE = datetime(2022, 7, 11, tzinfo=local_tz)
72
73 # Set only description, schedule_interval, start_date
74 # Example: the DAG is executed every day at 8am
75 dag = DAG(
76     V_DAG_ID,
77     default_args=default_args,
78     description='eProg AP Data Pipeline (Airflow test)',
79     schedule_interval='0 1 * * *',
80     start_date=START_DATE,
81     catchup=False
82 )

```

Figure 57: Eprog Airflow DAG

```

84 # -----
85 #           Define and set DAG's tasks
86 # -----
87 # To use variables in task command, the command must be defined outside of the operator
88 UNXAPPLI = '/gpfs/user/sd01865/crf0a'
89 CMD_EPROG_AP = 'cd ' + UNXAPPLI + ';script/eprog_ap_data_pipeline.sh;'
90
91 with dag:
92     # Only BashOperator can be used
93     t1 = BashOperator(
94         # Task id must be unique in the DAG
95         task_id='eprog_ap',
96         # To use variables in command, the command must be defined outside of the operator
97         bash_command=str(CMD_EPROG_AP),
98     )
99
100
101 # Organize dependencies between tasks
102 t1

```

Figure 58: Eprog Airflow BashOperator

## 4.5. The Result

- Here below are the values inserted in our BRC\_SD01865.EPS\_AP table.

Table 22: Eprog Data Pipeline Result

	PROGRAM_COUNTRY	FAMILY	MONTHYEAR	MEASURE	VALEUR	INSERTED_DATE	CYCLE
1	TURQUIE	1PT5	01/11/2022	PROD ALL	6150	11-JUL-22	202207
2	TURQUIE	1PP5	01/11/2022	PROD ALL	6150	11-JUL-22	202207
3	TURQUIE	1PU6	01/11/2022	PROD ALL	6150	11-JUL-22	202207
4	TURQUIE	1PT6	01/11/2022	PROD ALL	6150	11-JUL-22	202207
5	TURQUIE	1PTZ	01/11/2022	PROD ALL	6150	11-JUL-22	202207
6	TURQUIE	1PPW	01/11/2022	PROD ALL	6150	11-JUL-22	202207
7	TURQUIE	1PS1	01/11/2022	PROD ALL	6150	11-JUL-22	202207
8	TURQUIE	1P54	01/11/2022	PROD ALL	6150	11-JUL-22	202207
9	TURQUIE	1PMS	01/11/2022	PROD ALL	6150	11-JUL-22	202207
10	TURQUIE	1PB0	01/11/2022	PROD ALL	6150	11-JUL-22	202207
11	TURQUIE	1PTC	01/9/2022	PROD DEM	7551	11-JUL-22	202207
12	TURQUIE	1PS2	01/9/2022	PROD DEM	7551	11-JUL-22	202207

## 5. Madax Pipelines and Automation jobs

### 5.1. SQL Query

To extract the data from the Madax system we will need to use SQL scripts as the data is in an Oracle Database called BRC06\_BDS00.

- This is the script to extract the wholesales data :

```
1  /* Query for WHOLESALES ACTUALS */
2
3  SELECT
4      a1.libelle_pp_5    program_country,
5      a1.famille_0      family,
6      CAST(a1.date_photo_2 AS DATE) monthyear,
7      'wholesales' measure,
8      SUM(a1.tfac_m_1) value,
9      to_char(sysdate) inserted_date,
10     '{cycle}' cycle
11 FROM
12     (
13         SELECT
14             a3.famille      famille_0,
15             a3.tfac_m      tfac_m_1,
16             CAST(a3.date_photo AS DATE) date_photo_2,
17             a3.qi_filial   qi_filial,
18             a2.qi_filiale  qi_filiale,
19             a2.libelle_pp  libelle_pp_5
20         FROM
21             brc06_bds00.rbvqtdne a3,
22             brc06_bds00.rbvqtpco a2
23         WHERE
24             a2.qi_filiale = a3.qi_filial
25             AND a3.date_photo >= TO_DATE('{firstday_lastMonth}', 'dd/mm/yyyy')
26             AND a3.date_photo <= TO_DATE('{lastday_lastMonth2}', 'dd/mm/yyyy')
27     ) a1
```

Figure 59: Madax SQL Script 1

- The tables we need to extract the data from using join operations :
  - BRC06\_BDS00.RBVQTDNE (Actual KPIs)
  - BRC06\_BDS00.RBVQTPCO (Programming Countries)

We will need to filter on the date using dynamic variables to be explained more in the data pipeline part, also on the Programming Countries that are part of the MEA region.

```

28 WHERE
29     ( a1.libelle_pp_5 = 'AFRIQUE' )
30     OR ( a1.libelle_pp_5 = 'ALGERIE' )
31     OR ( a1.libelle_pp_5 = 'ZONE ALGERIE' )
32     OR ( a1.libelle_pp_5 = 'EGYPTE' )
33     OR ( a1.libelle_pp_5 = 'MASHREQ' )
34     OR ( a1.libelle_pp_5 = 'MAURICE AC' )
35     OR ( a1.libelle_pp_5 = 'MAURICE OV' )
36     OR ( a1.libelle_pp_5 = 'MAURICE AP' )
37     OR ( a1.libelle_pp_5 = 'MASHREQ ZONE OV' )
38     OR ( a1.libelle_pp_5 = 'MAROC' )
39     OR ( a1.libelle_pp_5 = 'NIGERIA' )
40     OR ( a1.libelle_pp_5 = 'ARABIE' )
41     OR ( a1.libelle_pp_5 = 'AFRIQUE DU SUD' )
42     OR ( a1.libelle_pp_5 = 'ZONE TUNISIE' )
43     OR ( a1.libelle_pp_5 = 'TURQUIE' )
44     OR ( a1.libelle_pp_5 = 'TURQUIE DS' )
45 GROUP BY
46     a1.libelle_pp_5,
47 HAVING
48     SUM(a1.tentr_m_1) IS NOT NULL

```

Figure 60: Madax SQL Script 2

- This is the script to extract the retail data similar to the previous script the only difference is in this script we extract the retail data and in the previous one, the wholesales data + same filters:

```

1  /* Query for Retails ACTUALS */
2
3  SELECT
4      a1.libelle_pp_5    program_country,
5      a1.famille_0       family,
6      CAST(a1.date_photo_2 AS DATE) monthyear,
7      'Retails' measure,
8      SUM(a1.tentr_m_1) value,
9      to_char(sysdate)   inserted_date,
10     '{cycle}' cycle
11 FROM
12     (
13         SELECT
14             a3.famille        famille_0,
15             a3.tentr_m        tentr_m_1,
16             CAST(a3.date_photo AS DATE) date_photo_2,
17             a3.qi_filial       qi_filial,
18             a2.qi_filiale      qi_filiale,
19             a2.libelle_pp      libelle_pp_5
20         FROM
21             brc06_bds00.rbvqtdne a3,
22             brc06_bds00.rbvqtpco a2
23         WHERE
24             a2.qi_filiale = a3.qi_filial
25             AND a3.date_photo >= TO_DATE('{firstday_lastMonth}', 'dd/mm/yyyy')
26             AND a3.date_photo <= TO_DATE('{lastday_lastMonth2}', 'dd/mm/yyyy')
27     ) a1

```

Figure 61: Madax SQL Script 3



## 5.2. Pipeline code

The Madax pipeline is similar to EProg one, but as here we don't have a cycle dimension to distinguish which cycle, we're in, we will use the business rules to extract.

We will need our famous cycle dynamic variable that we used all along this project and we will also need two other variables which are the {firstday\_lastmonth} and {lastday\_lastmonth} dynamic variables:

- {firstday\_lastmonth}: helps us get the first day of the previous month
- {lastday\_lastmonth}: helps us get the last day of the previous month
- These two will help us get only the data of the last month (we can change these parameters by the business demand, if they for example want the data from the start of the year this the current day and use the overwrite mode to actualize the data)

```
42 # #### Read
43
44 today = str(date.today())
45 cycle = today[0:4]+today[5:7]
46
47
48 todayy = datetime.date.today()
49 first = todayy.replace(day=1)
50 lastday_lastMonth1 = (first - datetime.timedelta(days=1))
51 lastday_lastMonth2 = (first - datetime.timedelta(days=1)).strftime("%d/%m/%Y")
52 print(lastday_lastMonth2)
53
54 firstday_lastMonth = (first - datetime.timedelta(days=lastday_lastMonth1.day)).strftime("%d/%m/%Y")
55 print(firstday_lastMonth)
56
```

Figure 62: Madax Pipeline Code Snippet 1

- Our Wholesales\_query is the SQL script explained above, this goes the same as the Eprog pipeline.

```

58 # SQL query
59 wholesales_query = f"""
60 (SELECT
61     a1.libelle_pp_5    program_country,
62     a1.famille_0      family,
63     CAST(a1.date_photo_2 AS DATE) monthyear,
64     'Wholesales' measure,
65     SUM(a1.tfac_m_1) value,
66     to_char(sysdate)   inserted_date,
67     '{cycle}' cycle
68 FROM
69 (

```

Figure 63:Madax Pipeline Code Snippet 2

- Read the data using the wholesales\_query and write in our Oracle table BRC\_SD01865.WS\_ACT.

```

119 # Read data from Oracle
120 wholesales = oracle_db.read_df_from_query(wholesales_query, fetchsize=20000)
121
122
123 # Write data to Oracle
124 oracle_db.write_df_to_oracle(
125     wholesales,
126     "BRC_SD01865.WS_ACT",
127     mode="overwrite"
128 )

```

Figure 64: Madax Pipeline Code Snippet 3

Output:

Table 23: Data output Madax

	PROGRAM_COUNTRY	FAMILY	MONTHYEAR	MEASURE	VALUE	INSERTED_DATE	CYCLE
0	ALGERIE	1CM3	2022-06-30	Wholesales	3311.0000000000	12-JUL-22	202207
1	ALGERIE	1SX8	2022-06-30	Wholesales	1925.0000000000	12-JUL-22	202207
2	AFRIQUE	1PP2	2022-06-30	Wholesales	1768.0000000000	12-JUL-22	202207
3	TURQUIE	1PP8	2022-06-30	Wholesales	1516.0000000000	12-JUL-22	202207
4	AFRIQUE	1PP1	2022-06-30	Wholesales	1483.0000000000	12-JUL-22	202207

### 5.3. Shell Script

- We will need the following Shell Script to automatically trigger the execution of our data pipeline using the airflow dags.
- Same as the previous script we only change the data pipeline path.

```

1  #!/usr/bin/sh
2  set -e
3  script_dirpath=$(dirname $0)
4  cd $script_dirpath/..
5  make install
6  source script/app_profile.sh
7
8  python "$UNXPACKAGE/pipeline/wholesales_actuals_data_pipeline.py"

```

Figure 65: Shell Script Wholesales

```

1  #!/usr/bin/sh
2  set -e
3  script_dirpath=$(dirname $0)
4  cd $script_dirpath/..
5  make install
6  source script/app_profile.sh
7
8  python "$UNXPACKAGE/pipeline/wholesales_actuals_data_pipeline.py"

```

```

1  #!/usr/bin/sh
2  set -e
3  script_dirpath=$(dirname $0)
4  cd $script_dirpath/..
5  make install
6  source script/app_profile.sh
7
8  python "$UNXPACKAGE/pipeline/retails_actuals_data_pipeline.py"

```

Figure 66: Shell Script Retails

## 5.4. Airflow dags

- Same as the previous Airflow Dag we only change the Dag's description and the Shell Script to trigger in this case.

```

70 local_tz = pendulum.timezone('Europe/Paris')
71 START_DATE = datetime(2022, 7, 11, tzinfo=local_tz)
72
73 # Set only description, schedule_interval, start_date
74 # Example: the DAG is executed every day at 8am
75 dag = DAG(
76     V_DAG_ID,
77     default_args=default_args,
78     description='Wholesales Actuals Data Pipeline (Airflow test)',
79     schedule_interval='0 1 * * *',
80     start_date=START_DATE,
81     catchup=False
82 )

```

Figure 67: Madax Airflow DAG

```

84 # -----
85 #           Define and set DAG's tasks
86 # -----
87 # To use variables in task command, the command must be defined outside of the operator
88 UNXAPPLI = '/gpfs/user/sd01865/crf0a'
89 CMD_EPROG_AC = 'cd ' + UNXAPPLI + ';script/wholesales_actuals_data_pipeline.sh;'
90
91 with dag:
92     # Only BashOperator can be used
93     t1 = BashOperator(
94         # Task id must be unique in the DAG
95         task_id='wholesales_act',
96         # To use variables in command, the command must be defined outside of the operator
97         bash_command=str(CMD_EPROG_AC),
98     )
99
100
101 # Organize dependencies between tasks
102 t1

```

Figure 68: Madax Airflow BashOperator

## 5.5. The Result

- Here below are the values inserted in our BRC\_SD01865.WS\_ACT table.

Table 24: Madax Data Pipeline Result

	PROGRAM_COUNTRY	FAMILY	MONTHYEAR	MEASURE	VALUE	INSERTED_DATE	CYCLE
1	ALGERIE	1CM3	30/06/22	Wholesales	3311	12-JUL-22	202207
2	ALGERIE	1SX8	30/06/22	Wholesales	1925	12-JUL-22	202207
3	AFRIQUE	1PP2	30/06/22	Wholesales	1768	12-JUL-22	202207
4	TURQUIE	1PP8	30/06/22	Wholesales	1516	12-JUL-22	202207
5	AFRIQUE	1PP1	30/06/22	Wholesales	1483	12-JUL-22	202207
6	TURQUIE	1GJO	30/06/22	Wholesales	1445	12-JUL-22	202207
7	TURQUIE	1GME	30/06/22	Wholesales	1414	12-JUL-22	202207
8	ALGERIE	1CK9	30/06/22	Wholesales	1309	12-JUL-22	202207
9	TURQUIE DS	1CW8	30/06/22	Wholesales	962	12-JUL-22	202207
10	TURQUIE	1CW8	30/06/22	Wholesales	962	12-JUL-22	202207

🚨 The same goes for the retails

## 6. Conclusion

In this chapter, we were able to delve into the methods and tools used in our implementation to create our solution.

I was not able to do the ETL pipelines for all the KPIs, still need the CSF prevision KPIs (Wholesales/Retails/Market), the project is huge, and I had little time to execute everything and the data analysis and the search for the right data took the majority of the time, but I still have one month and a half to finish the project so hopefully I will be able to finish the project by then.

# General Conclusion

In this time and era, Business Intelligence with the support of Data Engineering made it easy for corporations to explore the massive data that wasn't as beneficial for them as it is in this era. The historical data could help the companies see patterns in their operation and even predict what could happen in the next 5 years.

A data department is a crucial part of every company, that's why there's a surge in the data professional demands in the last decade. Stellantis like many of these companies decided to jump on data the train.

One of Stellantis's important projects in the MEA region is the CARFLOW MEA DASHBOARDS project, this project aims to use the data from different heterogeneous sources to track their supply chain operations in the region of MEA

The problem with the project is that the ETL process to extract, transform and load the data in the data warehouse is a manual process which is a waste of time and energy and could lead to errors in the data as it is a manual process done by a human. Stellantis decided to change this and solve this problem, so they hired me as a data engineering intern to work on this problem.

The method I decided to follow to solve this is the following. First, I need to set the working environment and I was able to do that with the help of one of the data architects on the data team, he showed me how things work there. Second, I needed to study the "Supply Chain" business of Stellantis and that was possible with the help of the supply chain business team. Next, I had to analyze the existing solution and extract the business needs. Then, comes the conception and design step where I needed to do the data mapping, the conception of the possible solution, and the analysis of our data sources and destinations. Finally, I had to implement the solution with the tools provided by the Stellantis Data department and do a test. I was able to do the steps above and I was able to implement  $\frac{3}{4}$  of the solution as I still need to implement the CSF ETL that I already did the data mapping for.

One of the difficulties I faced during this internship is a short time and the struggle to get the right information from the right person. But my manager and my supervisor/professor helped me a lot to get through this and deliver a good result.

# References

- Groupe Calliope. (2022). *Qu'est-ce que la Business Intelligence dans l'entreprise ?* [online] Available at: <https://www.groupe-calliope.com/business-intelligence/qu-est-ce-que-business-intelligence/#:~:text=Quelle%20est%20> [Accessed 6 Jul. 2022].
- Lambert, N. (2020). *Pourquoi utiliser un ETL ? Avantages, enjeux et cas d'usage*. [online] Axysweb. Available at: <https://www.axysweb.com/pourquoi-utiliser-un-etl/> [Accessed 6 Jul. 2022].
- Smith, E. (2021). *World's fourth-largest carmaker rallies on first day of trade after \$52 billion merger*. [online] CNBC. Available at: <https://www.cnbc.com/2021/01/18/stellantis-rallies-on-first-day-of-trade-after-52-billion-merger.html> .
- Scherfner, E., Смирнов, Д., Nisay, A.A., Gayle, A.T., Otieno, W., Павлова, К., Alaimo, K.-K., Tramm, K., Jhaveri, N. and Gustafson, M. (n.d.). *Stellantis - Wiki*. [online] Golden. Available at: <https://golden.com/wiki/Stellantis-JY9E5E> [Accessed 6 Jul. 2022].
- chcom (2021). *Stellantis N.V.* [online] CompaniesHistory.com - The largest companies and brands in the world. Available at: <https://www.companieshistory.com/stellantis-n-v/> [Accessed 6 Jul. 2022].
- Howard, P.W. (n.d.). *FCA to change name to Stellantis after merger with PSA in 2021*. [online] Detroit Free Press. Available at: <https://eu.freep.com/story/money/cars/chrysler/2020/07/15/stellantis-fca-fiat-chrysler-peugeot-sa/5444259002/> [Accessed 6 Jul. 2022].
- www.nowcar.com. (n.d.). *NowCar | Fiat Chrysler Is Named Stellantis After Merger Is Finally Complete*. [online] Available at: <https://www.nowcar.com/blog/archive/fiat-chrysler-is-named-stellantis-after-merger-is-finally-complete/> [Accessed 6 Jul. 2022].
- www.leadersleague.com. (n.d.). *Peugeot and Fiat Chrysler Create Fourth Largest Automaker*. [online] Available at: <https://www.leadersleague.com/en/news/peugeot-and-fiat-chrysler-create-fourth-largest-automaker> [Accessed 6 Jul. 2022].
- AP NEWS. (2021). *Stellantis, Foxconn team up to make cars more connected*. [online] Available at: <https://apnews.com/article/europe-technology-business-3b076c99f3a70ee9bffa1b423a0803b1a> [Accessed 6 Jul. 2022].
- www.abcbourse.com. (n.d.). *Stellantis, secteur d'activité en bourse et sociétés comparables*. [online] Available at: <https://www.abcbourse.com/marches/secteur/STLAp> [Accessed 6 Jul. 2022].
- www.stellantis.com. (n.d.). *Full Year 2021 Results | Stellantis*. [online] Available at: <https://www.stellantis.com/en/news/press-releases/2022/february/full-year-2021-results>.
- JeuneAfrique.com. (n.d.). *Maroc : PSA débutera sa production le 2 juillet dans un secteur automobile en plein boom – Jeune Afrique*. [online] Available at: <https://www.jeuneafrique.com/555292/economie/maroc-psa-debutera-sa-production-le-2-juillet-dans-un-secteur-automobile-en-plein-boom/> [Accessed 14 Jul. 2022].
- Cegid. (n.d.). *Business Intelligence*. [online] Available at: <https://www.cegid.com/fr/glossaire/bi-business-intelligence/> [Accessed 7 Jul. 2022].
- L, B. (2018). *Data Warehouse (entrepôt de données) définition : qu'est-ce que c'est ?* [online] LeBigData.fr. Available at: <https://www.lebigdata.fr/data-warehouse-entrepot-donnees-definition> .

Softwares, L. (2014). *What is Web Data Extraction*. [online] Loginworks Softwares Pvt. Ltd. Available at: <https://www.loginworks.com/blogs/209-web-data-extraction/> .

Garnier, A. (n.d.). *Qu'est-ce que le processus ETL ?* [online] blog.hubspot.fr. Available at: <https://blog.hubspot.fr/marketing/extract-transform-load> [Accessed 7 Jul. 2022].

Softwares, L. (2014). *What is Web Data Extraction*. [online] Loginworks Softwares Pvt. Ltd. Available at: <https://www.loginworks.com/blogs/209-web-data-extraction/> .

StreamSets. (2022). *The Relationship Between ETL and Data Pipelines*. [online] Available at: <https://streamsets.com/blog/relationship-between-etl-and-data-pipelines/#:~:text=ETL%20refers%20to%20a%20set> [Accessed 7 Jul. 2022].

<sup>1</sup> Oracle.com. (2022). *SQL Developer*. [online] Available at: <https://www.oracle.com/database/sqldeveloper/> [Accessed 14 Jul. 2022].

Oracle.com. (2018). *SQL Developer History*. [online] Available at: <https://www.oracle.com/database/technologies/appdev/sqldev/history.html> [Accessed 14 Jul. 2022].

airflow.apache.org. (n.d.). *Apache Airflow Documentation — Airflow Documentation*. [online] Available at: <https://airflow.apache.org/docs/apache-airflow/stable/index.html> .

the.earth.li. (n.d.). *Using PuTTY*. [online] Available at: <https://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter3.html> [Accessed 14 Jul. 2022].