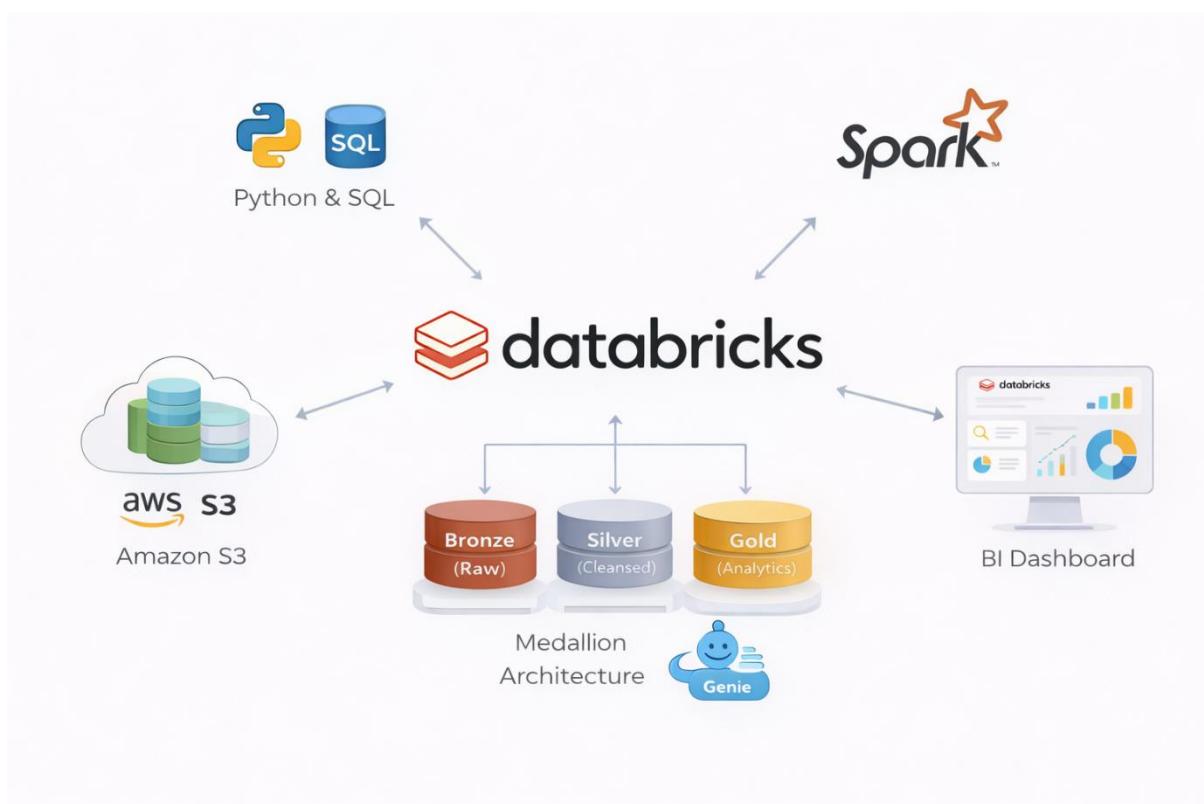


PIPELINE DU SUIVI DES VENTES

DU STREAMING DE DONNÉES À LA VISUALISATION DANS DATABRICKS



Francois Louis Marie NTONGA



francoislouismarie.contact@gmail.com

Data Engineer Junior

Objectif du projet

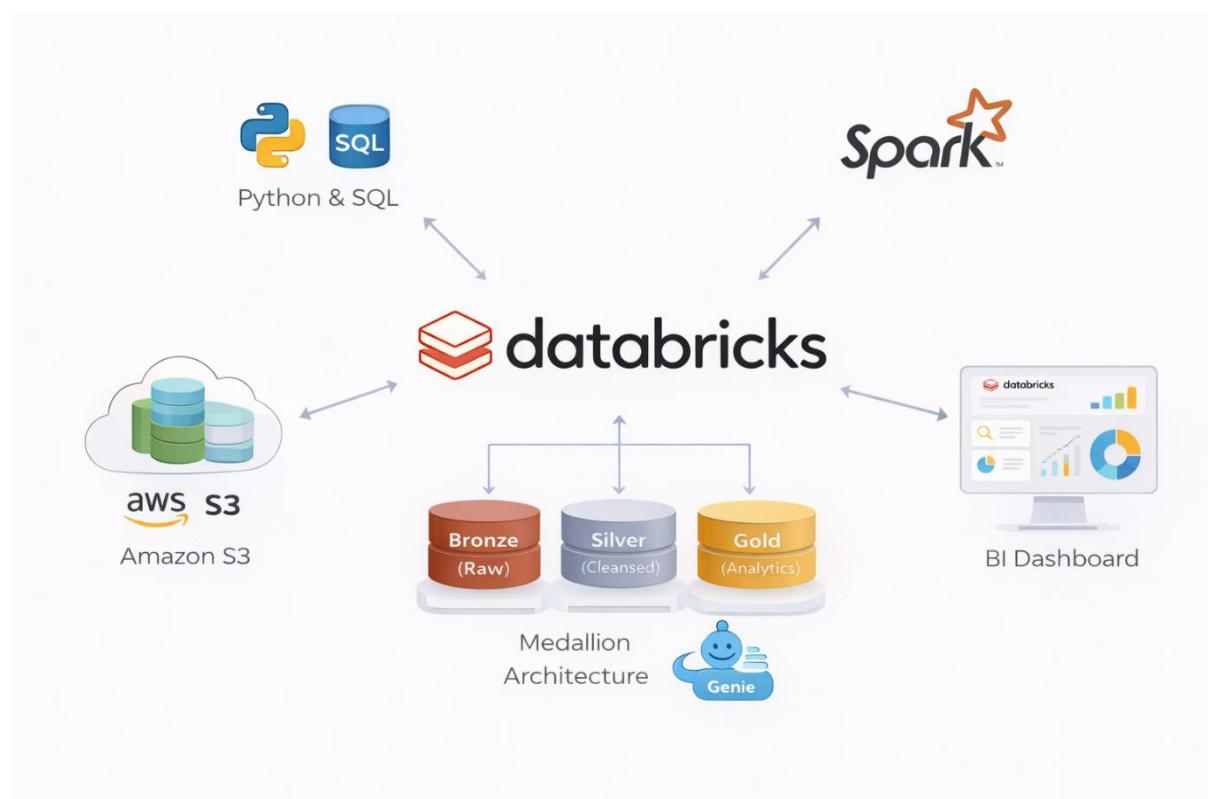
Concevoir et industrialiser un **pipeline** utilisant **Databricks**, dans le contexte d'une entreprise FMCG (Fast-Moving Consumer Goods). L'objectif est de construire un **pipeline ETL bout-à-bout** pour consolider les données de ventes de deux entreprises après une acquisition.

Créer une architecture **Lakehouse** permettant :

- D'ingérer des données provenant de deux sociétés,
- De les transformer selon le modèle **Médaillasson (Bronze → Silver → Gold)**,
- De produire des tables analytiques,
- Et de construire un tableau de bord final dans Databricks.

Architecture & Stack

- Databricks
- Python & SQL
- Apache Spark
- Amazon S3
- Medallion Architecture
- Genie
- BI Dashboard



Qui utilise ces données

- Équipes métiers ventes et finance
- Data Analysts et équipes BI
- Managers et décideurs opérationnels

Décisions métiers possibles

- Suivi et pilotage du chiffre d'affaires,
- Analyse des performances par client produit canal et période,
- Identification des produits et clients les plus rentables,
- Analyse des tendances et de la saisonnalité des ventes,
- Optimisation des stratégies commerciales et de distribution

Résultat clé

Un pipeline data automatisé scalable et orienté production fournissant des insights fiables accessibles via dashboards et IA conversationnelle et directement exploitables par les équipes métier.

Résumé

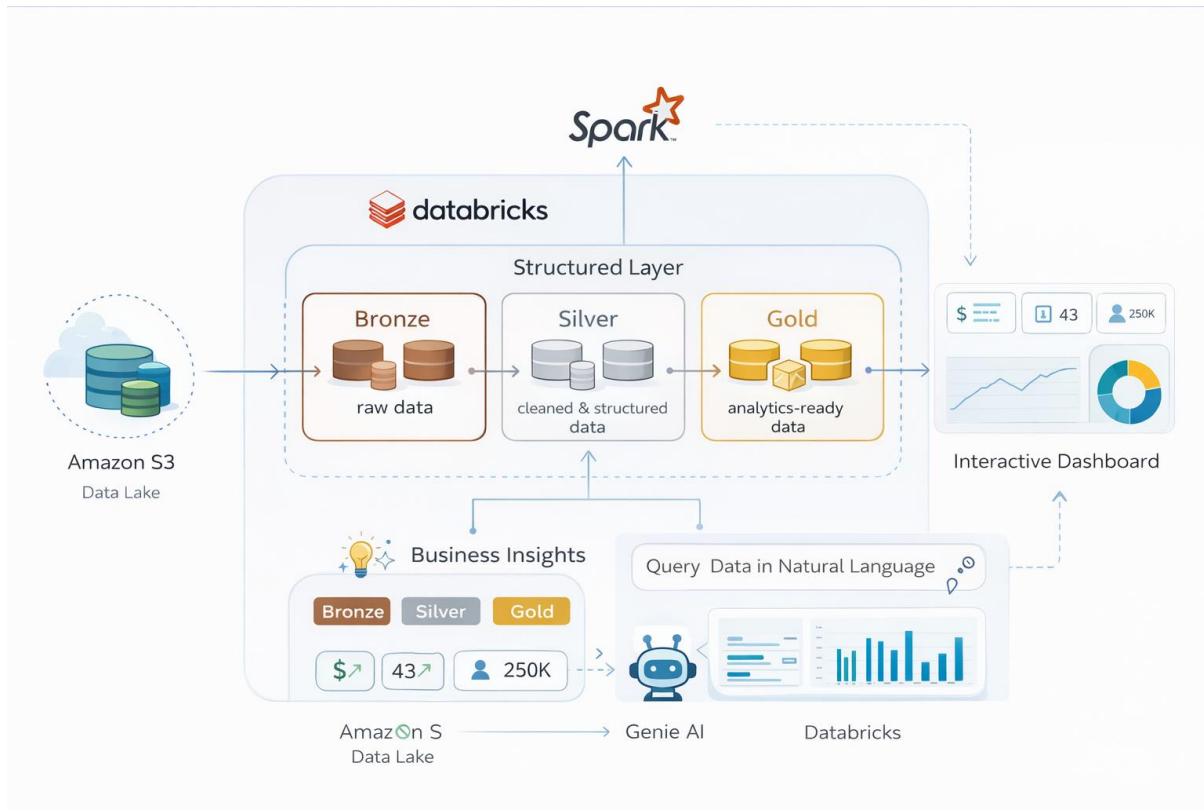
Ce projet vise la conception et la mise en œuvre d'une plateforme Data Engineering complète couvrant l'ensemble du cycle de vie de la donnée depuis l'ingestion des données brutes jusqu'à leur exploitation analytique et décisionnelle en s'appuyant sur des outils et architectures utilisés en environnement professionnel.

Les données sources ont été stockées dans Amazon S3 organisé comme un data lake puis ingérées dans Databricks à l'aide de Spark afin de garantir un traitement distribué et scalable. Une architecture Bronze - Silver - Gold a été mise en place pour structurer les données depuis les données brutes jusqu'aux données analytiques finales modélisées en schéma en étoile. Une vue analytique enrichie a été créée pour centraliser les informations métier et un pipeline Databricks Jobs a permis d'orchestrer les traitements avec gestion des dépendances exécution incrémentale supervision des runs et planification automatique rendant le pipeline fiable automatisé et industrialisable.

La consommation de la donnée est assurée via un Genie IA connecté à la couche Gold permettant l'exploration en langage naturel la génération automatique de requêtes SQL de visualisations et d'analyses business ainsi que par un Dashboard interactif Databricks intégrant KPI analyses temporelles classements et filtres dynamiques.

INTRODUCTION

À travers ce projet, la donnée est abordée comme un produit à part entière depuis son ingestion jusqu'à sa mise à disposition pour la prise de décision. Il a été conçu pour illustrer la chaîne complète de valorisation de la donnée depuis son ingestion jusqu'à sa consommation analytique en combinant des technologies cloud et Big Data largement utilisées en entreprise. L'objectif est de démontrer la capacité à structurer des pipelines fiables automatisés et exploitables tout en apportant une réelle valeur métier à travers l'analyse et la visualisation des données.



Étape 1 – Création et importation des données dans le bucket Amazon S3

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and various icons. Below it, the path is shown as 'Amazon S3 > Compartiments > sportbar-bp'. The main area is titled 'sportbar-bp' with a 'infos' link. A horizontal menu bar includes 'Objets', 'Métadonnées', 'Propriétés', 'Autorisations', 'Métriques', 'Gestion', and 'Points d'accès'. The 'Objets' tab is selected, showing a list of 4 items. Each item has a checkbox, a name, a type (Dossier), a last modified date (all '-'), a size (all '-'), and a storage class (all '-'). Action buttons include 'Copier l'URI S3', 'Copier l'URL', 'Télécharger', 'Ouvrir', 'Supprimer', 'Actions', 'Créer un dossier', and 'Charger'.

Nom	Type	Dernière modification	Taille	Classe de stockage
customers/	Dossier	-	-	-
gross_price/	Dossier	-	-	-
orders/	Dossier	-	-	-
products/	Dossier	-	-	-

Dans cette première étape, j'ai créé un **bucket Amazon S3** servant de **data lake** pour le projet. Les données sources (fichiers CSV) ont été organisées par domaine métier dans des dossiers distincts.

Cette organisation permet :

- Une meilleure lisibilité des données,
- Une ingestion plus simple dans Databricks,
- Une base solide pour une architecture **Bronze / Silver / Gold**.

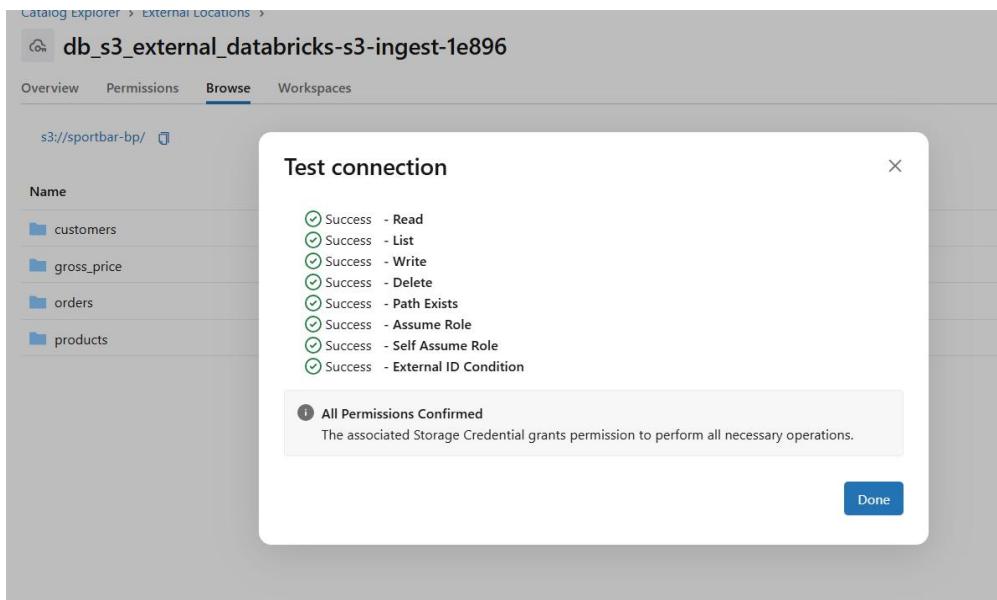
Étape 2 – Connexion et liaison entre Databricks et Amazon S3

The screenshot shows the AWS CloudFormation console with the stack 'databricks-s3-ingest-1e896'. The 'Events' tab is active, showing a list of 38 events. The first event is highlighted with a blue box and labeled 'CREATE_COMPLETE'. Other events show various actions like 'databricks-s3-ingest-1e896' and 'createExternalLocation'.

The screenshot shows the Databricks Catalog Explorer with the location 'db_s3_external_data'. A yellow arrow points to the location name. A red box highlights the URL 's3://sportbar-bp/'. The catalog also lists tables like 'customers', 'gros,price', 'orders', and 'products'.

Afin de permettre à Databricks d'accéder aux données stockées dans S3, une **liaison sécurisée** a été mise en place via une **External Location** et un rôle IAM dédié. La connexion permet à Databricks de lire et écrire directement dans le bucket S3 sans exposer de clés sensibles.

Étape 3 – Test de la connexion entre Databricks et S3



Étape 4 – Crédation du catalog et des schémas dans Databricks

A screenshot of the Databricks workspace interface. The left sidebar shows 'New', 'Home', 'Workspace' (selected), 'Recents', 'Catalog', 'Jobs & Pipelines', 'Compute', 'Marketplace', 'SQL', 'SQL Editor', 'Queries', 'Dashboards', 'Genie', 'Alerts', 'Query History', and 'SQL Warehouses'. The main area shows a notebook titled 'setup_catalog' with two cells. Cell 1 contains the SQL command: 'CREATE CATALOG IF NOT EXISTS fmcf;' followed by 'USE CATALOG fmcf;'. Cell 2 contains the SQL command: 'CREATE SCHEMA IF NOT EXISTS fmcf.gold;' followed by 'CREATE SCHEMA IF NOT EXISTS fmcf.silver;' and 'CREATE SCHEMA IF NOT EXISTS fmcf.bronze;'. Both cells have a status of 'OK' and a note: 'This result is stored as _sqldf and can be used in other Python and SQL cells.'

Dans Databricks, un **catalog** dédié au projet a été créé, suivi de la mise en place de trois **schémas**

Étape 5 – Lecture des données clients depuis S3 avec Spark

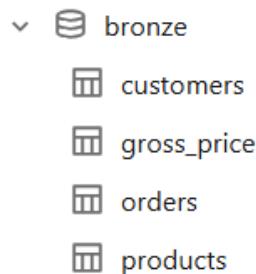
The screenshot shows a Databricks notebook interface. At the top, a code cell (cell 6) is displayed with the following Python code:

```
basepath = f's3://sportbar-bp/{data_source}/*csv'  
print(basepath)  
  
s3://sportbar-bp/customers/*csv
```

Below it, another code cell (cell 7) shows the result of running the code, which is a DataFrame named 'df'. The DataFrame has three columns: '_c0', '_c1', and '_c2'. The data is as follows:

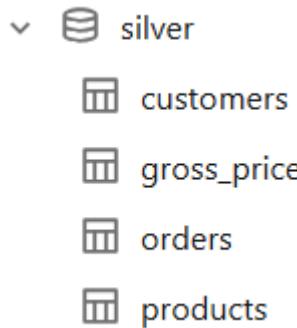
	$\text{\texttt{A}}_{\text{c}} \text{\texttt{.}} \text{\texttt{c0}}$	$\text{\texttt{A}}_{\text{c}} \text{\texttt{.}} \text{\texttt{c1}}$	$\text{\texttt{A}}_{\text{c}} \text{\texttt{.}} \text{\texttt{c2}}$
1	customer_id	customer_name	city
2	789201	FitFuel Market	Bengaluru
3	789202	FitFuel Market	Hyderabad
4	789203	FitFuel Market	New Delhi
5	789301	Athlete's Choice Store	Bengaluru
6	789303	Athlete's Choice Store	New Delhi

Étape 6 – Mise en place de la couche Bronze (données brutes)



La couche Bronze représente le niveau d'ingestion brute des données. À ce stade, les données issues de S3 sont stockées dans Databricks sans transformation métier majeure, en conservant leur structure d'origine.

Étape 7 – Création de la couche Silver (données nettoyées et structurées)



La couche **Silver** correspond au niveau de **nettoyage, normalisation et structuration** des données. Les données provenant de la couche Bronze y sont :

- Filtrées,
- Typées correctement,
- Enrichies si nécessaire
- Prêtes pour un usage analytique.

Étape 8 – Construction de la couche Gold (modèle analytique)



La couche **Gold** est dédiée à l'**analyse décisionnelle**. Elle repose sur une **modélisation en étoile (Star Schema)** composée de :

- Tables de dimensions (`dim_customers`, `dim_products`, `dim_date`, `dim_gross_price`)
- Table de faits (`fact_orders`)
- Cette modélisation permet :
- Des requêtes SQL performantes,
- Une lecture simple par les outils BI,
- Une séparation claire entre faits et dimensions.

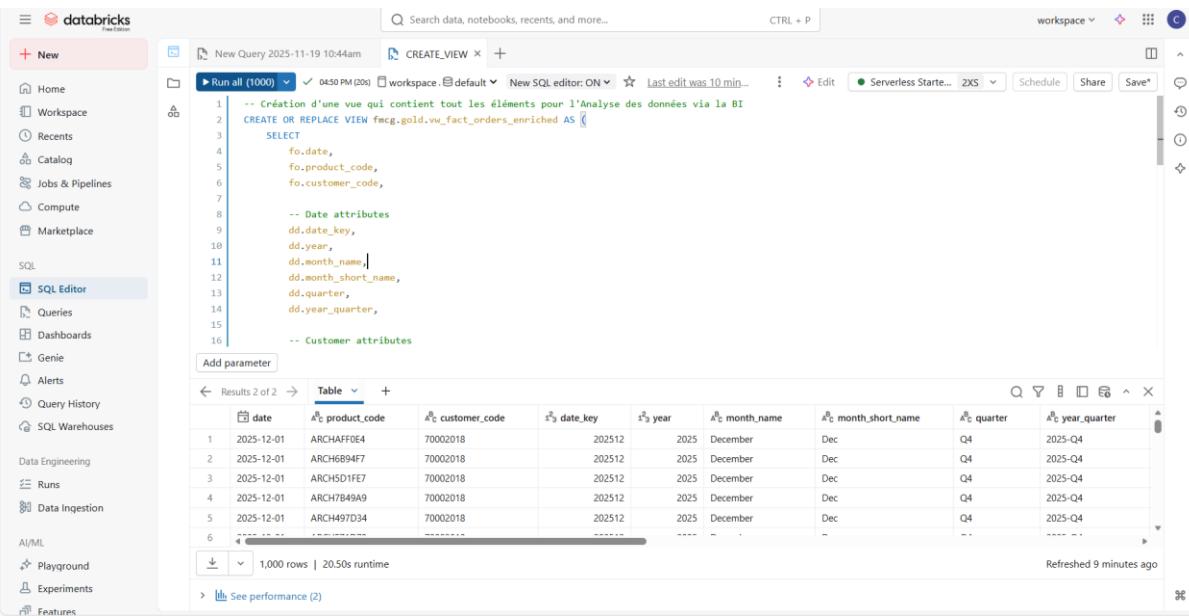
Nous avons une vue globale des Schémas Bronze, Silver et Gold :

L'organisation des schémas dans le **Catalog Databricks** permet de visualiser clairement :

- Le cheminement des données,
- La séparation des responsabilités,
- La progression logique du pipeline.

Name	Reason for suggestion	Type
<code>dim_customers</code> fmcg.gold	You view frequently	Table
<code>sb_fact_orders</code> fmcg.gold	You view frequently	Table
<code>orders</code> fmcg.bronze	You view frequently	Table
<code>customers</code> fmcg.silver	You view frequently	Table
<code>dim_date</code> fmcg.gold	You view frequently	Table
<code>bronze</code> fmcg	You view frequently	Schema
<code>gold</code> fmcg	You view frequently	Schema
<code>fact_orders</code> fmcg.gold	You viewed • 1 hour ago	Table
<code>customers</code> fmcg.bronze	You viewed • 2 hours ago	Table
<code>orders</code> fmcg.silver	You viewed • 18 hours ago	Table
<code>sb_dim_gross_price</code> fmcg.gold	You viewed • 23 hours ago	Table

Étape 9 – Création d'une vue analytique enrichie pour la BI



The screenshot shows the Databricks SQL Editor interface. On the left is the sidebar with various workspace options like Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, and SQL. The SQL Editor tab is selected. In the main area, there's a search bar at the top right. Below it, a query editor window titled 'CREATE_VIEW x' contains the following SQL code:

```
-- Création d'une vue qui contient tous les éléments pour l'analyse des données via la BI
CREATE OR REPLACE VIEW fmcg.gold.vw_fact_orders_enriched AS
SELECT
    fo.date,
    fo.product_code,
    fo.customer_code,
    --
    -- Date attributes
    dd.date_key,
    dd.year,
    dd.month_name,
    dd.month_short_name,
    dd.quarter,
    dd.year_quarter,
    --
    -- Customer attributes
```

Below the code, there's a table preview titled 'Table' showing results from 1 to 6. The columns are: date, product_code, customer_code, date_key, year, month_name, month_short_name, quarter, and year_quarter. The data is as follows:

	date	product_code	customer_code	date_key	year	month_name	month_short_name	quarter	year_quarter
1	2025-12-01	ARCHAFF0E4	70002018	202512	2025	December	Dec	Q4	2025-Q4
2	2025-12-01	ARCH68B4F7	70002018	202512	2025	December	Dec	Q4	2025-Q4
3	2025-12-01	ARCH5D1FE7	70002018	202512	2025	December	Dec	Q4	2025-Q4
4	2025-12-01	ARCH7B49A9	70002018	202512	2025	December	Dec	Q4	2025-Q4
5	2025-12-01	ARCH497D34	70002018	202512	2025	December	Dec	Q4	2025-Q4
6									

At the bottom of the table preview, it says '1,000 rows | 205ms runtime'. To the right, it says 'Refreshed 9 minutes ago'.

Une **vue SQL enrichie** (`vw_fact_orders_enriched`) a été créée afin de centraliser **toutes les informations nécessaires à l'analyse BI** dans un seul objet.

Cette vue combine :

- Les faits de commandes,
- Les dimensions temporelles,
- Les informations clients et produits.

Elle permet aux analystes ou outils BI :

- D'éviter des jointures complexes,
- D'accéder rapidement aux indicateurs clés,
- De consommer la donnée de manière simple et performante.

Étape 10 – Crédation du pipeline d'ingestion et de transformation des données

The screenshot shows the Databricks interface for creating a pipeline. On the left, the sidebar is open with the 'Jobs & Pipelines' section selected. The main area displays a pipeline graph with four tasks: 'dim_processing_customer', 'dim_processing_products', 'dim_processing_prices', and 'fact_processing_orders'. These tasks are connected by arrows indicating their sequence. Below the graph, a message says 'No task selected' and 'Choose a task from the graph to edit its properties'. To the right of the graph, there are sections for 'Job details', 'Schedules & Triggers', and 'Job parameters'. The 'Job details' section includes fields for Job ID, Creator, Run as, Description, Lineage, and Performance optimization. The 'Schedules & Triggers' section shows 'None' and a 'Add trigger' button. The 'Job parameters' section indicates 'No job parameters are defined for this job' and a 'Edit parameters' button.

Un **pipeline Databricks Jobs** a été créé afin d'orchestrer l'ensemble des traitements de données du projet. Ce pipeline permet d'enchaîner automatiquement les différentes étapes de transformation à savoir l'ingestion, la transformation (Bronze, Silver, Gold), la structuration, le stockage et la mise à disposition des données pour l'Analyse BI.

Définition des tâches et des dépendances

The screenshot shows the Databricks interface for monitoring a pipeline run. The sidebar is open with the 'Jobs & Pipelines' section selected. The main area displays the 'Runs' tab for a pipeline named 'FMCG INCREMENTAL UPDATE'. A chart titled 'Runs' shows the total duration of each run, with the most recent run taking 30 seconds. Below the chart, a list of tasks is shown: 'dim_processing_customer', 'dim_processing_products', 'dim_processing_prices', and 'fact_processing_orders'. A table at the bottom provides detailed information for the latest run, including Start time (Feb 06, 2026, 02:...), Run ID (3067269902...), Launched (Manually), Duration (34s), Status (Running), Error code (None), and Run parameters. A 'Cancel runs' button is also visible.

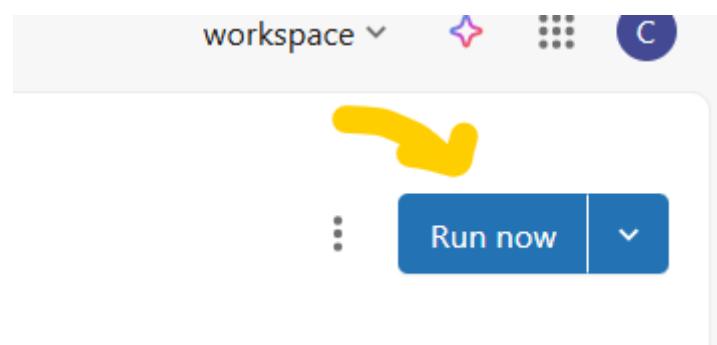
Les tâches ont été organisées sous forme de **graphe de dépendances** :

- Les dimensions sont traitées en amont,
- La table de faits est calculée uniquement une fois les dimensions prêtes.

Cette organisation garantit :

- La cohérence des données,
- L'intégrité référentielle,
- Une exécution fiable du pipeline.

Étape 11 – Lancement manuel du pipeline pour test



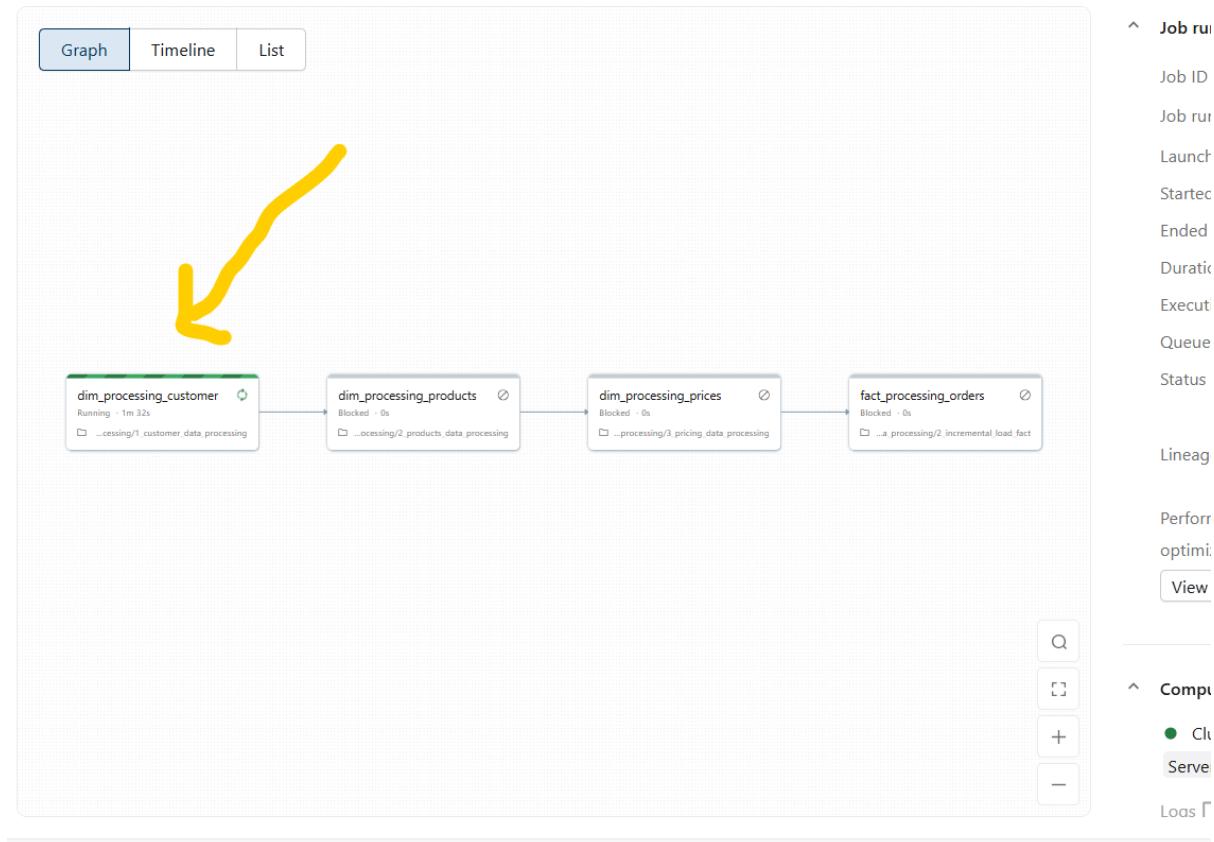
Le pipeline peut être déclenché manuellement via le bouton **Run now**, ce qui permet :

- De tester les traitements,
- De valider les dépendances,
- De vérifier les performances d'exécution.

Étape 12 – Exécution du pipeline et suivi en temps réel

Jobs & Pipelines > FMCG INCREMENTAL UPDATE >

FMCG INCREMENTAL UPDATE run Send feedback



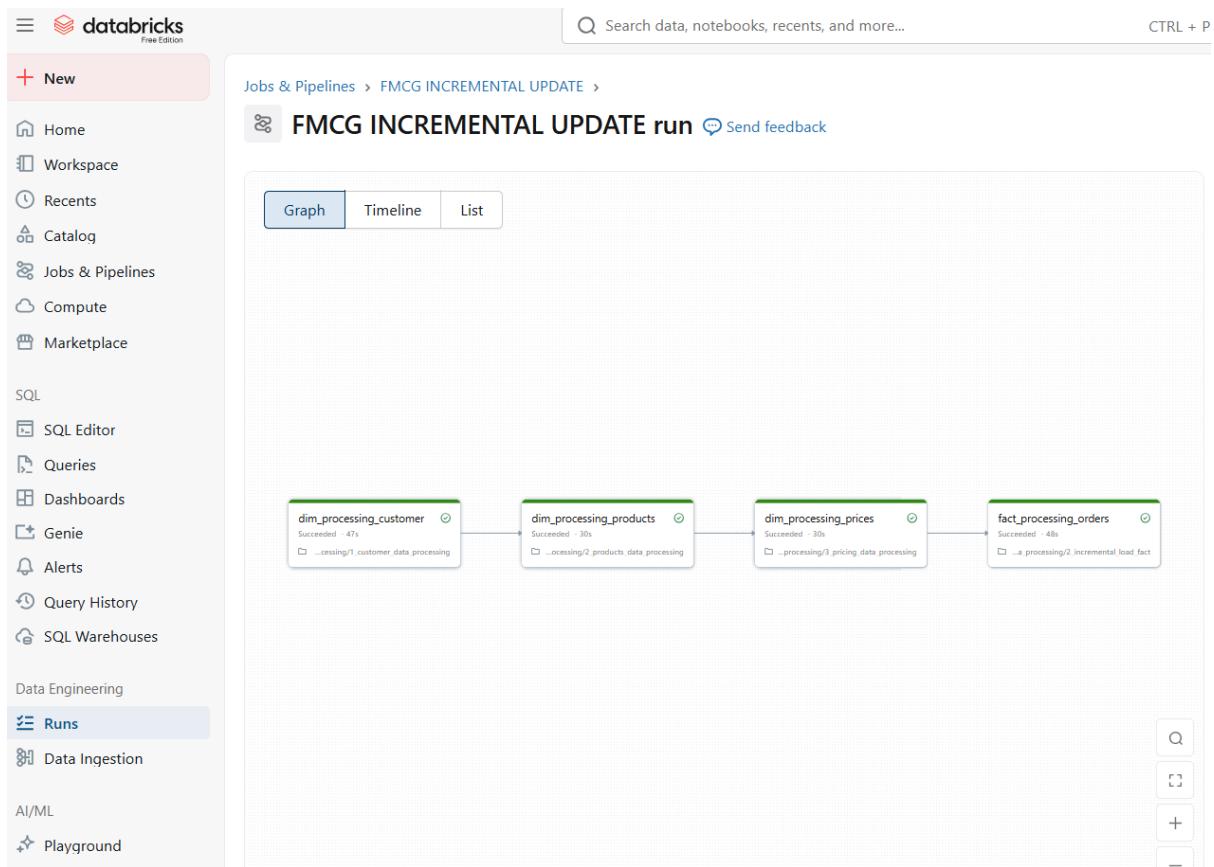
Pendant l'exécution, Databricks permet de suivre :

- L'état de chaque tâche,
- Les durées d'exécution,
- Les éventuels blocages ou erreurs.

Ce suivi visuel facilite :

- Le debugging,
- L'optimisation des performances,
- L'analyse du comportement du pipeline.

Pipeline exécuté avec succès (statut OK)



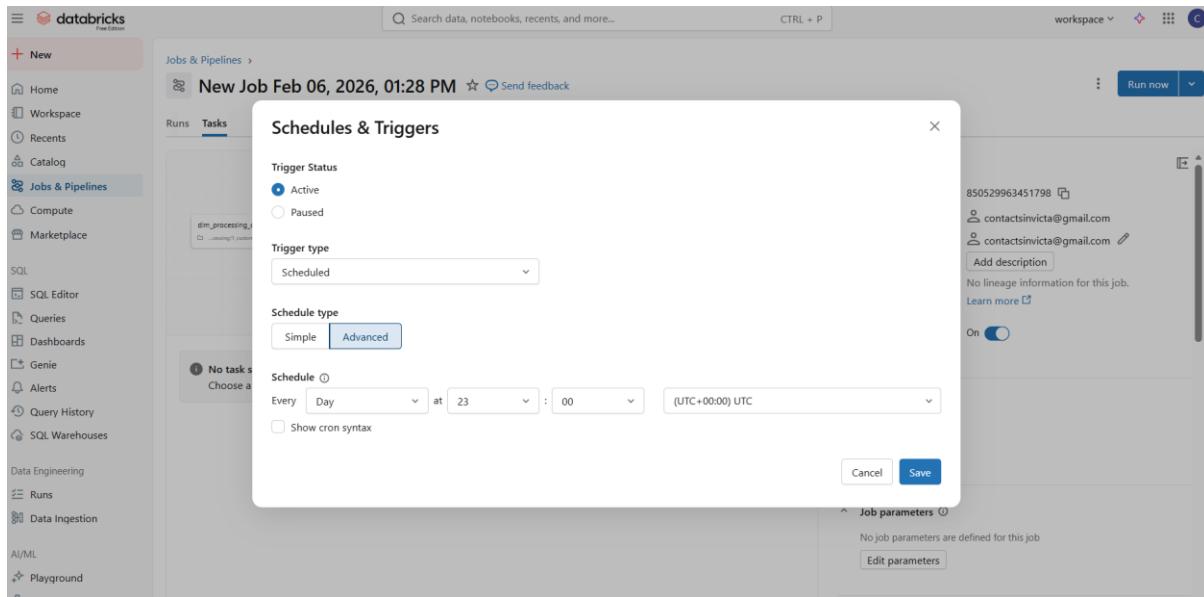
Une fois l'exécution terminée, l'ensemble des tâches apparaît en **vert**, indiquant un succès complet du pipeline.

Cela confirme :

- La bonne orchestration des traitements,
- La cohérence des dépendances,
- La stabilité du pipeline.

Cet état valide la capacité du projet à fonctionner de manière fiable de bout en bout.

Étape 13 – Planification automatique du pipeline (Triggers)



Un **trigger de planification** a été configuré afin de définir quand le pipeline doit s'exécuter automatiquement (exécution quotidienne à heure fixe).

Cette automatisation permet :

- Une ingestion régulière des nouvelles données,
- Un pipeline sans intervention manuelle,
- Une meilleure fiabilité opérationnelle.

Cette étape transforme le projet en **pipeline data autonome**.

Étape 14 – Crédit d'un espace Genie IA connecté à la vue analytique

The screenshot shows the Databricks Genie interface. On the left, a sidebar lists various navigation options: Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL Editor, Queries, Dashboards, Genie (which is selected), Alerts, Query History, and SQL Warehouses. Below these are sections for Data Engineering (Runs, Data Ingestion) and AI/ML (Playground, Experiments). The main area is titled "New Space" and contains a search bar and a "Data" tab. Under the "Data" tab, there's a section for "Review one suggested query" with a "Review" button. A table lists a single table named "vw_fact_orders_enriched" from the "fmcg.gold" database. The interface also includes sections for "Instructions" and "Settings".

Un espace Databricks Genie (Agent IA Databricks) a été créé et connecté directement à la vue qu'on a créée précédemment. L'objectif est de permettre une **interrogation directe des données en langage naturel**, sans écrire de SQL. Le Genie IA est configuré pour consommer **uniquement les données de la vue**, garantissant :

- Des données fiables,
- Des résultats cohérents,
- Une expérience proche d'un usage BI en production.

Interrogation des données en langage naturel

New Space ★

Show me total revenues by quarter

Analysis (click to view) vw_fact_orders_enriched

Total revenues by quarter are provided for each period. The highest revenue was in 2025-Q4 with ₹40,166,491,590, while the lowest was in 2024-Q2 with ₹3,714,797,270; revenues generally increased over time.

Result table (8 rows)

	year_quarter	total_revenue_inr
1	2024-Q1	3,943,669,280
2	2024-Q2	3,714,797,270
3	2024-Q3	7,676,783,970
4	2024-Q4	19,735,740,810
5	2025-Q1	13,471,970,533
6	2025-Q2	13,406,465,654
7	2025-Q3	17,816,862,991
8	2025-Q4	40,166,491,590

Total Revenue by Quarter

data bricks Free Edition

+ New

- Home
- Workspace
- Recents
- Catalog
- Jobs & Pipelines
- Compute
- Marketplace

SQL

- SQL Editor
- Queries
- Dashboards

Genie

- Alerts
- Query History
- SQL Warehouses

Data Engineering

- Runs
- Data Ingestion

AI/ML

- Playground
- Experiments

Show me total revenues by quarter

Analysis (click to view) vw_fact_orders_enriched

Total revenues by quarter are provided for each period. The highest revenue was in 2025-Q4 with ₹40,166,491,590, while the lowest was in 2024-Q2 with ₹3,714,797,270; revenues generally increased over time.

Result table (8 rows)

Total Revenue by Quarter

Quarter	total_revenue_inr
2024-Q1	3,943,669,280
2024-Q2	3,714,797,270
2024-Q3	7,676,783,970
2024-Q4	19,735,740,810
2025-Q1	13,471,970,533
2025-Q2	13,406,465,654
2025-Q3	17,816,862,991
2025-Q4	40,166,491,590

What are the total revenues by year?

What are the total revenues by quarter for the year 2024?

What are the total rev →

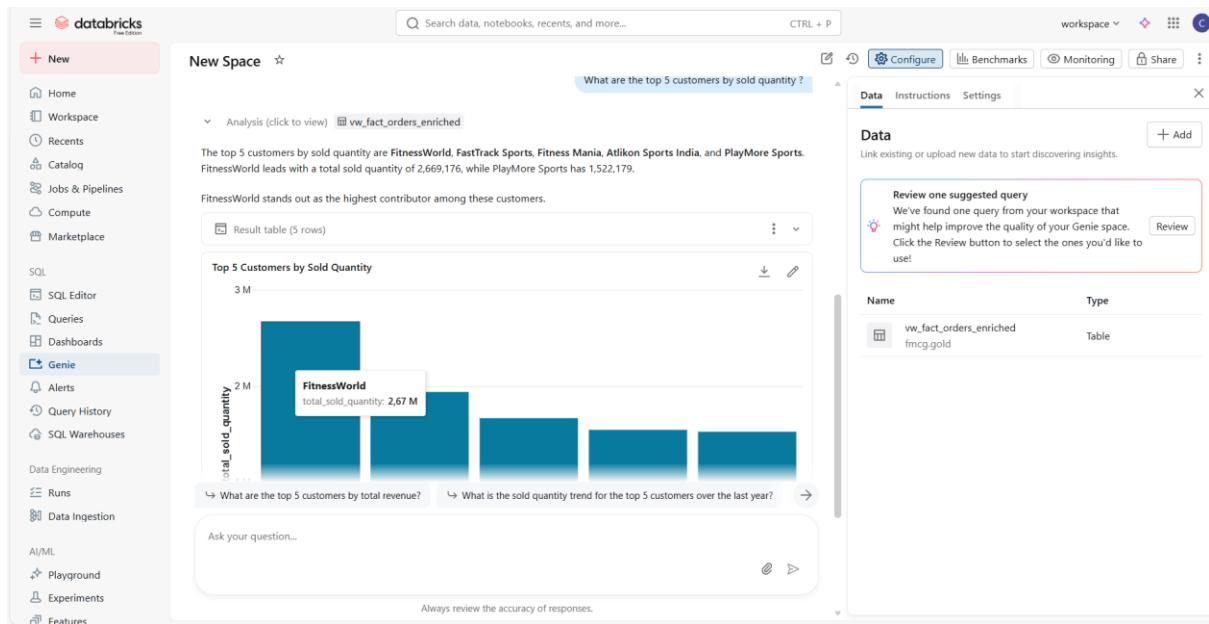
Ask your question...

Always review the accuracy of responses.

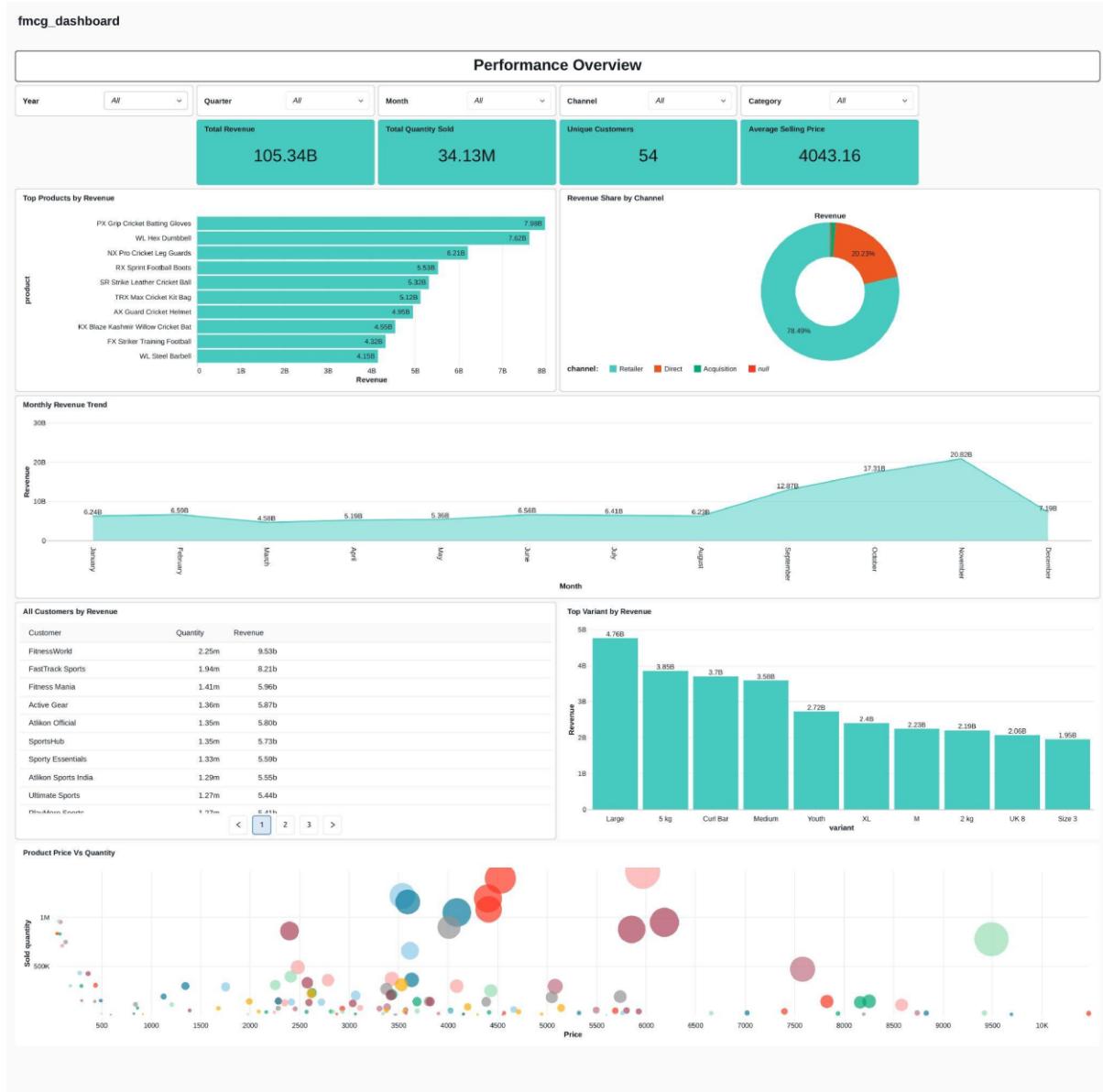
En complément des tableaux, le Genie IA génère automatiquement des **visualisations graphiques** :

- Histogrammes,
- Bar charts,
- Évolutions temporelles.

Cela permet une **lecture rapide des tendances** sans passer par un outil BI externe.



Étape 15 (Final) – Création du dashboard analytique en temps réel dans DataBricks (Sales Insights)



Un **Dashboard interactif** a été créé directement dans Databricks afin de restituer les données issues de la **couche Gold** et de la vue analytique [vw_fact_orders_enriched](#).

Ce Dashboard constitue la **couche de restitution finale** du pipeline data.

Il permet aux utilisateurs métiers d'accéder rapidement aux indicateurs clés sans connaissance technique.

Quels sont les KPI que nous pouvons en tirer de ce Dashboard ?

1- KPI de performance globale

Chiffre d'affaires total (Total Revenue)

- 105.34 B

Volume vendu total (Total Quantity Sold)

- 34.13 M unités

Nombre de clients uniques

- 54

Prix moyen de vente (Average Selling Price)

- 4043.16

2- KPI de mix produit (stratégie produit)

Top produits par chiffre d'affaires

- Classement des produits générateurs de valeur
- Sert à :
 - Focus marketing
 - Gestion des stocks

KPI dérivés :

- % du CA réalisé par le Top 10 produits
- Dépendance produit (risque)

Chiffre d'affaires par variant

- Ex : Large a fait X chiffre d'affaires, 5kg a fait Y Chiffre d'affaires, Curl Bar a fait Z chiffre d'affaires, etc.
- Permet :
 - D'optimiser les formats
 - D'identifier les tailles rentables
 - D'éliminer les variants faibles

3- KPI canal de vente :

Part du chiffre d'affaires par canal

- Retailer = 78 %
- Direct = 20 %

4- KPI temporels :

- Identification claire :
 - Saisonnalité
 - Pics (Q4 très fort)
 - Creux d'activité

KPI clés :

- Croissance MoM (Mois par Mois)
- Variabilité mensuelle
- Détection d'anomalies

5- KPI client (valeur & concentration)

Top clients par chiffre d'affaires

- FitnessWorld
- FastTrack Sports
- Fitness Mania, etc.

Ce que j'ai appris avec ce projet

- Construire un pipeline ETL complet dans Databricks
- Gérer des données dimensionnelles et factuelles
- Implémenter des chargements historiques et incrémentaux
- Utiliser S3 comme source de données
- Orchestrer un workflow de data engineering
- Créer une table analytique et un Dashboard

CONCLUSION

Ce projet illustre une approche complète et structurée du Data Engineering moderne en mettant en œuvre une chaîne de traitement des données de bout en bout depuis l'ingestion jusqu'à la restitution analytique et décisionnelle. Il démontre la capacité à concevoir des pipelines automatisés et évolutifs.

À travers l'utilisation de technologies cloud et Big Data telles qu'Amazon S3, Spark et Databricks, le projet met en évidence une compréhension concrète des enjeux de performance de scalabilité et d'industrialisation. L'intégration de traitements incrémentaux de l'orchestration et de la planification rapproche l'ensemble d'un contexte réel de production.

Enfin la mise à disposition des données via des dashboards interactifs et une exploration assistée par IA souligne la volonté de replacer la donnée au cœur de la prise de décision. Ce projet constitue ainsi une base solide pour un environnement Data professionnel et reflète les compétences attendues d'un profil Data Engineer junior ou en stage capable de transformer des données brutes en valeur métier exploitable.