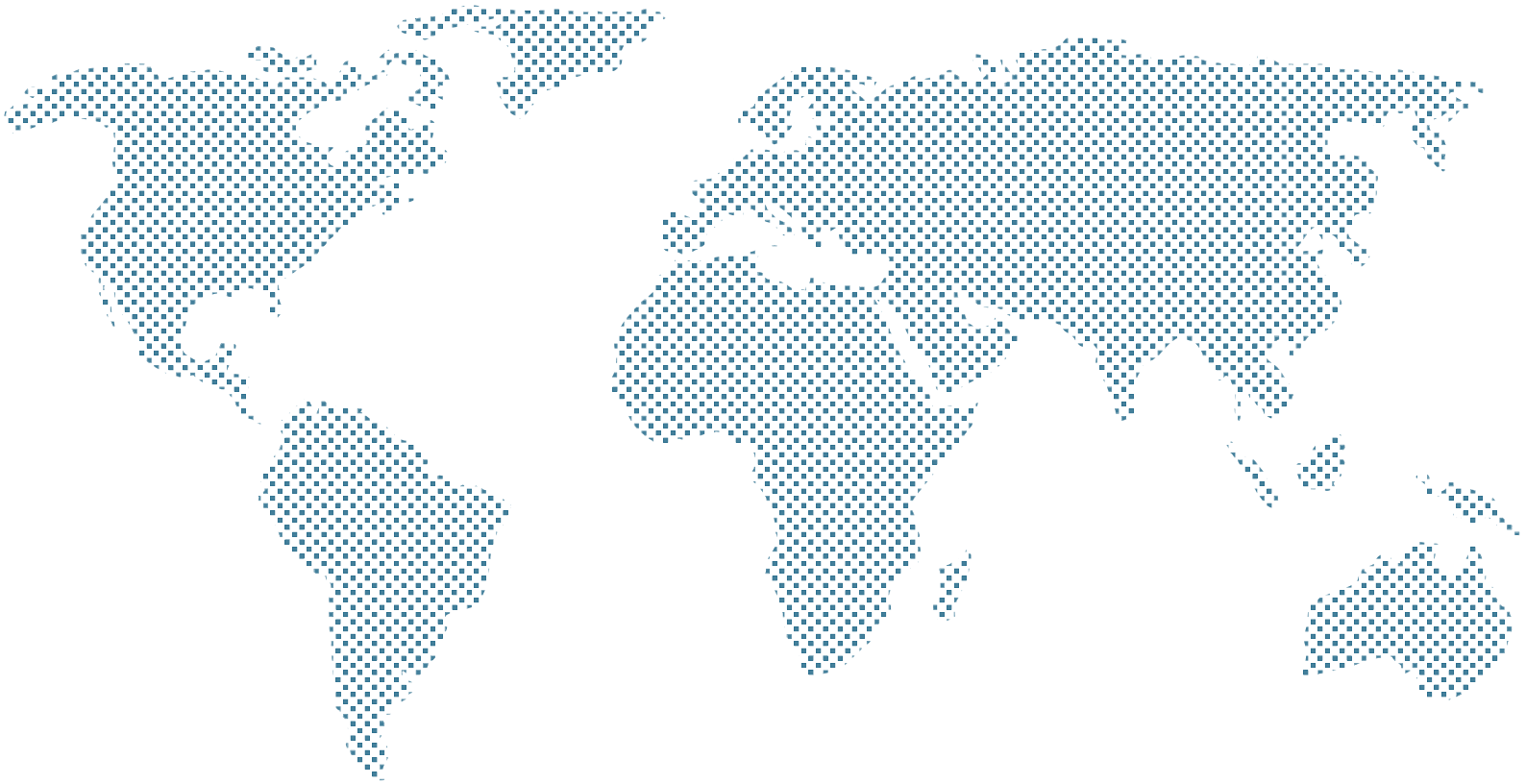
RAPPORT DE PRESENTATION

Databrick Spark Pipeline on Online Retail



16 JANVIER 2026

MSC 2 DATA ENGINEERING & CLOUD COMPUTING
AIVANCITY

PySpark et Delta Lake sur Online Retail

Dataset UCI Online Retail | Databricks (Serverless) | Unity Catalog Volumes
High level lakehouse with PySpark, Professeur Mohamed-Amine Baazizi, aivancity School of AI and Data
MSc 2 Data Engineering & Cloud Computing (2025–2026) Rapport de presentation
Karl Sondeji

1. Introduction

Imaginez les fiches de caisse d'une boutique en ligne avec des centaines de milliers de lignes, parfois incomplètes ou incohérentes, annulations mélangées aux ventes, prix aberrants. Sans traitement, impossible de répondre sereinement à des questions simples comme :

Quel est le chiffre d'affaires par pays ? Quels clients achètent le plus ? Quelles catégories de produits génèrent le plus de retours ?

C'est pourquoi j'ai industrialisé la chaîne données brutes vers données prêtes pour le reporting, avec des gains mesurables sur la fiabilité, la richesse analytique et la vitesse des requêtes.

J'ai conçu un pipeline automatisé qui prend ce fichier brut, le nettoie, l'enrichit (segments clients, catégories de produits, rattachement géographique) et le stocke dans un format adapté à l'analyse à grande échelle. Le tout est reproductible (une même commande produit toujours le même résultat) et contrôlé, si les données ne passent pas les règles de qualité, le traitement s'arrête plutôt que de publier des chiffres faux. Le pipeline conserve également l'historique de chaque modification, ce qui permet de revenir sur une version antérieure des données ou de comparer deux états dans le temps.

Le contexte métier

Le jeu de données provient d'une boutique en ligne britannique (Online Retail, période fin 2010 à fin 2011) et compte près de 540 000 lignes de transactions. Ces données présentent les problèmes classiques d'une plateforme de retail en ligne, avec des informations manquantes sur les clients, le prix ou la quantité, des annulations mélangées aux vraies transactions, des valeurs aberrantes sur les quantités ou les prix, et une structure brute peu adaptée au calcul direct d'indicateurs métier comme le chiffre d'affaires ou la segmentation client.

Les objectifs

Le premier objectif est de fiabiliser le chiffre d'affaires et les segments clients, pour que les décisions commerciales s'appuient sur des chiffres cohérents plutôt que sur des données brutes non vérifiées.

Le deuxième est de garantir la qualité à chaque chargement, afin d'éviter que des erreurs de saisie ou des incohérences ne se propagent jusque dans les tableaux de bord finaux.

Le troisième est d'accélérer les requêtes d'analyse, notamment par un partitionnement pensé pour les schémas d'interrogation les plus fréquents, afin de réduire le temps d'attente pour les équipes métier ou data.

Le dernier est de permettre l'évolution des données dans le temps, avec un historique consultable, des corrections traçables et des mises à jour automatisées plutôt que des remplacements silencieux.

2. Architecture du pipeline

Le pipeline suit une architecture en médaillon, un standard de l'ingénierie des données qui organise le traitement en couches successives, chacune apportant un niveau de qualité et de structuration supérieur à la précédente. Cette organisation permet de toujours remonter à la source en cas de problème, plutôt que de travailler sur des données déjà transformées sans trace de leur origine.

La couche brute reçoit le fichier CSV tel quel, sans aucune modification, et sert de point de départ immuable, si une erreur est détectée plus loin dans la chaîne, on peut toujours repartir de cette version d'origine.

La couche bronze ingère ensuite ces données dans un format structuré et versionné, en conservant l'intégralité des lignes, y compris celles qui présentent des anomalies, l'idée étant de ne rien perdre à ce stade, uniquement de rendre les données exploitables techniquement.

La couche silver applique les règles de nettoyage : suppression des lignes sans identifiant client lorsque l'analyse l'exige, élimination des doublons, filtrage des valeurs aberrantes sur les prix et les quantités, et distinction claire entre les ventes et les annulations. C'est également à ce niveau que des contrôles de qualité automatisés valident les données avant de les laisser progresser, si un contrôle échoue, le pipeline s'arrête et signale l'anomalie plutôt que de publier des chiffres non fiables en aval.

La couche gold enrichit enfin les données nettoyées avec les informations nécessaires au reporting métier, à savoir la segmentation des clients selon leur comportement d'achat, catégorisation des produits, rattachement géographique par pays et continent. C'est cette couche, organisée et partitionnée selon les axes d'analyse les plus fréquents, comme le pays ou la période, qui sert de socle direct aux tableaux de bord et aux requêtes d'analyse.

L'ensemble repose sur le format Delta Lake, qui ajoute à chaque couche un historique des versions et la possibilité de revenir en arrière ou de comparer deux états dans le temps. Les mises à jour de données ne remplacent donc jamais silencieusement l'existant, elles s'inscrivent dans un historique consultable, ce qui permet de tracer précisément l'effet de chaque traitement et de garantir que le pipeline reste reproductible d'un bout à l'autre, de la donnée brute jusqu'au chiffre affiché dans un tableau de bord.

3. Les différentes phases

Le pipeline se déroule en trois phases distinctes, chacune correspondant à une responsabilité précise et à des choix techniques justifiés par la nature du problème rencontré à cette étape.

L'ingestion consiste à lire le fichier CSV source et à l'écrire dans un format structuré et versionné. Le choix s'est porté sur une lecture avec un schéma explicite plutôt que sur l'inférence automatique de Spark, afin d'éviter qu'un typage incorrect ne se glisse silencieusement dans le pipeline dès la première étape. Un prix (UnitPrice) interprété comme une chaîne de caractères plutôt qu'un nombre, par exemple, fausserait tous les calculs en aval sans qu'aucune erreur ne soit levée. Les données sont ensuite écrites au format Delta plutôt que directement en Parquet classique, ce qui permet dès cette couche de bénéficier du versionnement et de la traçabilité des écritures, y compris sur des données encore brutes. Aucune ligne n'est filtrée à ce stade, y compris celles présentant des valeurs manifestement incohérentes, l'objectif de la couche bronze est de rendre la donnée techniquement exploitable, pas de la traitée.

```python
def write_delta(
    df: DataFrame,
    path: str,
    mode: str = "overwrite",
    partition_cols: list[str] | None = None,
    *,
    merge_schema: bool = False,
    overwrite_schema: bool = False,
) -> None:
    """Écrit un DataFrame au format Delta Lake sur un chemin local ou cloud."""
    if not path.startswith(_REMOTE_PREFIXES):
        Path(path).parent.mkdir(parents=True, exist_ok=True)

    writer = df.write.format("delta").mode(mode)
    if merge_schema:
        writer = writer.option("mergeSchema", "true")
    if overwrite_schema:
        writer = writer.option("overwriteSchema", "true")
    if partition_cols:
        writer = writer.partitionBy(*partition_cols)
    writer.save(path)
```

Le nettoyage est la phase la plus critique du pipeline, puisque c'est elle qui détermine la fiabilité de tout ce qui suit. Elle applique successivement plusieurs règles : élimination des lignes sans identifiant client lorsque l'analyse en dépend, suppression des doublons stricts identifiés par la combinaison numéro de facture, code produit et client, et filtrage des valeurs de prix ou de quantité incohérentes avec la réalité métier d'une transaction. Le choix a été fait de traiter les annulations, reconnaissables par un numéro de facture préfixé (précédée par un

« C » pour Cancelled), comme une catégorie de transaction à part entière plutôt que de les supprimer purement et simplement. Cela permet de conserver une information analytique utile, comme le taux de retour, qui aurait été perdue par un filtrage aveugle. Cette phase se termine par des contrôles de qualité automatisés qui valident notamment l'absence de valeurs nulles sur les colonnes critiques et la cohérence du volume de lignes après nettoyage. Ces contrôles conditionnent la suite du traitement (un échec interrompt le pipeline), et matérialisent concrètement le principe de fiabilité mentionné en introduction.

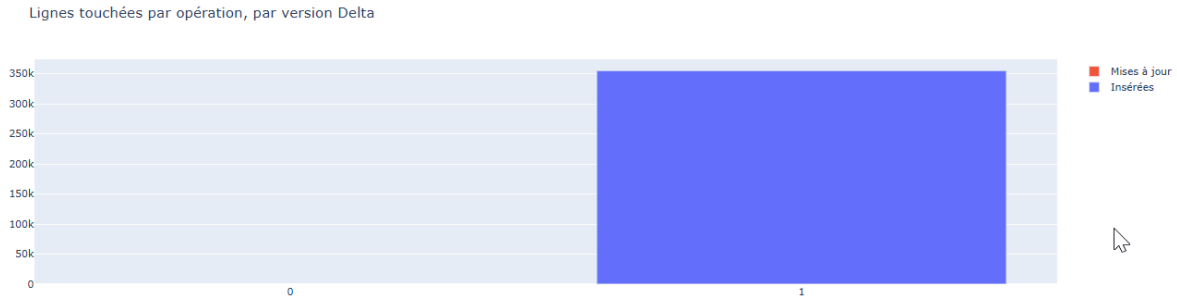
```python
def clean_transactions(df: DataFrame) -> DataFrame:
    quantity_as_str = col("Quantity").cast("string")
    unit_price_as_str = col("UnitPrice").cast("string")
    invoice_date_as_str = col("InvoiceDate").cast("string")

    return (
        _drop_corrupt_records(df)
        .filter(col("CustomerID").isNotNull())
        .filter(col("UnitPrice").isNotNull())
        .filter(col("Quantity").isNotNull())
        .filter(col("InvoiceNo").isNotNull())
        .filter(~lower(col("InvoiceNo")).startswith("c"))
        .filter(col("InvoiceNo") != "541431")
        .filter(length(col("StockCode")) == 5)
        .filter(quantity_as_str.rlike(_QUANTITY_PATTERN))
        .filter(unit_price_as_str.rlike(_UNIT_PRICE_PATTERN))
```

L'enrichissement ajoute aux données nettoyées les colonnes nécessaires à l'analyse métier, sans lesquelles chaque requête de reporting aurait dû recalculer les mêmes transformations à chaque exécution (coûteux en calcul). Le montant de chaque ligne de commande est calculé une fois pour toutes à partir de la quantité et du prix unitaire, plutôt que d'être recalculé dans chaque requête d'analyse en aval, ce qui simplifie et uniformise tous les calculs de chiffre d'affaires qui en dépendent. Les clients sont répartis en segments d'achat, les produits sont rattachés à des catégories déduites de leur description, et les pays sont associés à leur continent, autant d'axes de lecture qui n'existent pas dans le fichier source et qui transforment une liste de transactions en un jeu de données directement interrogeable pour répondre aux questions métier posées en introduction. C'est également à ce niveau qu'intervient le choix du partitionnement physique des données, pensé en fonction des filtres les plus fréquents dans les requêtes d'analyse, de façon à ce que le gain de performance obtenu à l'enrichissement se retrouve concrètement au moment de l'exploitation par les équipes métier.

```python
def _continent_from_country(country_col: Column = col("Country")) -> Column:
    """Mapping notebook : Country.contains(...) → Europa, Asia, etc."""
    if not COUNTRY_CONTINENT_CONTAINS:
        return lit("Other")

    first_country, first_continent = COUNTRY_CONTINENT_CONTAINS[0]
    result = when(country_col.contains(first_country), lit(first_continent))
    for country, continent in COUNTRY_CONTINENT_CONTAINS[1:]:
        result = result.when(country_col.contains(country), lit(continent))
    return result.otherwise(lit("Other"))


def enrich_transactions(df):
    order_amount = col("Quantity") * col("UnitPrice")

    return (
        df.withColumnRenamed("StockCode", "ItemCode")
        .withColumn("desc_clean", lower(col("Description")))
        .withColumn("OrderAmount", order_amount)
        # Notebook : segments par montant de ligne, pas par CustomerID
        .withColumn(
            "Purchase_segment",
            when(order_amount < lit(5), lit("Low"))
            .when(order_amount < lit(20), lit("Medium"))
            .otherwise(lit("High")),
        )
```

4. Consolidation Delta et partitionnement

Au-delà des trois phases principales, le pipeline intègre une étape de consolidation qui met à l'épreuve la robustesse du mécanisme de mise à jour avant de le considérer comme fiable. Plutôt que de supposer que les règles de fusion fonctionnent correctement, un échantillon des données de la phase 4 est isolé et confronté au reste de la base via une opération MERGE, qui insère les nouvelles lignes et met à jour les correspondances existantes selon la combinaison numéro de facture, code produit et client. Cette simulation permet de vérifier concrètement que la logique de correspondance produit le résultat attendu, sans doublon ni perte de ligne, avant qu'elle ne soit appliquée à des mises à jour réelles où une erreur silencieuse serait beaucoup plus coûteuse à détecter.



Le format Delta rend cette vérification particulièrement simple à mener, puisque chaque opération d'écriture, qu'il s'agisse d'une création de table ou d'un MERGE, génère automatiquement une nouvelle version consultable dans l'historique. La comparaison entre la version précédant la fusion et celle qui la suit, exploitée par une requête de type time travel, permet de chiffrer précisément l'écart introduit par l'opération, ligne par ligne et pays par pays si nécessaire. Cette approche remplace une vérification manuelle ou une confiance aveugle dans le bon déroulement de la fusion par une preuve chiffrée et reproductible, et constitue un filet de sécurité qui peut être rejoué à chaque nouvelle consolidation.

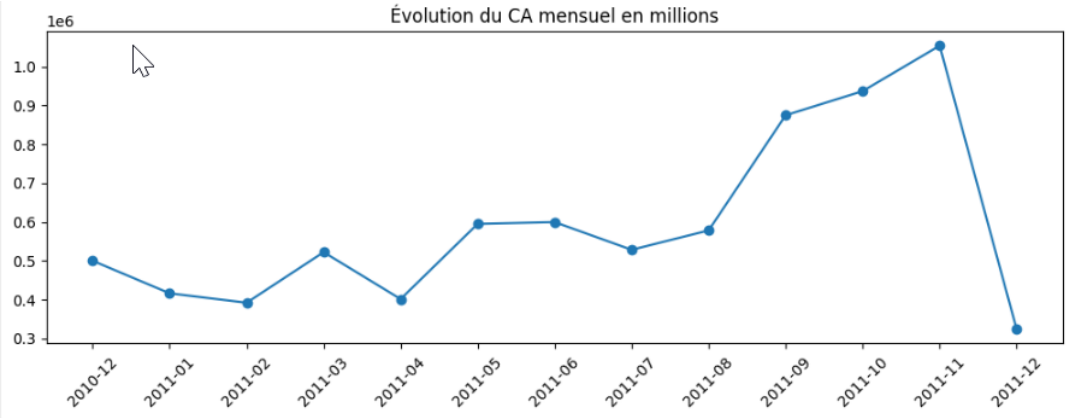
Le partitionnement physique des données gold vient compléter cette consolidation en organisant le stockage selon les axes d'interrogation les plus fréquents en analyse : le pays, la période mensuelle, et une combinaison hiérarchique continent puis pays. Cette dernière stratégie illustre un principe propre aux moteurs de traitement distribué, celui du pruning progressif, où une requête filtrant sur plusieurs niveaux de la hiérarchie élimine les données non pertinentes à chaque niveau successif plutôt qu'en une seule passe. Les mesures effectuées montrent un gain de temps de requête réel par rapport à une table non partitionnée, de l'ordre de 25 à 40% selon la stratégie retenue, avec dans certains cas une requête entièrement résolue à partir des métadonnées de la table sans lecture d'aucun fichier de données. Le choix d'inclure une stratégie hiérarchique n'apporte cependant pas de gain supplémentaire par rapport à un partitionnement simple sur cette taille de données précise : il a été conservé pour démontrer le principe et sa mécanique, avec la nuance assumée que son intérêt réel se manifeste surtout à une échelle de données nettement supérieure à celle de ce projet.

Table +

	strategy	1.2 flat_seconds	1.2 partitioned_seconds	1.2 speedup_pct
1	Country	0.4271	0.2619	38.7
2	year_month	0.3765	0.2795	25.8
3	Continent + Country	0.4474	0.2841	36.5

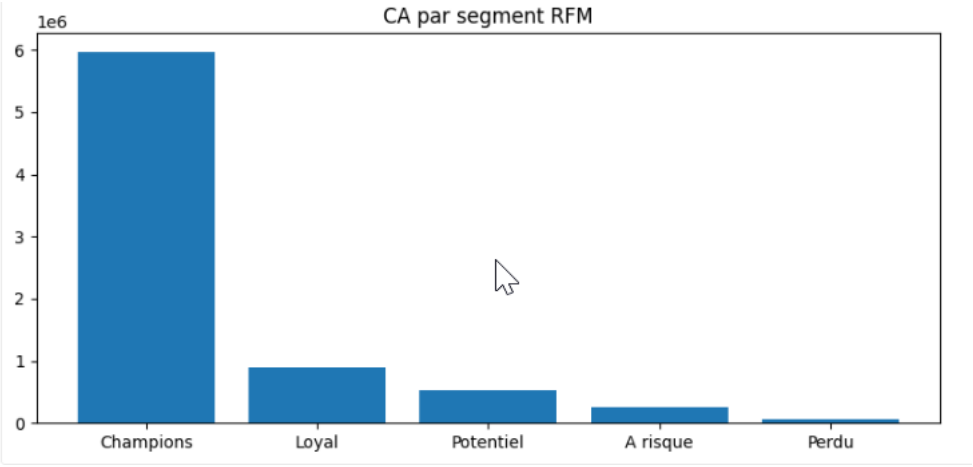
5. Analyses métier

L'analyse temporelle du chiffre d'affaires met en évidence une saisonnalité marquée sur la période couverte par le jeu de données. Le mois de novembre 2011 se détache nettement des autres avec près de 1,2 million de livres de chiffre d'affaires, un pic cohérent avec la préparation des achats de fin d'année dans le retail. Cette lecture mensuelle donne un premier axe de pilotage concret car elle permet d'anticiper les besoins en stock ou en logistique sur les périodes de forte activité.



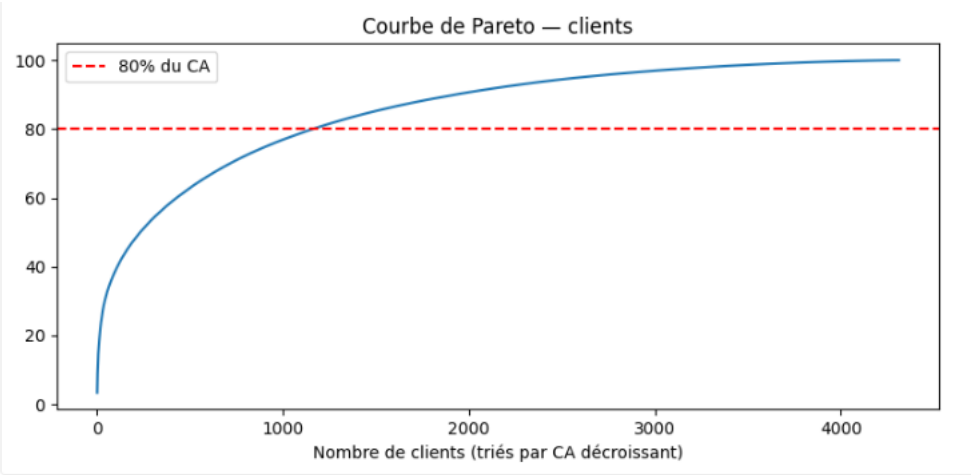
La segmentation RFM, construite à partir de la récence, de la fréquence et du montant des achats de chaque client, révèle une concentration de la valeur particulièrement forte, près de 80% du chiffre d'affaires provient des clients classés dans le segment champion, c'est-à-dire ceux ayant acheté récemment, fréquemment et pour des montants élevés.

Cette segmentation va plus loin que la logique d'un simple classement par montant total, puisqu'elle distingue un client fidèle et actif d'un client ayant simplement effectué un gros achat isolé, une nuance essentielle pour orienter des actions de fidélisation plutôt que des relances ponctuelles.



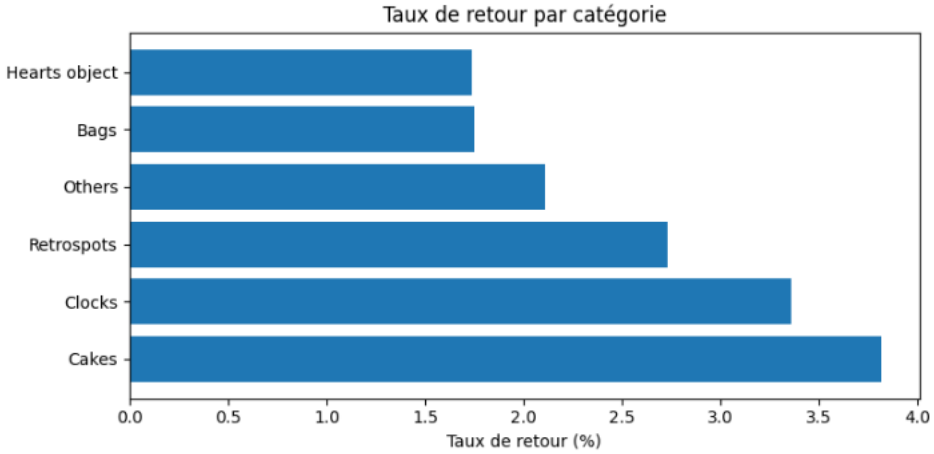
L'analyse de Pareto client confirme et quantifie précisément ce constat. Sur environ 4 300 clients actifs, un peu plus de 1 200 d'entre eux, soit environ 28% de la base, génèrent à eux seuls 80% du chiffre d'affaires total.

Cette lecture cumulative donne une base chiffrée directement actionnable pour prioriser les efforts commerciaux sur un sous-ensemble de clients identifiable plutôt que de traiter l'ensemble de la base de façon uniforme.



Le taux de retour par catégorie met en lumière les produits qui posent le plus de difficultés après achat.

Le taux moyen tous produits confondus s'établit autour de 2%, un niveau globalement maîtrisé, mais la catégorie Cakes se distingue avec un taux proche de 3,8%, sensiblement au-dessus de la moyenne. Cet écart, invisible dans une lecture globale du chiffre d'affaires, désigne une catégorie à surveiller en priorité, que ce soit sur la qualité du produit, sa description ou son adéquation avec l'attente du client.



Enfin, l'analyse des top produits, croisée entre chiffre d'affaires et volume, distingue deux logiques de succès commercial différentes.

En valeur, le produit *Regency Cakestand 3 Tier* domine largement avec plus de 142 000 livres de chiffre d'affaires généré, suivi par des articles comme la guirlande *Party Bunting* ou les ornements en forme d'oiseaux. En volume, un classement différent apparaît, porté par des articles à faible prix unitaire mais vendus en très grande quantité, à l'image des planeurs à thème Seconde Guerre mondiale, écoulés à plus de 54 000 unités. Ce double classement est utile pour distinguer les produits qui structurent la rentabilité de ceux qui génèrent du trafic et

de la rotation, deux leviers différents à ne pas traiter de la même façon dans une stratégie de mise en avant ou de promotion.

Table +

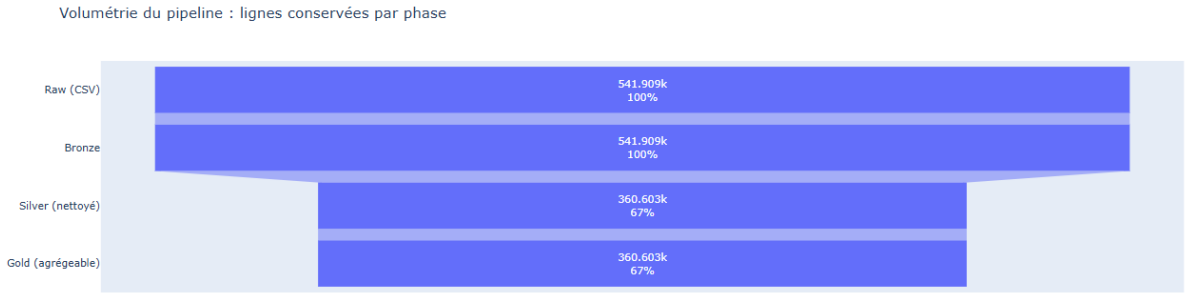
	ItemCode	Description	total_quantity
1	84077	WORLD WAR 2 GLIDERS ASSTD DESIGNS	54415
2	84879	ASSORTED COLOUR BIRD ORNAMENT	35362
3	21212	PACK OF 72 RETROSPOT CAKE CASES	33693
4	22197	POPCORN HOLDER	30931
5	23084	RABBIT NIGHT LIGHT	27202
6	22492	MINI PAINT SET VINTAGE	26076
7	22616	PACK OF 12 LONDON TISSUES	25345
8	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24264
9	17003	BROCADE RING PURSE	22963
10	22178	VICTORIAN GLASS HANGING T-LIGHT	22433

Table +

	ItemCode	Description	1.2 total_revenue
1	22423	REGENCY CAKESTAND 3 TIER	142592.95
2	47566	PARTY BUNTING	68844.33
3	84879	ASSORTED COLOUR BIRD ORNAMENT	56580.34
4	23084	RABBIT NIGHT LIGHT	51346.2
5	79321	CHILLI LIGHTS	46286.51
6	22086	PAPER CHAIN KIT 50'S CHRISTMAS	42660.83
7	22502	PICNIC BASKET WICKER 60 PIECES	39619.5
8	21137	BLACK RECORD COVER FRAME	39064.55
9	22386	JUMBO BAG PINK POLKADOT	37289.59
10	23284	DOORMAT KEEP CALM AND COME IN	35913.85

6. Qualité des données et rejet

Afin d'analyser plus finement quels sont les variables les plus impactées par le nettoyage des données et l'ajout des contraintes, j'ai réalisé un tableau et un graphique permettant de les voir plus en détail.




```
INFO:src.utils.logger:Rejection breakdown : lecture CSV raw /Volumes/main/default/raw/Online_Retail.csv
{'Total bronze': 541909,
 'CustomerID null': 133931,
 'UnitPrice null': 1239,
 'Quantity null': 1239,
 'InvoiceNo null': 0,
 'Annulation (InvoiceNo c*)': 9288,
 'InvoiceNo 541431': 1,
 'StockCode longueur != 5': 54170,
 'Quantity format invalide': 4796,
 'UnitPrice format invalide': 3729,
 'UnitPrice contient / ou :': 3729,
 'InvoiceDate format invalide': 4764,
 'Quantity non castable (int)': 6035,
 'UnitPrice non castable (double)': 4968,
 'Quantity <= 0 (après cast)': 10545,
 'UnitPrice <= 0 (après cast)': 2532,
 'Doublons (InvoiceNo+StockCode)': 9689}
```

Sur les 540k lignes de départ près d'un tiers de celles-ci sont supprimées (181 306) après nettoyage, et avec la sortie ci-dessus on voit que la majeure partie de ces suppressions provient de lignes sans *`Customer ID`* (133 931) ou dont le code produit n'est pas valide (54 170). À noter que ces différentes erreurs peuvent se combiner (pas d'identifiant client et pas de code produit valable).

7. Partitionnement et performances

Le choix du partitionnement physique de la couche gold s'est porté sur plusieurs stratégies testées en parallèle plutôt que sur une seule solution imposée d'emblée, afin de comparer objectivement leur intérêt respectif. Trois axes ont été retenus. Un partitionnement par pays, un partitionnement par mois d'achat, et un partitionnement hiérarchique combinant continent puis pays. Ces axes correspondent aux filtres les plus naturels d'une analyse de retail en ligne, que ce soit pour isoler un marché géographique ou pour observer une période donnée, et ils permettent d'illustrer à la fois un partitionnement simple et un partitionnement à plusieurs niveaux.

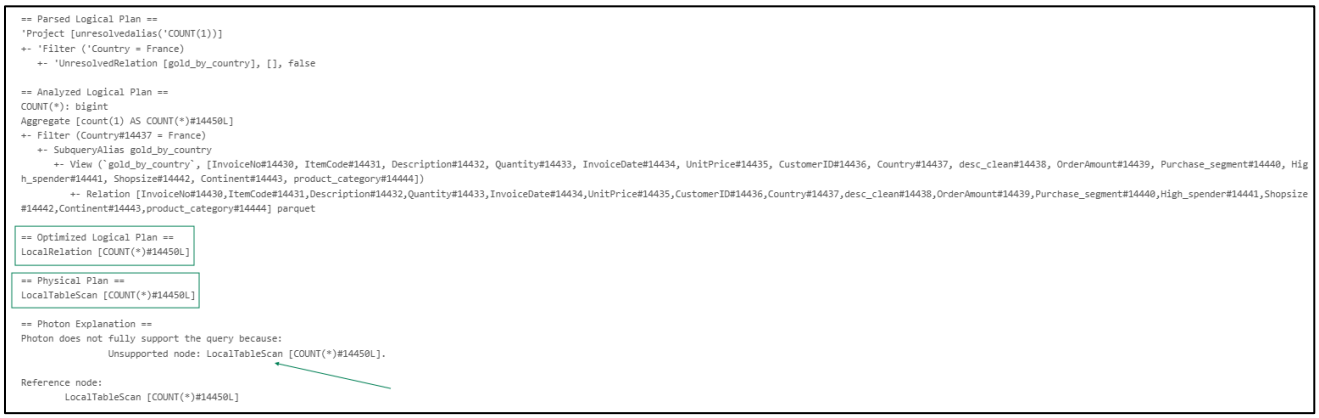
L'impact sur les performances de requête est net et mesurable. Sur chacune des trois stratégies, filtrer une table partitionnée plutôt que la table gold complète réduit le temps de réponse de 25 à 40% selon le cas, un gain obtenu simplement en évitant de parcourir des données non pertinentes pour la requête posée.

Ce résultat vérifie concrètement le principe du partition pruning, plutôt que de lire l'ensemble des fichiers puis de filtrer les lignes après lecture, le moteur élimine directement les répertoires qui ne peuvent pas contenir de résultat, avant même d'ouvrir un seul fichier de données.

Ces chiffres doivent néanmoins être remis en contexte pour rester honnêtes sur leur portée. Le fichier source ne représente qu'environ 540 000 lignes et 48 mégaoctets de données brutes, un volume nettement inférieur aux échelles auxquelles le partitionnement révèle habituellement son plein intérêt (généralement à partir de plusieurs gigaoctets voire téraoctets) et où la réduction du nombre de fichiers scannés devient déterminante face à des temps de requête qui se compteraient sinon en minutes.

Sur un volume aussi restreint, le gain de performance observé reste réel mais modeste en valeur absolue, et le partitionnement hiérarchique continent puis pays n'apporte d'ailleurs aucun avantage supplémentaire par rapport à un partitionnement simple par pays, la sélectivité du pays seul étant déjà suffisante pour éliminer l'essentiel des données non pertinentes. Cette stratégie a été conservée non pas pour son gain mesuré, mais pour illustrer le principe du pruning progressif à plusieurs niveaux, un mécanisme dont l'intérêt se manifesterait pleinement sur un jeu de données organisé selon une hiérarchie géographique beaucoup plus large.

Le résultat le plus parlant ne vient toutefois pas des temps mesurés, mais de la comparaison des plans d'exécution générés par Spark pour une même requête sur la table partitionnée et sur la table non partitionnée.



Sur la table non partitionnée, le plan physique montre un scan réel des fichiers Parquet, avec un filtre sur le pays appliqué après lecture des données.

Sur la table partitionnée par pays, le plan optimisé se réduit à un simple LocalTableScan car la requête est intégralement résolue à partir des statistiques conservées dans l'historique Delta, sans qu'aucune donnée ne soit effectivement lue. Ce contraste, visible directement dans le plan d'exécution plutôt que déduit d'un simple chronométrage, constitue la preuve la plus concrète que le partitionnement ne se contente pas d'accélérer la lecture des données, il peut dans certains cas supprimer entièrement le besoin d'y accéder, déplaçant le travail du moteur de calcul vers la seule couche de métadonnées.

8. Conclusion

Ce projet répond au constat posé en introduction, qui est de pouvoir transformer un fichier de données brutes, incomplètes et incohérentes, en une base fiable capable de répondre sereinement aux questions métier d'une boutique en ligne. Le pipeline mis en place, organisé en couches bronze, silver et gold, a permis d'éliminer près d'un tiers des enregistrements bruts, soit environ 181 000 lignes malformées ou inexploitables, avant même d'aborder la moindre analyse. Cette étape de nettoyage, souvent moins visible que les résultats finaux, conditionne pourtant la validité de tout ce qui en découle, sans elle, les chiffres de chiffre d'affaires ou de segmentation client présentés plus haut n'auraient aucune fiabilité.

Les analyses métier conduites sur les données ainsi consolidées dessinent une stratégie claire. La segmentation RFM et l'analyse de Pareto convergent vers un même constat à savoir qu'une minorité de clients, environ 28% de la base, concentre 80% du chiffre d'affaires, ce qui justifie de prioriser les actions de fidélisation sur ce segment plutôt que de traiter l'ensemble de la clientèle de façon uniforme. De la même manière, l'identification des produits qui dominent en valeur et de ceux qui dominent en volume donne deux leviers distincts à actionner selon que l'objectif soit la rentabilité ou le trafic, et le taux de retour par catégorie signale un point d'attention précis sur la catégorie *Cakes* plutôt qu'un problème de qualité diffus et difficile à cibler.

La consolidation des données par fusion d'un échantillon avec le reste de la base a par ailleurs confirmé la robustesse des règles de nettoyage mises en place, aucune ligne n'a nécessité de mise à jour lors de cette fusion, ce qui valide la standardisation du traitement plutôt que de la supposer. Enfin, la mise en place de plusieurs stratégies de partitionnement a démontré un gain de performance mesurable sur les requêtes filtrées, avec dans certains cas une requête entièrement résolue depuis les métadonnées sans lecture de données. Ce résultat reste à interpréter à l'échelle du projet, le volume de données traité étant volontairement modeste, mais il illustre un principe dont l'intérêt ne peut que croître face à des volumes de production réels, ce qui constitue précisément l'objectif pédagogique de cette démonstration.

Au-delà des résultats obtenus sur ce jeu de données précis, ce projet a permis de mettre en pratique l'ensemble d'une chaîne de traitement de données de bout en bout, de l'ingestion brute jusqu'au reporting analytique, en intégrant les exigences propres à un contexte de production :

- Qualité contrôlée
- Traçabilité des versions
- Performance des requêtes pensée dès la conception du stockage

Merci !