Rapport de Mini Projet

Génie Informatique

Performances des étudiants

Réalisé par :

Madani Ouail **Ezzouak Mohammed**

El Otmani Abderrahim **Benaouda Salma**

Bamhaouch Fatima-zahra

Encadré par

M. Yacine EL YOUNOUSSI **Encadrant ENSATé**

Année universitaire : 2024-2025

Résumé

Ce rapport présente le travail réalisé dans le cadre du mini-projet de Business Intelligence portant sur la conception et la mise en œuvre d'un système décisionnel académique. L'objectif principal de ce projet est de centraliser et d'analyser les données académiques des étudiants, actuellement dispersées dans diverses sources, afin d'offrir une vue unifiée permettant d'évaluer leurs performances et de suivre leur assiduité.

Le projet comprend la création d'une base de données opérationnelle, la génération de fichiers CSV comme sources de données supplémentaires, la conception d'un Data Warehouse en architecture dimensionnelle, ainsi que la mise en œuvre d'un processus ETL (Extraction, Transformation, Chargement) à l'aide de **Python** et de l'outil **Apache Airflow** pour intégrer les données de manière cohérente et fiable.

Un tableau de bord interactif a été développé à l'aide de **Power BI** pour permettre aux responsables académiques d'explorer les données consolidées grâce à des indicateurs clés de performance (KPI) et des cubes OLAP. Ce rapport détaille chaque étape du projet, y compris la modélisation du Data Warehouse, le processus ETL, ainsi que les outils et technologies utilisés pour les phases d'analyse et de reporting.

Ce projet a pour objectif d'améliorer la prise de décision en milieu académique, en facilitant le suivi des performances et l'identification des tendances permettant d'optimiser les stratégies pédagogiques.

Mots clés : gestion des inscriptions, doctorants, digitalisation, méthodologie agile, UML, Angular, Spring Boot, PostgreSQL, Cypress

Abstract

This report presents the work carried out as part of the Business Intelligence mini-project focused on the design and implementation of an academic decision-making system. The main objective of this project is to centralize and analyze student academic data, which is currently scattered across various sources, in order to provide a unified view for evaluating student performance and tracking attendance.

The project involved creating an operational database, generating CSV files as additional data sources, designing a dimensional Data Warehouse, and implementing an ETL (Extraction, Transformation, Loading) process using **Python** and **Apache Airflow** to integrate data in a consistent and reliable manner.

An interactive dashboard was developed using **Power BI** to allow academic staff to explore consolidated data through Key Performance Indicators (KPIs) and OLAP cubes. This report details each phase of the project, including the Data Warehouse modeling, the ETL process, and the tools and technologies used for data analysis and reporting.

The project aims to improve decision-making in academic environments by facilitating performance tracking and identifying trends to optimize educational strategies.

Key words: Business Intelligence, Data Warehouse, ETL, Python, Apache Airflow, Reporting, Power BI, OLAP, Decision-making analysis, Interactive dashboard, KPI, Academic performance.

Liste des tableaux

Liste des figures

Table des matières

Table des matières

Contexte du projet

Dans le domaine académique, la gestion et l'analyse des performances des étudiants sont devenues des enjeux majeurs pour améliorer la qualité de l'enseignement et assurer un suivi personnalisé. Les établissements d'enseignement supérieur disposent souvent de vastes quantités de données, réparties dans divers systèmes tels que des bases de données internes, des fichiers CSV, ou des plateformes numériques d'apprentissage. Cependant, cette fragmentation des données constitue une barrière majeure à l'exploitation efficace de ces informations.

Les données académiques, incluant les moyennes annuelles, les résultats obtenus par module, et les informations sur l'assiduité, sont essentielles pour évaluer les progrès des étudiants et identifier les éventuelles difficultés. Actuellement, ces données sont souvent sous-utilisées en raison de leur dispersion et de l'absence d'un cadre unifié pour leur analyse. Ce manque de centralisation complique la prise de décisions informées, que ce soit pour les enseignants, les responsables pédagogiques, ou les administrations.

Dans ce contexte, le projet se concentre sur la conception et la mise en œuvre d'une solution intégrant un entrepôt de données (Data Warehouse). Ce dernier sera alimenté par un processus ETL (Extraction, Transformation, Chargement) qui assurera la collecte et la transformation des données issues de multiples sources opérationnelles. En s'appuyant sur ces fondations, le projet vise à offrir des outils analytiques dynamiques, tels que des tableaux de bord et des rapports détaillés, pour répondre aux besoins des parties prenantes dans leur mission d'accompagnement pédagogique.

Chapitre 1 : Architecture et Mise en Place du Système de Données

I. Conception de la base de données opérationnelle

1. Modèle Conceptuel

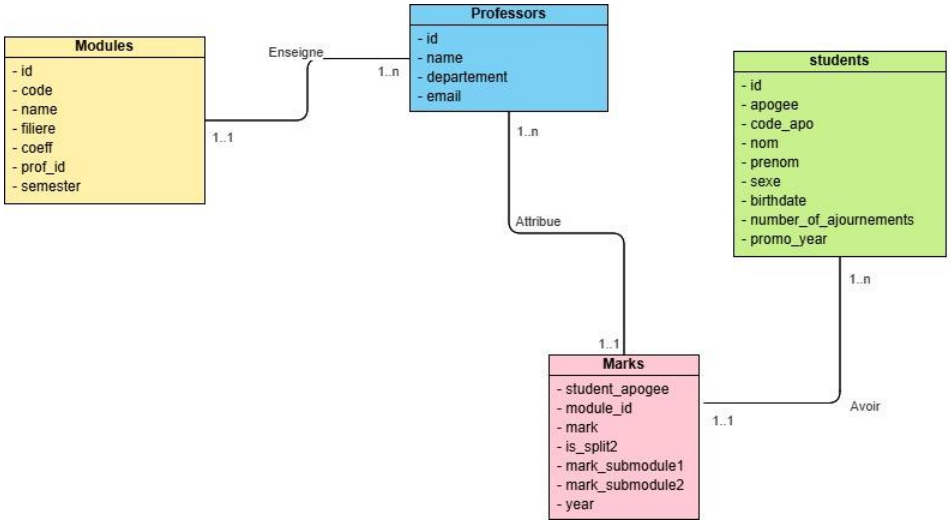

Figure 1: Modèle conceptuel de données (MCD)

L'objectif du modèle conceptuel de données dans le cadre de ce projet est de structurer et d'organiser les informations académiques de manière cohérente afin de centraliser les données des étudiants, des enseignants, des modules et des résultats. Ce modèle permet de définir les entités principales et les relations entre elles afin d'assurer l'intégrité et la qualité des données nécessaires pour les analyses décisionnelles.

2. Dictionnaire de données

Le dictionnaire de données a pour but de décrire les informations manipulées au sein du système décisionnel mis en place dans le cadre de ce projet. Il permet de définir de manière précise les données essentielles, en détaillant leur structure, leur type, et les différentes contraintes associées. Cette documentation vise à garantir la cohérence et la compréhension des données tout au long des étapes de collecte, transformation et analyse.

Les informations décrites dans ce dictionnaire assurent une meilleure gestion des données académiques et facilitent leur exploitation pour le suivi des performances des étudiants.

2.1 Table : students

Cette table contient les informations personnelles des étudiants inscrits. Chaque étudiant est identifié de manière unique grâce à son numéro Apogée.

Attribut	Type	Description	Contraintes
Id	SERIAL	Identifiant unique de l'étudiant	Clé primaire
Apogee	BIGINT	Numéro Apogée de l'étudiant	Unique, Non null
Nom	VARCHAR (100)	Nom de l'étudiant	Non null
Prenom	VARCHAR (100)	Prénom de l'étudiant	Non null
Sexe	CHAR (1)	Sexe de l'étudiant (M/F)	Non null
Birthdate	DATE	Date de naissance	Non null
number_of_ajournements	INTEGER	Nombre d'ajournements	Valeurs positives
promo_year	INTEGER	Année de promotion	Non null

Tableau 1: Dictionnaire de données de la table - students

2.2 Table : professors

Cette table regroupe les informations des enseignants responsables des modules d'enseignement.

Attribut	Type	Description	Contraintes
Id	SERIAL	Identifiant unique de l'enseignant	Clé primaire
Name	VARCHAR (255)	Nom complet de l'enseignant	Non null
department	VARCHAR (100)	Département auquel l'enseignant appartient	Non null
Email	VARCHAR (100)	Adresse e-mail de l'enseignant	Unique, Non null

Tableau 2:Dictionnaire de données de la table – professors

2.3 Table : modules

Cette table liste les différents modules enseignés dans le cadre du parcours académique des étudiants.

Attribut	Type	Description	Contraintes
Id	SERIAL	Identifiant unique du module	Clé primaire
Code	VARCHAR (50)	Code du module	Unique, Non null
Name	VARCHAR (255)	Nom du module	Non null
Filiere	VARCHAR (50)	Filière d'enseignement	Non null
Coeff	DOUBLE PRECISION	Coefficient du module	Valeurs positives
prof_id	INTEGER	Identifiant du professeur responsable	Clé étrangère vers professors(id)
Semester	VARCHAR (5)	Semestre d'enseignement	Non null

Tableau 3: Dictionnaire de données de la table - modules

2.4 Table : marks

Cette table contient les notes obtenues par les étudiants pour chaque module suivi.

Attribut	Type	Description	Contraintes
student_apogee	BIGINT	Numéro Apogée de l'étudiant	Clé étrangère vers students(apogee)
module_id	INTEGER	Identifiant du module	Clé étrangère vers modules(id)
Mark	DOUBLE PRECISION	Note obtenue par l'étudiant	Valeurs entre 0 et 20
is_split2	BOOLEAN	Indicateur si le module est divisé	Par défaut : FALSE
mark_submodule1	DOUBLE PRECISION	Note du premier sous-module	Valeurs entre 0 et 20
mark_submodule2	DOUBLE PRECISION	Note du second sous-module	Valeurs entre 0 et 20
Year	VARCHAR (9)	Année académique	Non null

Tableau 4: Dictionnaire de données de la table – marks

Conclusion

La conception de la base de données opérationnelle constitue une étape cruciale pour structurer et organiser les données académiques de manière cohérente et centralisée. Le modèle conceptuel de données (MCD) élaboré dans ce projet assure une représentation claire des entités principales — étudiants, enseignants, modules et résultats — ainsi que des relations qui les unissent, garantissant l'intégrité et la qualité des informations pour les analyses décisionnelles.

Le dictionnaire de données documente précisément les tables essentielles et leurs attributs, tels que students, professors, modules, et marks. Cette documentation garantit la cohérence et la compréhension des données tout au long du processus de collecte, transformation et exploitation. Chaque table répond à des besoins spécifiques, par exemple :

- Students centralise les informations personnelles et académiques des étudiants.
- Professors regroupe les informations des enseignants responsables.
- Modules détaille les caractéristiques des cours enseignés.
- Marks enregistre les performances des étudiants, avec des mécanismes pour gérer les notes des sous-modules.

En résumé, cette base de données opérationnelle offre une fondation solide pour l'intégration des données académiques dans un système décisionnel, facilitant les analyses futures et permettant une gestion efficace et précise des informations académiques.

II. Génération de fichiers CSV

Dans le cadre d'une analyse BI (Business Intelligence) centrée sur les comportements étudiants, nous avons conçu un modèle détaillé simulant les données liées aux absences et aux performances académiques. Cette modélisation repose sur une approche méthodique visant à refléter les dynamiques réelles observées dans les environnements éducatifs. Nous détaillons ici les étapes majeures : la génération des profils étudiants, la création des données d'absences et l'attribution des notes académiques.

1. Génération des Profils Étudiants

La première étape consiste à créer un ensemble réaliste de profils d'étudiants. Chaque profil inclut des informations sociodémographiques et des attributs comportementaux, notamment un **biais d'absence**, représentant la tendance individuelle à être absent(e).

Composants du Profile Étudiant

- **Identité** : Un identifiant unique, nom et prénom, associés à un sexe. Ces données sont générées pour refléter une diversité plausible, basée sur des contextes socioculturels (par exemple, des noms marocains).
- **Année de naissance** : Calculée pour représenter l'âge typique à l'université (19-22 ans).
- **Biais d'absence** : Un score entre 1 et 9, où 1 représente une grande assiduité et 9 une forte tendance à l'absentéisme.

Algorithme de génération

1. Générer un identifiant unique pour chaque étudiant.
2. Attribuer aléatoirement un prénom, un nom et un sexe, en utilisant des listes pondérées pour refléter des distributions réalistes.
3. Déterminer une année de naissance plausible en fonction de l'année de promotion.
4. Assigner un biais d'absence tiré aléatoirement d'une distribution uniforme ou biaisée pour modéliser différents profils.

Cette méthodologie garantit une diversité dans les profils, essentielle pour simuler des interactions variées entre assiduité et performances.

```python
import random
from faker import Faker

male_first_names = ["Mohammed", "Ahmed", "Youssef", "Hamza"]
female_first_names = ["Rania", "Hana", "Malak"]
last_names = ["El-Badri", "El-Khayat", "Amrani"]

fake = Faker()

def generate_student(student_id, promo_year):
    sexe = random.choice(['M', 'F'])
    prenom = random.choice(male_first_names) if sexe == 'M' else random.choice(female_first_names)
    nom = random.choice(last_names)
    birth_year = promo_year - 19  # Typiquement 19 ans à l'entrée en université
    absence_bias = random.randint(1, 9)  # Biais influençant les absences
    return {
        "id": student_id,
        "nom": nom,
        "prenom": prenom,
        "sexe": sexe,
        "birth_year": birth_year,
        "absence_bias": absence_bias
    }
```

Figure 2 : Extrait simplifié d'algorithme de génération des profiles des étudiants

2. Génération des Données d'Absences

Les absences constituent un indicateur clé dans l'analyse des performances scolaires. Pour cette simulation, les absences ont été modélisées en tenant compte de **facteurs individuels** (comme le biais d'absence de l'étudiant) et **contextuels** (comme le type de cours ou le module étudié).

Modèle d'absences

- **Probabilité de présence** : Calculée en partant d'une probabilité de base élevée (95 %), ajustée par :
    - Le biais d'absence de l'étudiant (comportement individuel).
    - Un biais lié au module (difficulté perçue ou engagement requis).

- **Différence entre CM et TP** : Les cours magistraux (CM) ont une probabilité légèrement plus basse d'assiduité que les travaux pratiques (TP), qui nécessitent souvent une participation plus active.

Algorithme de génération

1. Définir une probabilité de présence initiale (par exemple, 95 %).
2. Réduire cette probabilité en fonction des biais d'absence de l'étudiant et du module (par exemple, une réduction de 5 % pour chaque niveau de biais).
3. Simuler chaque session de cours en tirant une valeur aléatoire pour déterminer la présence ou l'absence.
4. Répéter le processus pour chaque étudiant et chaque session.

Ce modèle produit une matrice binaire où chaque colonne représente une session (CM ou TP) et chaque ligne un étudiant, avec 1 pour présence et 0 pour absence.

```python
def generate_absence(cour_range, tp_range, student_bias, module_bias):
    base_attendance = 0.95
    attendance_probability = base_attendance - ((student_bias - 1) / 8 * 0.6) - ((module_bias - 1) / 8 * 0.5)
    attendance_probability = max(min(attendance_probability, 1.0), 0.0)

    absence_data = {}
    for i in cour_range:
        cour_attended = random.random() < attendance_probability
        absence_data[f'cour{i}'] = int(cour_attended)
        tp_attended = cour_attended or (random.random() < attendance_probability * 1.1)
        absence_data[f'tp{i}'] = int(tp_attended)
    return absence_data
```

Figure 3 : Extrait simplifié d'algorithme de génération des absences.

3. Attribution des Notes Académiques

Les performances académiques sont influencées par les absences, mais également par d'autres facteurs, comme les capacités intrinsèques de l'étudiant ou les ajournements passés. La simulation des notes intègre ces éléments pour générer des données réalistes.

- **Approche conceptuelle**

1. **Définir une note médiane** pour chaque module, représentant la moyenne attendue (par exemple, 12/20).

2. **Ajuster la note de base** en fonction :

    - Des absences : Une forte absence réduite significativement la note.

    - Des ajournements : Chaque ajournement passé diminue légèrement la moyenne attendue.

3. **Introduire des variations aléatoires** autour de la moyenne ajustée pour modéliser les différences individuelles.

4. **Gérer les cas spécifiques** :

    - Étudiants ayant des notes très basses (<8) : Augmentation marginale pour éviter des extrêmes irréalistes.

    - Étudiants proches des seuils critiques (par exemple, 12/20) : Ajustement pour refléter une politique académique fréquente (légères augmentations).

    - Cas de rattrapage : Les notes inférieures à 12 sont plafonnées après amélioration.

- **Algorithme simplifié**

1. Départ de la note médiane du module.

2. Appliquer une pénalité proportionnelle aux absences.

3. Réduire la note en fonction du nombre d'ajournements passés.

4. Introduire une variation aléatoire pour refléter les capacités individuelles.

5. Ajuster les cas critiques ou particuliers (rattrapage, seuils critiques).

Cette approche garantit une distribution réaliste des notes tout en reflétant les pratiques académiques habituelles.

```python
def generate_marks(student, module):
    median = module["median_mark"]
    absence_bias = student["absence_bias"]
    ajournement = student["Number of Ajournements"]

    mean = median - (ajournement * 0.5) - (absence_bias * 0.3)
    std_dev = 2.5
    mark = round(max(0, min(20, np.random.normal(mean, std_dev))), 2)

    if mark < 8: mark += 1
    elif 12 <= mark < 14: mark += 0.25

    rattrapage = mark < 12
    if rattrapage:
        mark = round(min(mark + np.random.uniform(0, 3), 12), 2)
    return mark, rattrapage
```

Figure 4 : Extrait simplifié d'algorithme de génération des notes.

4. Interprétation et Exploitation des Données

Les données générées offrent une richesse analytique permettant d'explorer plusieurs axes :

1. **Corrélation entre absences et performances** : Identifier les seuils critiques d'absences ayant un impact significatif sur les notes.

2. **Analyse par module** : Comprendre quels modules sont plus affectés par l'absentéisme.

3. **Étude des comportements** : Déceler des profils d'étudiants (par exemple, "assidus mais performants moyens" ou "absents fréquents avec bonnes notes").

Ces données sont idéales pour alimenter des outils BI, comme Power BI ou Tableau, permettant de visualiser et d'interpréter les relations complexes entre comportement et performance. Par exemple, une heatmap des absences par module ou une distribution des notes en fonction du taux d'assiduité peut révéler des tendances intéressantes.

Conclusion

La simulation des données académiques, incluant absences et performances, constitue une base solide pour des analyses approfondies. Les biais intégrés reflètent des scénarios réalistes, et l'approche méthodique garantit la pertinence des résultats pour des applications éducatives ou administratives. Ces données, associées à des outils BI performants, permettent de mieux comprendre et éventuellement anticiper les dynamiques académiques.

III. Conception de la data Warehouse

1. Architecture dimensionnelle (Schéma en étoile)

La conception de notre Data Warehouse repose sur une **architecture dimensionnelle** en schéma en étoile, qui permet de structurer les données de manière optimale pour les analyses décisionnelles. Ce type d'architecture est particulièrement adapté pour les **analyses rapides et efficaces**, car il centralise les mesures chiffrées dans une table de faits, entourée de plusieurs tables de dimensions fournissant le contexte des analyses.

La **table de faits** contient les informations essentielles à analyser, telles que les performances académiques des étudiants (notes), tandis que les **tables de dimensions** décrivent les entités contextuelles comme les étudiants, les modules, les professeurs, et le temps.

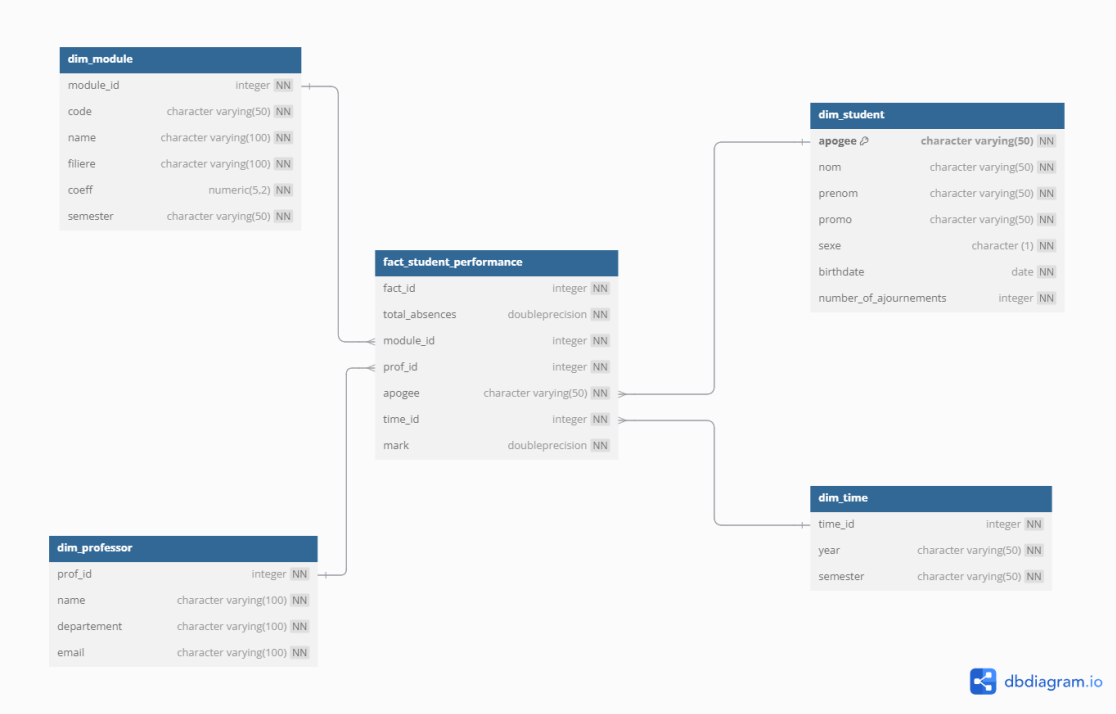

Figure 5: Architecture dimensionnelle du Data Warehouse (Schéma en étoile)

2. Table de faits : fact_student_performance

La table de faits centralise les **performances académiques des étudiants**. Elle contient des informations sur :

- **Les notes obtenues** par les étudiants dans chaque module.
- **Les modules suivis**, avec le détail des professeurs et des périodes d'enseignement.

Attribut	Type	Description
fact_id	INTEGER	Identifiant unique pour chaque enregistrement de performance
total_absences	DOUBLE PRECISION	Nombre total de présence de l'étudiant pour le module.
module_id	INTEGER	Clé étrangère référençant dim_module(module_id)
prof_id	INTEGER	Clé étrangère référençant dim_professor(prof_id)
Apogee	VARCHAR (50)	Clé étrangère référençant dim_student(apogee)
time_id	INTEGER	Clé étrangère référençant dim_time(time_id)
Mark	DOUBLE PRECISION	Note ou grade obtenu par l'étudiant dans le module.

Tableau 5 : Description de la table de faits : fact_student_performance

3. Tables de dimension

3.1 Table de Dimension : dim_professor

Attribut	Type	Description
prof_id	INTEGER	Identifiant unique pour chaque professeur (Clé Primaire)
name	VARCHAR (100)	Nom complet du professeur
departement	VARCHAR (100)	Département auquel le professeur appartient
Email	VARCHAR (100)	Adresse e-mail du professeur

Tableau 6 : Description de la table de dimension : dim_professor

3.2 Table de dimension : dim_module

Attribut	Type	Description
module_id	INTEGER	Identifiant unique pour chaque module (Clé Primaire)
Code	VARCHAR (50)	Code représentant le module
name	VARCHAR (100)	Nom complet du module
filiere	VARCHAR (100)	Filière associée au module
Coeff	NUMERIC (5,2)	Coefficient du module
Semester	VARCHAR (50)	Semestre pendant lequel le module est enseigné

Tableau 7 : Description de la table de dimension : dim_module

3.3 Table de dimension : dim_student

Attribut	Type	Description
Apogee	VARCHAR (50)	Identifiant unique pour chaque étudiant (Clé Primaire)
Nom	VARCHAR (50)	Nom de famille de l'étudiant

Prenom	VARCHAR (50)	Prénom de l'étudiant
Promo	VARCHAR (50)	Année de promotion ou promotion de l'étudiant
Sexe	CHAR (1)	Genre de l'étudiant ('M' pour Masculin, 'F' pour Féminin).
Birthdate	DATE	Date de naissance de l'étudiant
number_of_ajournements	INTEGER	Nombre de fois où l'étudiant a été ajourné

Tableau 8 : Description de la table de dimension : dim_student

3.4 Table de dimension : dim_time

Attribut	Type	Description
Time_id	INTEGER	Identifiant unique pour chaque période
Year	VARCHAR (50)	Année académique
Semester	VARCHAR (50)	Semestre

Tableau 9 : Description de la table de dimension : dim_time

Conclusion

La conception du Data Warehouse repose sur une architecture dimensionnelle en schéma en étoile, qui répond aux besoins spécifiques des analyses décisionnelles académiques. Ce modèle structure les données de manière claire et efficace, avec une table de faits centralisant les mesures chiffrées, telles que les performances académiques des étudiants, et des tables de dimensions fournissant le contexte analytique (étudiants, modules, professeurs, et périodes).

La **table de faits**, nommée *fact_student_performance*, joue un rôle central en agrégeant les données critiques, telles que les notes, les absences, et les relations avec les dimensions associées. Les **tables de dimensions** enrichissent l'analyse en détaillant :

- Les informations des professeurs (*dim_professor*),
- Les caractéristiques des modules enseignés (*dim_module*),
- Les données démographiques et académiques des étudiants (*dim_student*),

- Les aspects temporels des périodes académiques (*dim_time*).

Ce schéma en étoile assure une navigation fluide entre les données et permet des analyses multidimensionnelles rapides et intuitives. Il offre également une flexibilité pour des explorations futures, comme l'identification des tendances académiques, la comparaison des performances par semestre, ou l'évaluation des enseignants et des modules.

En conclusion, cette conception du Data Warehouse fournit une base robuste et performante pour la Business Intelligence, facilitant des décisions stratégiques basées sur des données académiques fiables et bien structurées.

Chapitre 2 : Processus ETL

I. Processus ETL

Le workflow ETL (Extraction, Transformation, Chargement) est le cœur du projet. Il permet de collecter, nettoyer et intégrer les données provenant de plusieurs sources dans l'entrepôt de données (Data Warehouse). Ce processus est orchestré à l'aide d'Apache Airflow, déployé dans un environnement Docker pour assurer la scalabilité et la cohérence. La figure ci-dessous illustre le workflow ETL tel qu'il est géré par Apache Airflow.

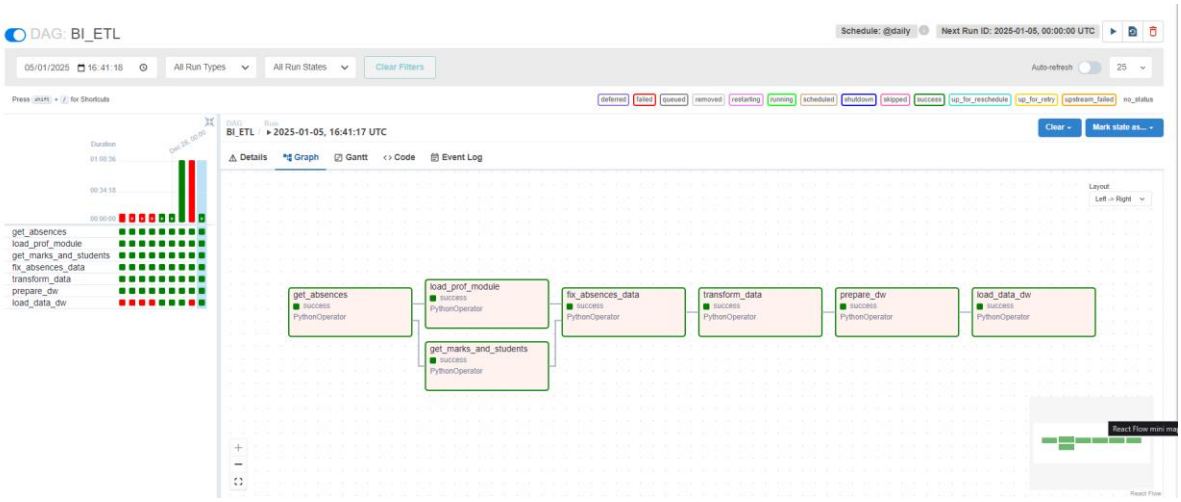

Figure 6:Workflow ETL orchestré par Apache Airflow

La figure ci-dessus montre le DAG (Directed Acyclic Graph) du workflow ETL dans Apache Airflow. Chaque tâche du processus ETL (Extraction, Transformation, Chargement) est représentée sous forme de nœuds, et les dépendances entre les tâches sont indiquées par des flèches. Ce workflow est déployé dans un environnement Docker pour assurer la portabilité et la cohérence.

1. Contexte et Choix Technologique

Le projet impliquait la gestion et l'analyse d'un grand nombre de fichiers CSV, ce qui a rapidement posé des défis en termes de complexité et de scalabilité. L'utilisation d'outils traditionnels comme Pentaho s'est avérée inadaptée, car bien qu'il offre des fonctionnalités pour traiter plusieurs fichiers CSV, leur configuration peut être complexe et nécessiter une intervention manuelle pour chaque fichier.

Pour pallier ces limitations, nous avons adopté une solution basée sur Apache Airflow et Python pour les raisons suivantes :

- **Automatisation avancée :** Airflow permet de définir des pipelines dynamiques et modulaires, facilement extensibles.
- **Personnalisation :** Python offre une grande flexibilité pour manipuler et transformer les données complexes.
- **Scalabilité :** Grâce à Docker, l'architecture est distribuée et capable de gérer un grand volume de fichiers CSV efficacement.
- **Surveillance en temps réel :** Airflow fournit une interface utilisateur intuitive pour surveiller et gérer l'exécution des tâches.
- **Gestion des erreurs robuste :** Les tâches échouées peuvent être relancées sans nécessiter une reprise complète du processus.

Figure 7:Apache Airflow Logo

2. Extraction

La phase d'extraction consiste à récupérer les données provenant de :

- Fichiers CSV : Utilisés pour les absences, les professeurs et les modules.
- Base de données PostgreSQL : Utilisée pour les informations sur les étudiants et leurs notes.

Tâches de la phase d'extraction :

1. get_absences_module_promo:

- Cette tâche lit les fichiers CSV contenant les données d'absences des étudiants par promotion et par module.
- Elle calcule le nombre total d'absences pour chaque étudiant et stocke les résultats dans un dictionnaire de DataFrames.
- Les données sont poussées dans XCom (mécanisme de communication inter-tâches d'Airflow) pour être utilisées dans les tâches suivantes.

2. extract_prof_module :

- Cette tâche extrait les données des fichiers CSV (prof.csv et modules.csv) à l'aide de Pandas.
- Elle fusionne les données des professeurs et des modules en se basant sur l'identifiant du professeur (profID).
- Le DataFrame résultant est poussé dans XCom pour un traitement ultérieur.

3. extract_students_marks :

- Cette tâche se connecte à la base de données opérationnelle PostgreSQL en utilisant psycopg2.
- Elle récupère les données des tables students et marks.
- Les données sont stockées dans des DataFrames et poussées dans XCom pour la phase de transformation.

Défis de l'extraction :

- Hétérogénéité des sources de données : Les données proviennent à la fois d'une base de données relationnelle et de fichiers CSV, nécessitant des méthodes d'extraction différentes.
- Performance : Les volumes de données importants peuvent ralentir le processus d'extraction, en particulier lors de la lecture de fichiers CSV.

3. Transformation

La phase de transformation consiste à nettoyer, agréger et enrichir les données extraites pour les préparer au chargement dans l'entrepôt de données.

Tâches de la phase de transformation :

1. fix_absences_data:

- Cette tâche consolide les données d'absences en fusionnant les sous-modules (par exemple, "Langues et Communication I1" et "Langues et Communication I2") en un seul module.
- Elle calcule le total des absences pour chaque étudiant et prépare les données pour le chargement.

2. transform_data:

- Cette tâche fusionne les données des étudiants, des notes et des absences dans un seul DataFrame.
- Elle effectue le nettoyage des données, comme la suppression de colonnes inutiles et la gestion des valeurs manquantes.

- Les données transformées sont poussées dans XCom pour le chargement dans l'entrepôt de données.

4. Chargement

La phase de chargement consiste à insérer les données transformées dans l'entrepôt de données, qui est également hébergé sur PostgreSQL.

Tâches de la phase de chargement :

1. **prepare_dw:**
   - Cette tâche prépare l'entrepôt de données en créant les tables nécessaires (dim_student, dim_module, dim_professor, dim_time, et fact_student_performance).
   - Elle s'assure que les tables sont correctement structurées avec des clés primaires, des clés étrangères et des contraintes.
2. **load_data_dw**:
   - Cette tâche charge les données transformées dans les tables de l'entrepôt de données.
   - Elle insère les données dans les tables de dimensions (dim_student, dim_module, dim_professor, dim_time) et dans la table de faits (fact_student_performance).
   - Des contraintes de clé étrangère sont ajoutées pour maintenir l'intégrité des données.

Défis du chargement :

- **Performance :** Le chargement de données massives dans plusieurs tables tout en respectant les contraintes relationnelles.
- **Intégrité des données :** Prévention des doublons et gestion des relations complexes.

5. Orchestration avec Apache Airflow et Docker :

L'ensemble du workflow ETL est orchestré à l'aide d'Apache Airflow, déployé dans un environnement Docker pour assurer la scalabilité et la cohérence. La figure 1 montre le DAG Airflow qui gère ce processus.

Composants clés de la configuration Dockerisée d'Airflow :

1. **Conteneur PostgreSQL** : Stocke les métadonnées d'Airflow (par exemple, les DAGs, les statuts des tâches).
2. **Conteneur Redis** : Joue le rôle de courtier en messages pour la communication entre le scheduler et les workers.

3. **Conteneur Airflow Webserver** : Fournit l'interface utilisateur pour surveiller et gérer les workflows.
4. **Conteneur Airflow Scheduler** : Planifie et déclenche les tâches en fonction des DAGs.
5. **Conteneur Airflow Worker** : Exécute les tâches définies dans les DAGs.

Avantages de l'utilisation de Docker :

- **Cohérence** : Le même environnement Airflow peut être répliqué à travers les étapes de développement, de test et de production.
- **Facilité de déploiement** : Docker Compose permet de démarrer et d'arrêter l'ensemble de la configuration Airflow avec une seule commande (docker-compose up).
- **Efficacité des ressources** : Les conteneurs sont légers et partagent le noyau du système hôte, réduisant la surcharge des ressources.
- **Isolation** : Chaque composant fonctionne dans son propre conteneur, évitant les conflits et assurant la stabilité.

Figure 8: Docker container

6. Outils et Bibliothèques Utilisés :

Pour implémenter ce workflow ETL, plusieurs **outils et bibliothèques Python** ont été utilisés :

1. **Apache Airflow** : Pour l'orchestration des tâches ETL, la planification des workflows et la surveillance en temps réel.
2. **Pandas** : Une bibliothèque Python puissante pour la manipulation et l'analyse des données. Elle a été utilisée pour lire les fichiers CSV, effectuer des transformations sur les données (nettoyage, agrégation, fusion) et préparer les données pour le chargement.
3. **Psycopg2** : Une bibliothèque Python pour interagir avec les bases de données PostgreSQL. Elle a été utilisée pour extraire les données de la base de données opérationnelle et charger les données transformées dans l'entrepôt de données.
4. **Docker** : Pour containeriser l'environnement Airflow, garantissant une configuration cohérente et portable entre les différents environnements (développement, test, production).

5. **XCom** : Un mécanisme intégré à Airflow pour partager des données entre les tâches. Il a été utilisé pour passer des données intermédiaires (comme les DataFrames) entre les étapes d'extraction, de transformation et de chargement.

Ces outils, combinés à la puissance d'Apache Airflow, ont permis de créer un workflow ETL robuste, automatisé et facile à surveiller, tout en garantissant la cohérence et la qualité des données.

Conclusion

Le workflow ETL est un élément essentiel du projet, permettant la collecte, le nettoyage et l'intégration des données provenant de multiples sources dans l'entrepôt de données. En utilisant Apache Airflow et Docker, le workflow est automatisé, scalable et cohérent. L'utilisation de Docker garantit que l'environnement Airflow est portable et facile à déployer, tandis qu'Airflow offre des capacités robustes d'orchestration et de surveillance. Cette combinaison d'outils et de techniques assure que le processus ETL est efficace, fiable et prêt à gérer une croissance future des données.

Chapitre 3 : Analyse et exploitation de données

I. Reporting

1. Introduction

Ce tableau de bord interactif est conçu pour fournir une vision globale des données académiques liées aux étudiants, modules, professeurs et performances dans un cadre éducatif. Structuré en cinq pages, il permet une exploration approfondie et une analyse détaillée des résultats et des comportements. Voici une présentation de la première page ainsi que l'analyse des visualisations qu'elle contient.

2. Description des visualisations et leur utilité

Figure 9 : première page du tableau de bord.

- **Distribution de Genre**
  - **Description** : Un graphique circulaire montre la répartition des étudiants par genre (masculin et féminin).
  - **Utilité** : Identifier l'équilibre ou les disparités entre les genres au sein des promotions, essentiel pour des initiatives inclusives.

- **KPI (Key Performance Indicators)**
  - **Nombre des Étudiants** : Affiche le total des étudiants inscrits.
  - **Nombre de Modules** : Indique le nombre total de modules enseignés.
  - **Nombre de Professeurs** : Représente les professeurs impliqués dans l'enseignement.
  - **Moyenne de Présence** : Montre le pourcentage moyen de présence en séances.
  - **Utilité** : Ces indicateurs clés offrent une vue rapide de l'envergure des données, facilitant le suivi des statistiques principales.

- **Note Moyenne par Semestre**
  - **Description** : Une courbe linéaire montre l'évolution des notes moyennes par semestre.
  - **Analyse** : Une tendance stable ou des fluctuations peuvent indiquer l'efficacité des méthodes pédagogiques ou la difficulté des modules.

- **Pourcentage d'Étudiants Validant par Module**
  - **Description** : Un histogramme affiche les taux de réussite par module, répartis par semestre.
  - **Analyse** : Les modules avec des taux élevés (ex. : "Programmation Réseaux et Sécurité") montrent une meilleure maîtrise des étudiants. Les taux plus faibles peuvent signaler des modules plus difficiles nécessitant un accompagnement supplémentaire.

- **Nombre d'Étudiants par Catégorie de Notes**
  - **Description** : Un graphique en barres représente la répartition des étudiants selon différentes plages de notes (7-10, 10-12, etc.) par module.
  - **Analyse** : Cette visualisation met en évidence les modules où les performances sont globalement meilleures ou plus faibles.

- **Présences par Professeur**
  - **Description** : Un diagramme circulaire illustre la répartition des présences par professeur.
  - **Analyse** : Permet de détecter les disparités dans l'assiduité des étudiants selon les professeurs, potentiellement liées à l'attractivité des cours.

3. Analyse des Résultats

1. **Équilibre de Genre** : Une répartition équilibrée est observée (49.52% F, 50.48% M), ce qui reflète une parité au sein des étudiants. Cela indique une absence de disparité majeure dans les inscriptions entre hommes et femmes.

2. **Présence en Séance** : Avec une moyenne de 71.68%, le taux de présence est relativement satisfaisant, mais il laisse une marge d'amélioration pour atteindre une meilleure implication des étudiants.
3. **Performance Académique** : Les modules avec des taux de réussite élevés, tels que "Programmation Réseaux et Sécurité", montrent un bon niveau de compréhension. Cependant, certains modules avec des taux plus faibles nécessitent une analyse approfondie pour identifier les problèmes rencontrés.
4. **Assiduité des Étudiants** : L'analyse des présences par professeur révèle des disparités. Par exemple, certains professeurs attirent significativement plus d'étudiants, ce qui peut être lié à leur style d'enseignement ou à la nature du cours.

La deuxième page du tableau de bord présente une analyse approfondie des performances académiques des étudiants en lien avec leur présence et leurs notes. Elle s'articule autour de plusieurs visualisations complémentaires, permettant une compréhension globale des données au fil des années universitaires et par module.

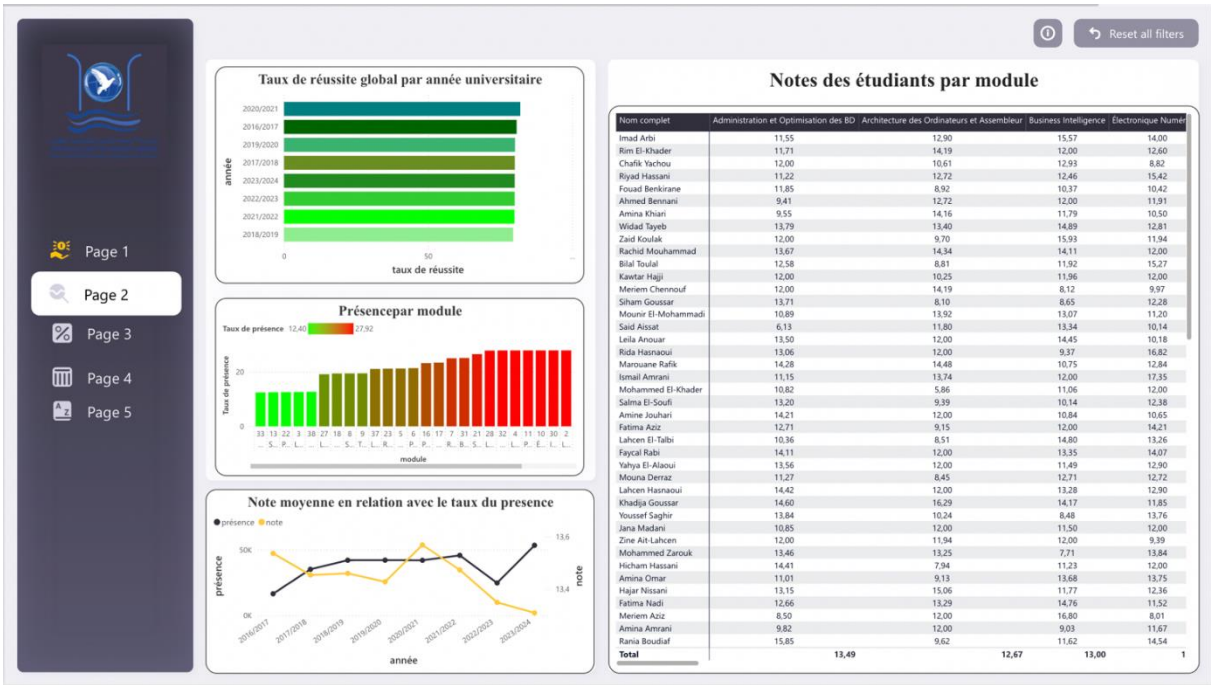

Figure 10 : deuxième page du tableau de bord.

4. Description des Visualisations et Leur Utilité

1. **Taux de Réussite Global par Année Universitaire**

- **Description** : Un histogramme vertical montre le pourcentage global de réussite des étudiants sur plusieurs années universitaires (ex. : 2016/2017 à 2023/2024).

- **Utilité** : Identifier les tendances de performance générale et évaluer si les taux augmentent ou diminuent. Cela peut également permettre de corréler les résultats avec des changements dans la pédagogie ou les curriculums.

2. **Présence par Module**

- **Description** : Un graphique en barres horizontales, avec une palette de couleurs allant du vert (taux de présence faible) au rouge (taux de présence élevé), illustre le taux de présence des étudiants pour chaque module.
- **Analyse** : Met en évidence les modules où l'assiduité est meilleure ou plus faible, offrant des opportunités d'intervention pour améliorer la présence.

3. **Notes Moyennes en Relation avec le Taux de Présence**

- **Description** : Un graphique linéaire superpose l'évolution des notes moyennes des étudiants et de leur présence au fil des années universitaires.
- **Analyse** : Vérifie si une corrélation existe entre la présence et la performance académique. Une forte corrélation positive indiquerait que la présence en classe est un facteur clé de la réussite.

4. **Tableau des Notes des Étudiants par Module**

- **Description** : Un tableau détaillé liste les notes obtenues par chaque étudiant pour différents modules (par exemple : "Administration et Optimisation des BD", "Architecture des Ordinateurs et Assembleur", etc.).
- **Utilité** : Permet d'identifier les étudiants ou les modules nécessitant une attention particulière. Une analyse détaillée par individu ou groupe devient ainsi possible.

Analyse des Résultats

1. **Taux de Réussite Global** :

- Les taux varient entre les années. Une amélioration est visible pour 2023/2024 (plus de 80%), ce qui peut refléter un enseignement plus efficace ou une meilleure implication des étudiants. Les années précédentes montrent des fluctuations qui méritent d'être explorées (ex. : 2018/2019 plus faible).

2. **Présence par Module** :

- Les modules avec des taux de présence faibles (en vert) pourraient nécessiter une révision des horaires, des méthodes d'enseignement ou une meilleure communication. Les modules avec une forte présence (en rouge) témoignent d'un intérêt accru des étudiants ou de meilleures pratiques pédagogiques.

3. **Corrélation Présence-Performance** :

- Le graphique montre une légère corrélation entre la présence et les notes, bien que des fluctuations soient visibles. Par exemple, une forte présence en 2020/2021 n'a pas entraîné une amélioration significative des notes. Cela peut indiquer que d'autres facteurs influencent la performance.

4. **Notes des Étudiants par Module** :

- Les moyennes varient significativement entre les modules. Par exemple :
    - "Business Intelligence" : Notes relativement bonnes (~13 en moyenne).
    - "Électronique Numérique" : Des performances légèrement inférieures (~12,5).

- Cela pourrait refléter la difficulté relative des modules ou des différences dans les méthodes d'évaluation.

1. Analyse de l'Engagement et des Absences des Étudiants – Tableau de Bord (Page 3)

Introduction

Cette page du tableau de bord met en lumière les relations entre les absences, le genre, les ajournements et l'engagement des étudiants dans différents modules. Ces données visent à identifier les tendances et facteurs pouvant influencer la réussite académique.

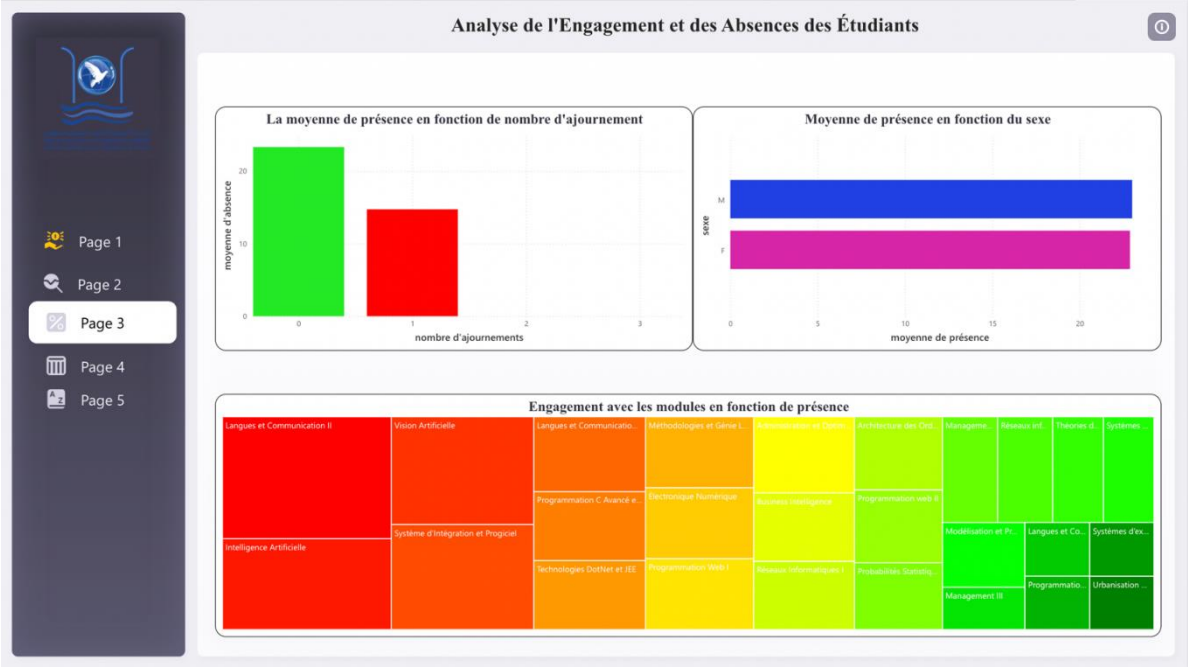

Figure 11 : troisième page du tableau de bord.

Description des Visualisations et Leur Utilité

1. **Moyenne de Présence en Fonction du Nombre d'Ajournements**
   - **Description** : Un histogramme présente la moyenne des absences selon le nombre d'ajournements des étudiants (0, 1 ou plus).
   - **Utilité** : Analyse de la corrélation entre l'assiduité et les ajournements. Un taux d'absences élevé est souvent associé à un risque accru d'échec académique.
2. **Moyenne de Présence en Fonction du Sexe**
   - **Description** : Un graphique à barres horizontales compare la moyenne de présence des étudiants masculins et féminins.

- **Analyse** : Identifie d'éventuelles disparités de genre en matière d'assiduité et explore les causes potentielles (motivation, obstacles spécifiques, etc.).

3. **Engagement avec les Modules en Fonction de la Présence (Treemap)**
   - **Description** : Une carte en mosaïque utilise des couleurs pour refléter les taux de présence des étudiants selon les modules. Les modules avec une faible présence sont en rouge, tandis que ceux avec une forte présence sont en vert.
   - **Utilité** : Met en évidence les modules qui nécessitent une attention particulière pour améliorer l'engagement.

Analyse des Résultats

1. **Corrélation entre Absences et Ajournements** :
   - Les étudiants sans ajournement ont une moyenne de présence plus élevée (colonne verte). Cependant, les absences augmentent de manière significative pour les étudiants ayant 1 ou plusieurs ajournements (colonne rouge).
   - Cela renforce l'importance de l'assiduité dans la réussite académique.
2. **Différences de Genre en Matière de Présence** :
   - Les moyennes de présence sont similaires entre les sexes, avec un léger avantage pour les femmes. Cela pourrait refléter une meilleure gestion du temps ou une plus grande implication académique.
   - Les différences étant minimes, d'autres facteurs externes pourraient avoir une plus grande influence.
3. **Engagement par Module** :
   - Les modules tels que "Langues et Communication II", "Vision Artificielle", et "Intelligence Artificielle" affichent des taux de présence faibles (rouge), suggérant des difficultés d'engagement. Ces modules pourraient bénéficier d'approches pédagogiques alternatives ou de ressources supplémentaires.
   - En revanche, des modules comme "Urbanisation SI", "Management III" et "Modélisation et Prédiction" montrent une forte présence (vert), témoignant d'un intérêt accru ou d'une meilleure adaptation des cours aux attentes des étudiants.

2. Analyse des Données - Tableau de Bord (Page 4)

Introduction

La quatrième page du tableau de bord propose une double analyse des données académiques des étudiants : une évaluation des performances moyennes par semestre et par promotion, ainsi qu'une étude détaillée de la présence par module. Ces visualisations complémentaires visent à fournir une vue d'ensemble des tendances et des comportements académiques sur plusieurs années.

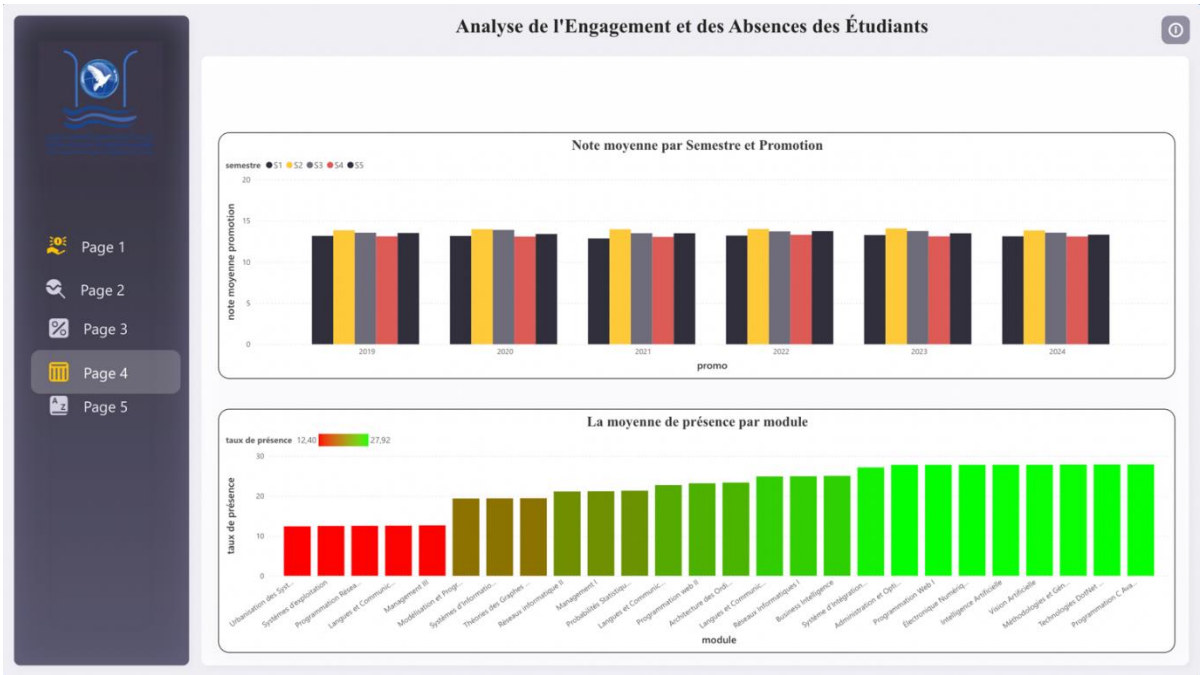

Figure 12 : quatrième page du tableau de bord.

Description des Visualisations et Leur Utilité

1. Note Moyenne par Semestre et Promotion

- **Description** : Un histogramme vertical illustre les notes moyennes obtenues par les promotions (2019 à 2024), segmentées par semestre (S1 à S5). Chaque barre est colorée pour différencier les semestres.
- **Utilité** : Identifier les fluctuations des performances académiques par promotion et repérer les semestres où les étudiants rencontrent le plus de difficultés. Cela peut aider à ajuster le contenu pédagogique ou les méthodes d'enseignement pour améliorer les résultats.

2. Taux de Présence par Module

- **Description** : Un graphique en barres horizontales affiche le taux de présence des étudiants pour chaque module. Les barres sont colorées en dégradé, du rouge (présence faible) au vert (présence élevée).
- **Analyse** : Met en évidence les modules avec une faible assiduité (rouge), qui pourraient nécessiter une investigation pour comprendre les causes (contenu, horaires, méthode d'enseignement). Les modules avec une forte présence (vert) reflètent l'intérêt ou l'efficacité pédagogique.

Analyse des Résultats

- **Performance Moyenne par Promotion** : Les notes moyennes restent relativement stables d'une promotion à l'autre, bien qu'une légère baisse soit perceptible pour certaines années. Cela indique la nécessité d'une analyse approfondie des facteurs spécifiques affectant chaque promotion.
- **Présence par Module** : Certains modules présentent des taux de présence alarmants (zones rouges), ce qui peut influencer négativement la performance des étudiants. À l'inverse, les modules avec une forte assiduité témoignent d'un intérêt accru ou d'une pédagogie engageante.

3. Analyse des Données - Tableau de Bord (Page 5)

Introduction

La cinquième page du tableau de bord propose une synthèse des performances académiques et de l'engagement des étudiants, combinant divers indicateurs clés pour une évaluation globale. Elle comprend des mesures comme le taux de réussite, la validation par module, la présence par semestre et année, ainsi que la moyenne générale. Ces visualisations permettent une compréhension approfondie des résultats et des écarts par rapport aux objectifs fixés.

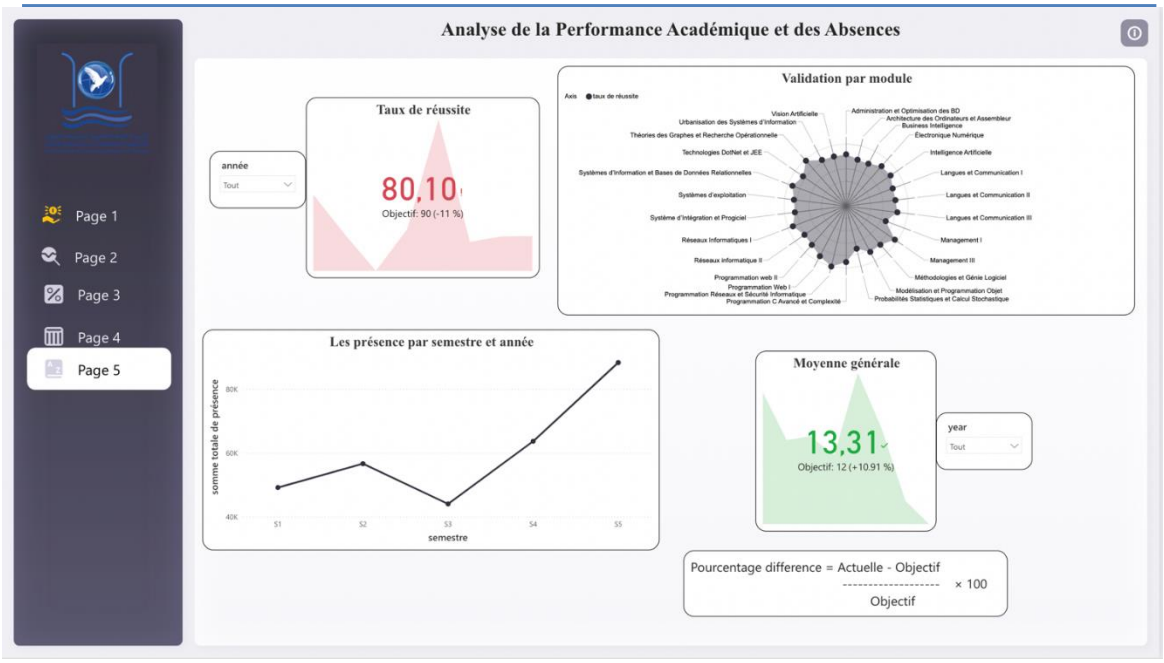

Figure 13 : dernière page du tableau de bord.

Description des Visualisations et Leur Utilité

1. Taux de Réussite

- **Description** : Un indicateur clé montre le taux global de réussite des étudiants (80,10 %) par rapport à un objectif fixé (90 %), avec la différence exprimée en pourcentage (-11 %).
- **Utilité** : Fournit une vue rapide des performances globales, aidant à identifier les écarts à combler pour atteindre les objectifs académiques.

2. Validation par Module

- **Description** : Un graphique en toile d'araignée affiche les taux de validation par module, mettant en avant les différences de performances académiques selon les matières enseignées.
- **Analyse** : Permet de repérer les modules où les étudiants rencontrent des difficultés (faible taux de validation) et ceux où ils excellent.

3. Présence par Semestre et Année

- **Description** : Un graphique linéaire illustre l'évolution de la somme totale de la présence des étudiants au fil des semestres (S1 à S5).
- **Utilité** : Analyse les variations d'assiduité au cours de l'année universitaire et permet de corréler cette donnée avec la réussite académique.

4. Moyenne Générale

- **Description** : Un indicateur montre la moyenne générale des étudiants (13,31) par rapport à un objectif fixé (12), avec la différence calculée (+10,91 %).
- **Analyse** : Fournit une mesure rapide des performances académiques moyennes et met en avant les progrès ou lacunes globales.

5. Formule de Calcul du Pourcentage de Différence

- **Description** : Une section explicative détaille la méthode de calcul utilisée pour évaluer les écarts entre les performances actuelles et les objectifs.
- **Utilité** : Assure la transparence dans l'interprétation des indicateurs.

Analyse des Résultats

- **Taux de Réussite** : Avec un taux de réussite actuel de 80,10 %, un effort est nécessaire pour atteindre l'objectif de 90 %. Cela peut impliquer des ajustements dans la pédagogie ou un soutien accru aux étudiants.
- **Validation par Module** : Les modules montrent une disparité notable dans les taux de validation. Certains modules techniques ou complexes (ex. : "Programmation Réseaux" ou "Systèmes d'Information") pourraient nécessiter des ressources pédagogiques supplémentaires.
- **Présence par Semestre** : La présence augmente significativement au fil des semestres, avec un pic à S5. Cela peut être dû à une implication accrue des étudiants à l'approche de la fin de leur cursus.
- **Moyenne Générale** : Avec une moyenne supérieure à l'objectif fixé (13,31 contre 12), les performances académiques globales sont satisfaisantes. Cependant, des disparités entre modules pourraient masquer des problématiques locales.

II. OLAP

L'OLAP (Online Analytical Processing) est une technologie essentielle dans le domaine de la Business Intelligence, permettant une analyse multidimensionnelle des données. Dans le cadre de ce projet, l'OLAP est utilisé pour explorer les performances académiques des étudiants à travers différentes dimensions telles que le temps, les modules, les professeurs et les étudiants eux-mêmes. Cette analyse multidimensionnelle permet de générer des insights précieux pour améliorer la prise de décision académique

1. Conception du Cube OLAP

Le cube OLAP a été conçu à l'aide de **Schema Workbench**, un outil permettant de modéliser et de configurer des cubes OLAP. Le cube est basé sur un **schéma en étoile**, avec la table de faits fact_student_performance au centre. Cette table contient des mesures clés telles que les notes moyennes des étudiants et le nombre total d'absences. Les dimensions suivantes sont utilisées pour contextualiser les données :

- **Dimension Étudiant (Dim_Student)** : Contient des informations sur les étudiants, telles que leur numéro Apogée, nom, prénom et genre.
- **Dimension Module (Dim_Module)** : Contient des informations sur les modules enseignés, tels que le code, le nom, la filière et le semestre.
- **Dimension Professeur (Dim_Professor)** : Contient des informations sur les professeurs, tels que leur nom et département.
- **Dimension Temps (Dim_Time)** : Permet d'analyser les données par année et par semestre.

1.1 Mesures et Dimensions

Le cube OLAP inclut les mesures suivantes :

- **Avg_note** : La note moyenne des étudiants par module.
- **Total_absences** : Le nombre total d'absences des étudiants.

Les dimensions permettent d'explorer ces mesures sous différents angles :

- **Par étudiant** : Analyse des performances individuelles.
- **Par module** : Évaluation des résultats par matière.
- **Par professeur** : Analyse de l'efficacité pédagogique.
- **Par temps** : Suivi des performances au fil des semestres et des années.

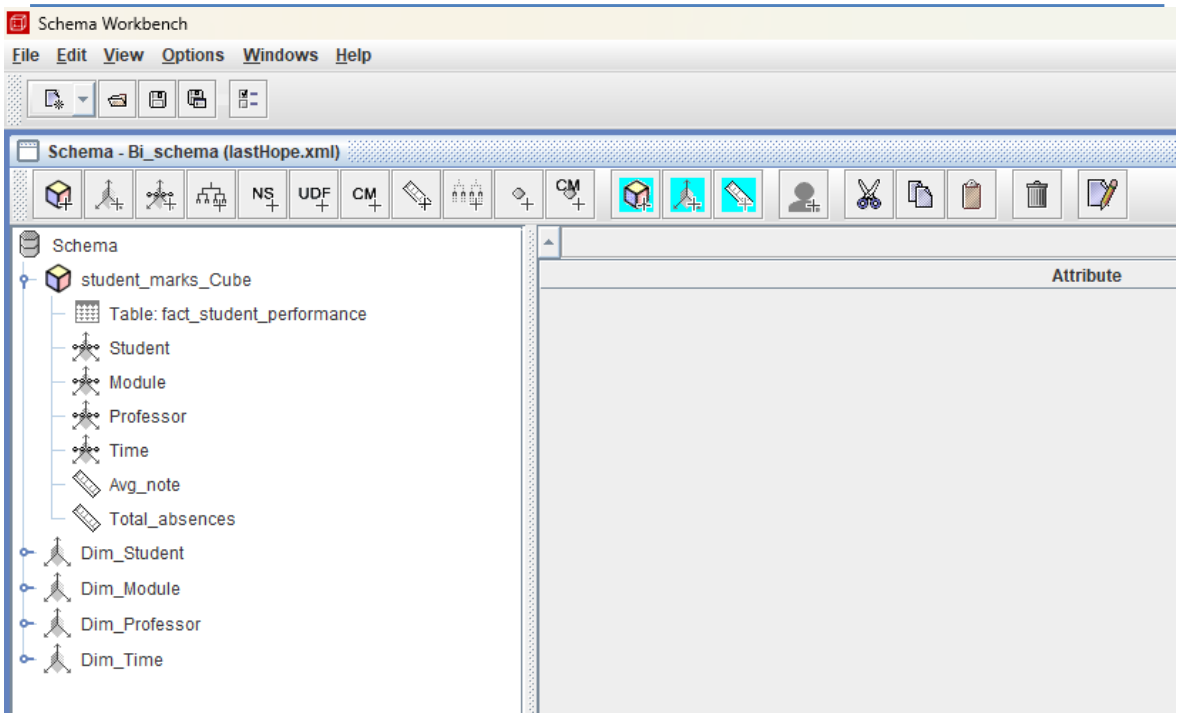

Figure 14 : Cube Olap

2. Requêtes OLAP et Analyse:

Voici quelques exemples de requêtes OLAP que vous pouvez exécuter pour analyser les données :

1. Obtenir la note moyenne globale :

Cette requête récupère la note moyenne de tous les étudiants pour tous les modules, sans segmentation.

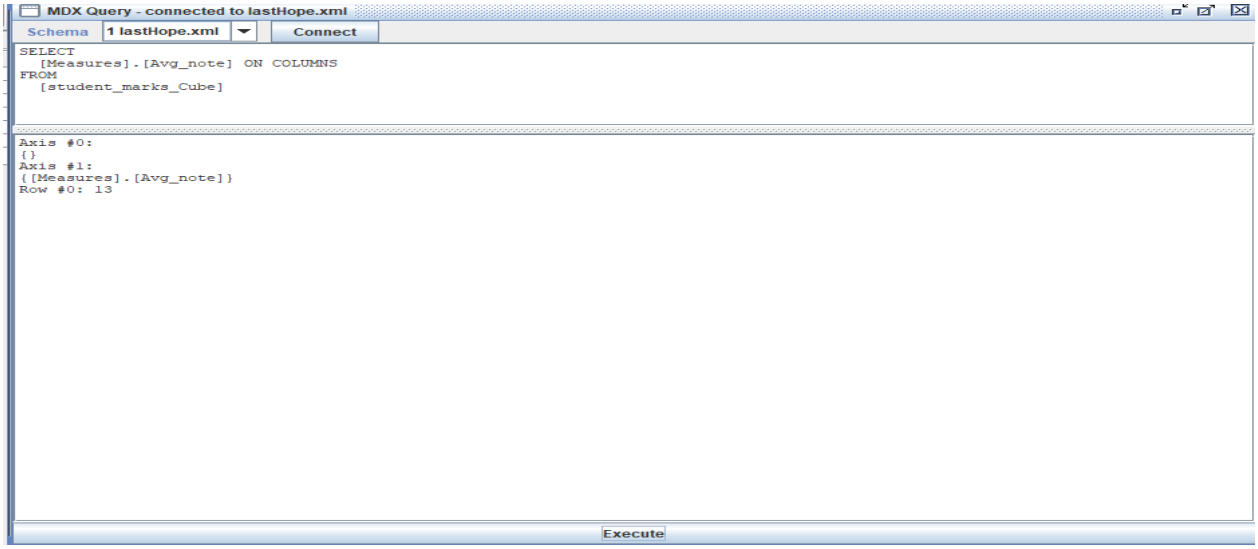

Figure 15:la note moyenne de tous les étudiants

2. Obtenir le nombre total d'absences :

Cette requête donne le total des absences de tous les étudiants pour tous les modules.

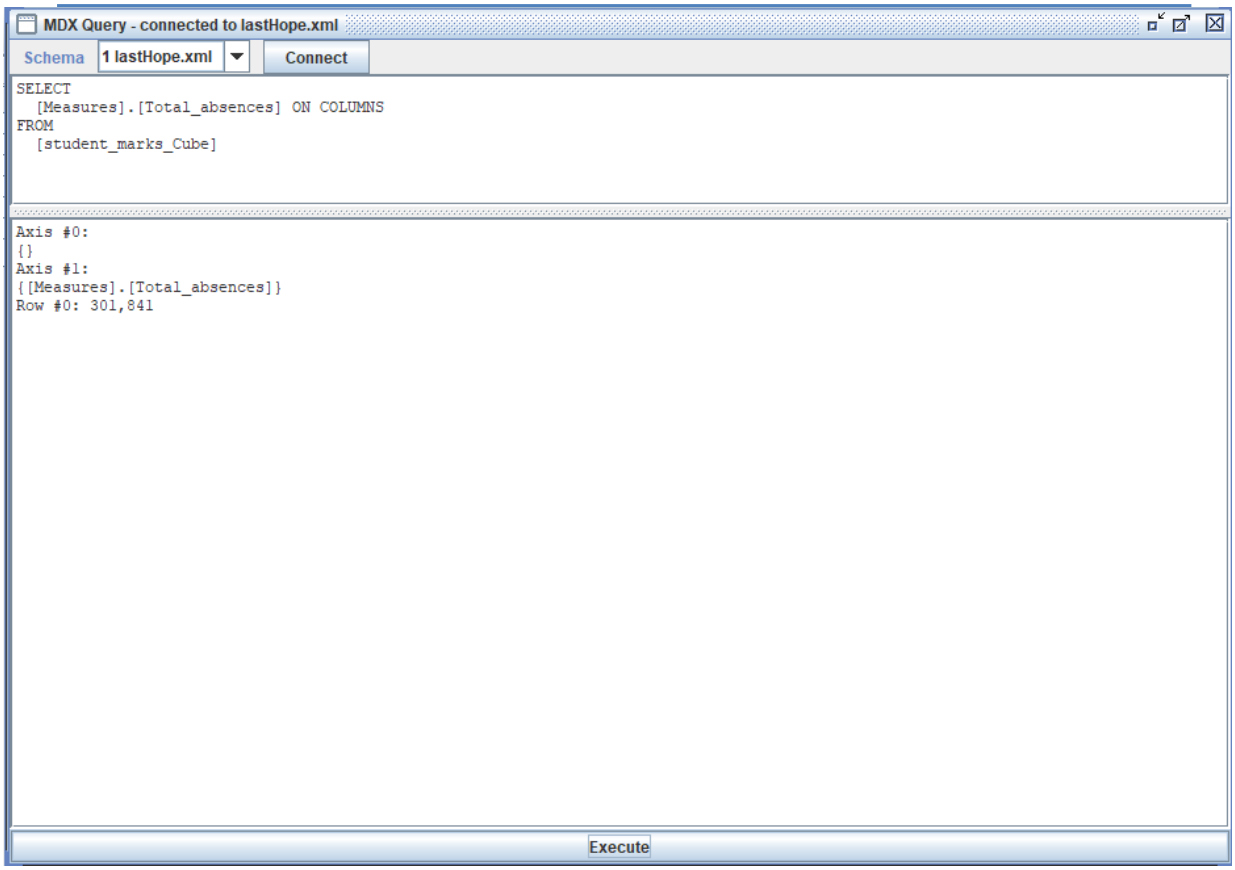

Figure 16:le total des présences de tous les étudiants

3. Obtenir la note moyenne par semestre :

Cette requête affiche la note moyenne pour chaque semestre.

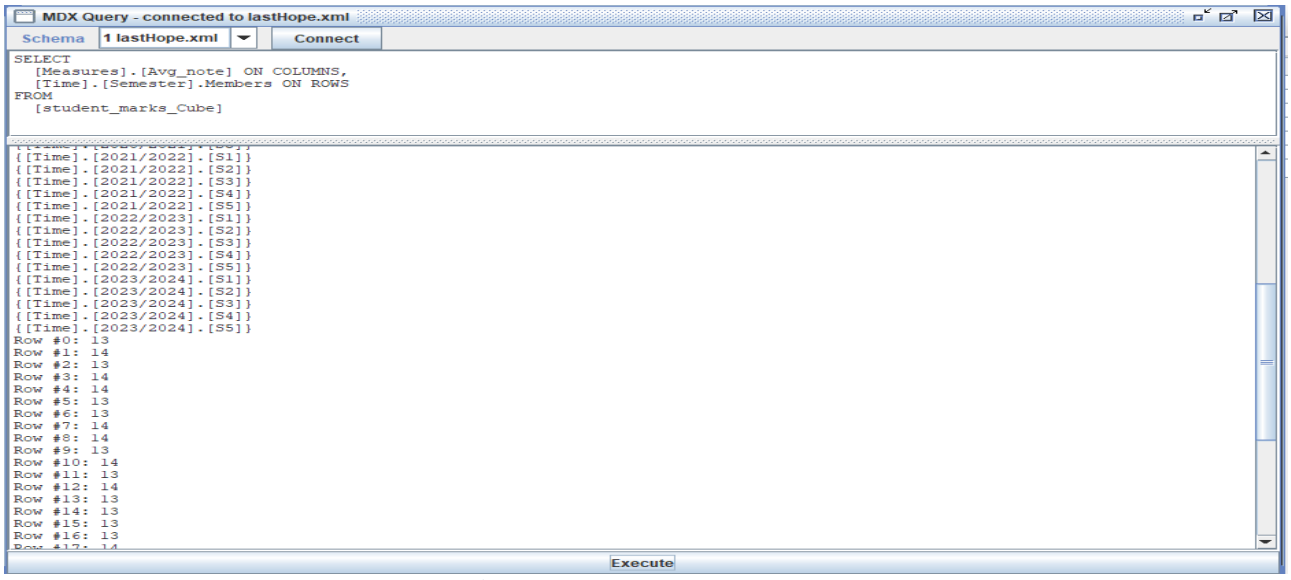

Figure 17:note moyenne par semestre

4. Obtenir le nombre total d'absences par étudiant :

Cette requête présente le nombre total d'absences pour chaque étudiant.

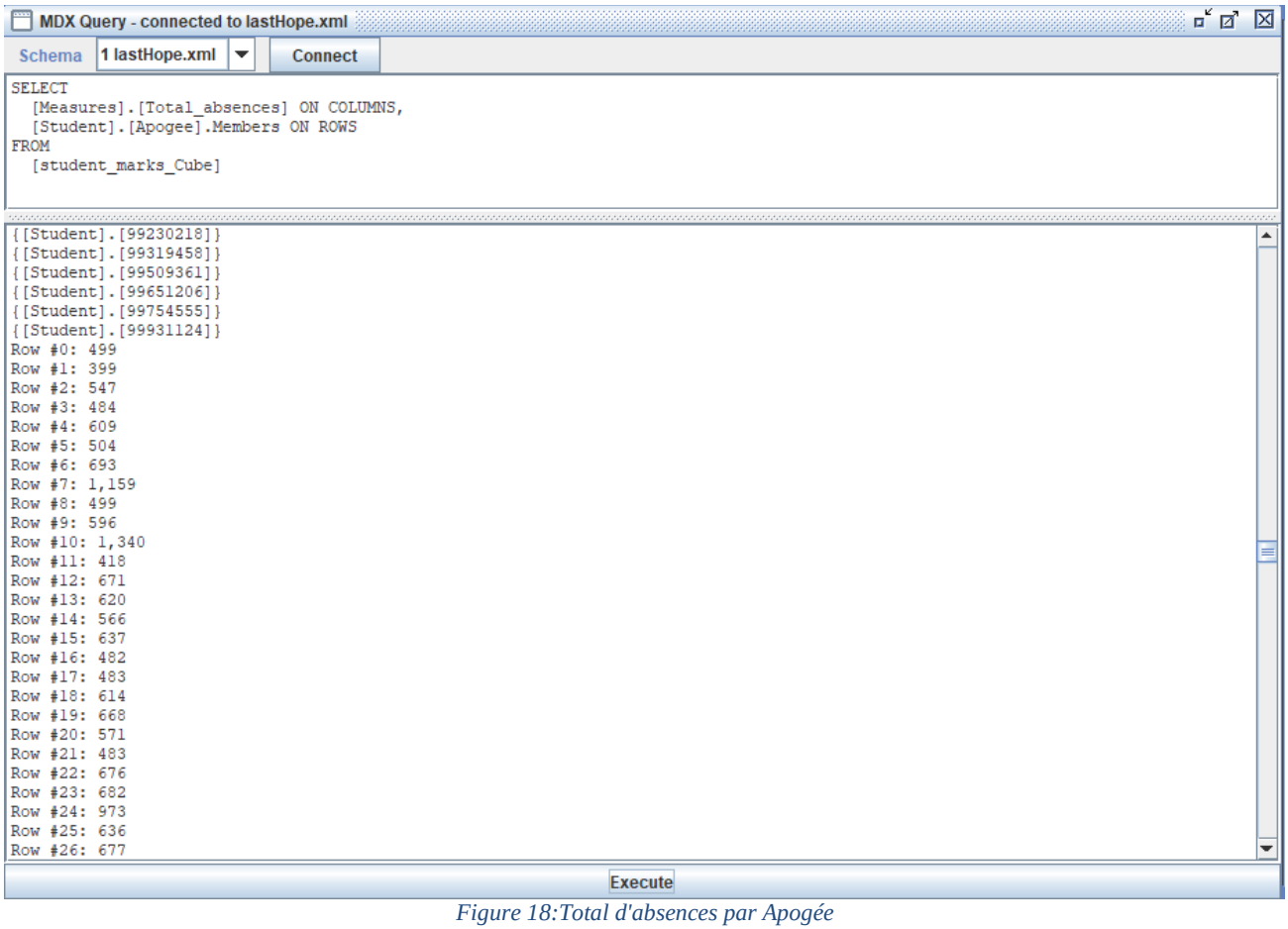

Figure 18:Total d'absences par Apogée

5. Obtenir la note moyenne par module :

Cette requête montre la note moyenne pour chaque module.

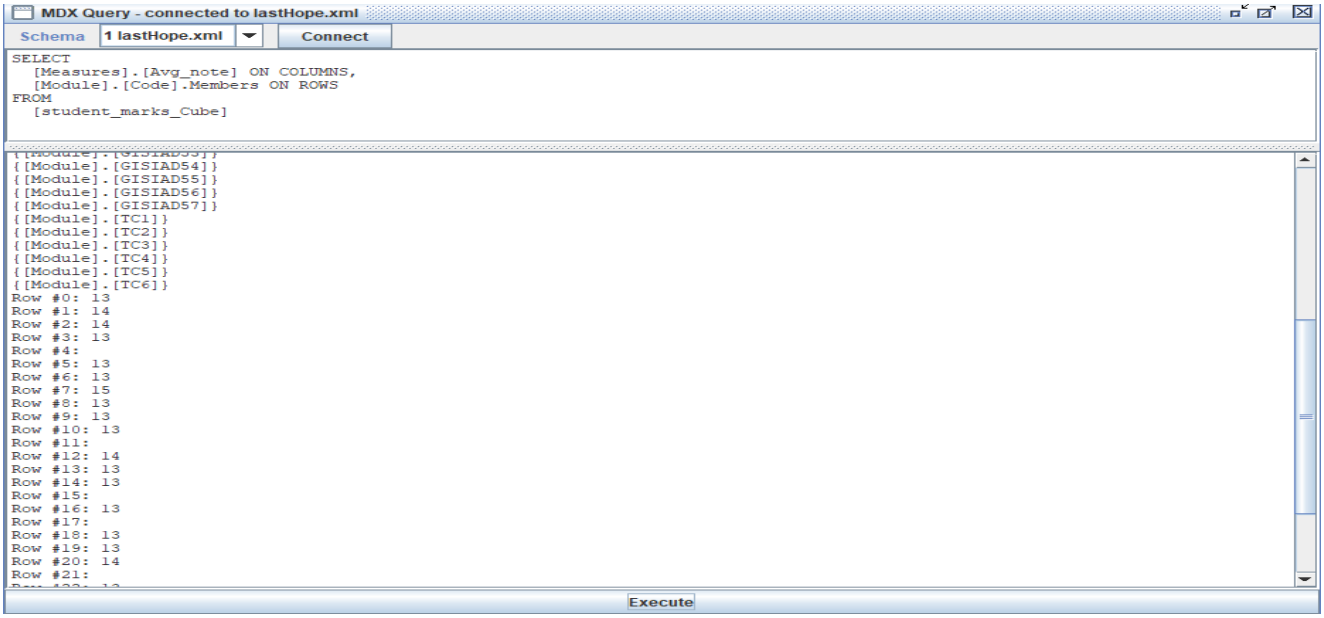

Figure 19: Avg Note par Module

Conclusion

Ce projet combine les capacités d'un tableau de bord interactif et l'intégration d'OLAP pour offrir un outil puissant d'analyse et de compréhension des données académiques. Le tableau de bord, avec ses visualisations variées, met en lumière les dynamiques clés telles que la répartition par genre, les performances académiques, et l'assiduité des étudiants. En parallèle, l'approche OLAP permet une exploration multidimensionnelle des données à travers des dimensions comme le temps, les modules, les professeurs, et les étudiants, tout en utilisant des mesures essentielles telles que les notes moyennes et le nombre total d'absences.

Les analyses révèlent à la fois des points forts, comme un équilibre de genre satisfaisant et des modules avec des taux de réussite élevés, et des axes d'amélioration, tels que l'augmentation de la présence et l’identification des causes des faibles performances ou absences. Ces insights offrent une base solide pour la prise de décisions pédagogiques et stratégiques.

En somme, la synergie entre le tableau de bord et OLAP enrichit la gestion des performances académiques en permettant un suivi régulier, une meilleure compréhension des tendances, et une adaptation des pratiques éducatives, tout en posant les bases pour des analyses futures dans le domaine de la Business Intelligence.

Conclusion générale

Ce projet nous a permis de concevoir et de mettre en œuvre un système décisionnel complet et intégré, couvrant plusieurs aspects essentiels pour l'analyse des données académiques. L'ensemble des étapes, de la conception de la base de données à l'exploitation des résultats à travers des outils de reporting et d'analyse OLAP, a été réalisé de manière cohérente afin d'offrir une vue d'ensemble claire et accessible des performances académiques des étudiants.

La première étape a consisté à structurer les données académiques en concevant une **base de données opérationnelle** efficace, accompagnée d'un modèle conceptuel permettant une gestion optimale des informations relatives aux étudiants, aux professeurs, aux modules, et aux résultats. Ces données ont ensuite été intégrées dans un **Data Warehouse** basé sur une architecture dimensionnelle en schéma en étoile, garantissant une organisation logique et adaptée aux besoins analytiques.

Une fois les données centralisées et structurées, le processus **ETL** a permis d'extraire, transformer et charger ces informations dans le système de stockage décisionnel, assurant ainsi leur cohérence et leur fiabilité pour les analyses à venir. Ce processus a été conçu pour garantir la qualité des données et faciliter leur utilisation pour l'analyse décisionnelle.

Enfin, des outils d'**analyse de données**, tels que des rapports détaillés et des analyses OLAP, ont été développés pour explorer les performances académiques sous différents angles, permettant une prise de décision éclairée. Ces outils fournissent des insights précieux sur les résultats des étudiants, les tendances et les domaines nécessitant des améliorations.

En conclusion, ce projet a permis de mettre en place un système complet de gestion et d'analyse des données académiques, facilitant ainsi la prise de décisions stratégiques et pédagogiques. La conception et l'intégration des différentes étapes ont contribué à une solution robuste, flexible et performante, prête à évoluer en fonction des besoins futurs.