

Projet Databricks : Transformation des Données en Zones Bronze, Silver et Gold

Ce document récapitule toutes les étapes effectuées, de l'importation des données jusqu'à leur analyse dans la zone Gold.

Contexte : Nous avons récupéré une base de données regroupant un certain nombre d'indicateurs de santé et de facteurs de risques liés aux maladies cardiaques sur un panel de 9500 personnes sur le lien suivant :

<https://www.kaggle.com/datasets/oktayrdeki/heart-disease?resource=download>

Ex :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Age	Sexe	Tension_arte	Taux_de_chc	Habitudes_s	Tabagisme	Antecedents	Diabete	IMC	Hypertension	Faible_chole	Cholesterol	Consommation	Niveau_de_s	Heures_de_s	Consommation	Taux_de_trig	Glycemie_a_j	Niveau_de_C	Niveau_hom	Maladie_card
2	56	Male	153	155	High	Yes	Yes	No	24.9915911	Yes	Yes	No	High	Medium	7.63322838	Medium	342	12.9692457	12.3872504	No	
3	69	Female	146	206	High	No	Yes	Yes	25.2217965	No	Yes	No	Medium	High	8.74483397	Medium	133	9.35538941	19.2968755	No	
4	46	Male	126	216	Low	No	No	No	29.8554471	No	Yes	Yes	Low	Low	4.44044012	Low	393	92	12.7098725	11.2309257	No
5	32	Female	122	293	High	Yes	Yes	No	24.1304769	Yes	No	Yes	Low	High	5.2494047	High	293	94	12.5090462	5.96195807	No
6	60	Male	166	242	Low	Yes	Yes	Yes	20.4862889	Yes	No	No	Low	High	7.03097143	High	263	154	10.3812592	8.15388692	No
7	25	Male	152	257	Low	Yes	No	No	28.1446815	No	No	No	Low	Medium	5.50487565	Low	126	91	4.29757473	10.8159827	No

1. Importation des données dans la zone Bronze

Objectif : Créer une table brute contenant les données initiales sans modification.

Requête SQL :

```
%sql
CREATE OR REPLACE TABLE default.BronzeHeartDisease AS
SELECT *
FROM default.heart_disease_csv;
```

- **Action effectuée :** Les données issues du fichier CSV ont été importées dans la table BronzeHeartDisease.
- **Objectif :** Stocker les données telles quelles, sans transformation, dans une table intermédiaire.

2. Transformation et nettoyage des données dans la zone Silver

Objectif : Supprimer les lignes contenant des cellules vides, convertir les colonnes numériques et enrichir les données.

2.1 Suppression des lignes contenant des cellules NULL

Requête SQL :

```
%sql
CREATE OR REPLACE TABLE default.SilverHeartDisease AS
SELECT *
FROM default.BronzeHeartDisease
WHERE
    Age IS NOT NULL
    AND Sexe IS NOT NULL
    AND Tension_arterielle IS NOT NULL
    AND Taux_de_cholesterol IS NOT NULL
    AND Habitudes_sportives IS NOT NULL
    AND Tabagisme IS NOT NULL
    AND Antecedents_Cardiaques IS NOT NULL
    AND Diabete IS NOT NULL
    AND IMC IS NOT NULL
    AND Hypertension IS NOT NULL
    AND Faible_cholesterol_HDL IS NOT NULL
    AND Cholesterol_LDL_eleve IS NOT NULL
    AND `Consommation_d'alcool` IS NOT NULL
    AND Niveau_de_stress IS NOT NULL
    AND Heures_de_sommeil IS NOT NULL
    AND Consommation_de_sucre IS NOT NULL
    AND Taux_de_triglycerides IS NOT NULL
    AND Glycemie_a_jeun IS NOT NULL
    AND Niveau_de_CRP IS NOT NULL
    AND Niveau_homocysteine IS NOT NULL
    AND `Maladie_cardiaque_?` IS NOT NULL;
```

- **Action effectuée :** Toutes les lignes ayant au moins une cellule NULL ont été supprimées.

2.2 Conversion des colonnes à plusieurs décimales

Requête SQL :

```
%sql
CREATE OR REPLACE TABLE default.SilverHeartDisease AS
SELECT
    Age,
    Sexe,
    Tension_arterielle,
    Taux_de_cholesterol,
    Habitudes_sportives,
    Tabagisme,
    Antecedents_Cardiaques,
    Diabete,
    ROUND(CAST(REPLACE(IMC, ',', '.') AS DOUBLE), 2) AS IMC_Cleaned,
    Hypertension,
    Faible_cholesterol_HDL,
    Cholesterol_LDL_eleve,
    `Consommation_d'alcool`,
    Niveau_de_stress,
    ROUND(CAST(REPLACE(Heures_de_sommeil, ',', '.') AS DOUBLE), 2) AS Heures_de_sommeil_Cleaned,
    Consommation_de_sucre,
    ROUND(CAST(REPLACE(Taux_de_triglycerides, ',', '.') AS DOUBLE), 2) AS Taux_de_triglycerides_Cleaned,
    ROUND(CAST(REPLACE(Glycemie_a_jeun, ',', '.') AS DOUBLE), 2) AS Glycemie_a_jeun_Cleaned,
    ROUND(CAST(REPLACE(Niveau_de_CRP, ',', '.') AS DOUBLE), 2) AS Niveau_de_CRP_Cleaned,
    ROUND(CAST(REPLACE(Niveau_homocysteine, ',', '.') AS DOUBLE), 2) AS Niveau_homocysteine_Cleaned,
    `Maladie_cardiaque_?`
FROM default.BronzeHeartDisease
WHERE
    Age IS NOT NULL
    AND Sexe IS NOT NULL
    AND Tension_arterielle IS NOT NULL
    AND Taux_de_cholesterol IS NOT NULL
    AND Habitudes_sportives IS NOT NULL
    AND Tabagisme IS NOT NULL
    AND Antecedents_Cardiaques IS NOT NULL
    AND Diabete IS NOT NULL
    AND IMC IS NOT NULL
    AND Hypertension IS NOT NULL
    AND Faible_cholesterol_HDL IS NOT NULL
    AND Cholesterol_LDL_eleve IS NOT NULL
    AND `Consommation_d'alcool` IS NOT NULL
    AND Niveau_de_stress IS NOT NULL
    AND Heures_de_sommeil IS NOT NULL
    AND Consommation_de_sucre IS NOT NULL
    AND Taux_de_triglycerides IS NOT NULL
    AND Glycemie_a_jeun IS NOT NULL
    AND Niveau_de_CRP IS NOT NULL
    AND Niveau_homocysteine IS NOT NULL
    AND `Maladie_cardiaque_?` IS NOT NULL;
```

- **Action effectuée :** Toutes les colonnes contenant des valeurs avec plusieurs décimales ont été converties en format à deux décimales.
- **Astuce technique :** Les virgules ont été remplacées par des points avant la conversion.

2.3 Enrichissement des données : Ajout de colonnes calculées

Requête SQL :

```
%sql
CREATE OR REPLACE TABLE default.SilverHeartDisease AS
SELECT
    *,
    CASE
        WHEN IMC_Cleaned < 18.5 THEN 'Maigreur'
        WHEN IMC_Cleaned >= 18.5 AND IMC_Cleaned < 25 THEN 'Normal'
        WHEN IMC_Cleaned >= 25 AND IMC_Cleaned < 30 THEN 'Surpoids'
        ELSE 'Obésité'
    END AS IMC_Category,
    CASE
        WHEN IMC_Cleaned < 18.5 THEN 'Risque élevé'
        WHEN IMC_Cleaned >= 18.5 AND IMC_Cleaned < 25 THEN 'Risque faible'
        WHEN IMC_Cleaned >= 25 AND IMC_Cleaned < 30 THEN 'Risque modéré'
        ELSE 'Risque élevé'
    END AS Risk_Score
FROM default.SilverHeartDisease;
```

Ajout des colonnes :

- **IMC_Category** : Catégorisation de l'IMC (Maigreur, Normal, Surpoids, Obésité).
- **Risk_Score** : Calcul du risque associé à l'IMC.

3. Transition vers la zone Gold

Objectif : Réutiliser les données Silver sans modifications supplémentaires.

Requête SQL :

```
%sql
CREATE OR REPLACE TABLE default.GoldHeartDisease AS
SELECT *
FROM default.SilverHeartDisease;
```

- **Action effectuée** : Les données Silver ont été copiées dans la table Gold sans transformation.
 - **Objectif** : La zone Gold représente les données finales, prêtes pour l'analyse.
-

4. Analyses et requêtes sur la zone Gold

Objectif : Extraire des statistiques et tendances à partir des données Gold.

4.1 Pourcentage de patients par catégorie IMC

Just now (3s) 1

```
%sql
SELECT
  IMC_Category,
  COUNT(*) AS Total_Patients,
  ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM default.GoldHeartDisease), 2) AS Percentage
FROM default.GoldHeartDisease
GROUP BY IMC_Category
ORDER BY Percentage DESC;
```

▶ (7) Spark Jobs

▶ _sqldf: pyspark.sql.dataframe.DataFrame = [IMC_Category: string, Total_Patients: long ... 1 more field]

Table +

	IMC_Category	Total_Patients	Percentage
1	Obésité	4352	45.81
2	Normal	2696	28.38
3	Surpoids	2238	23.56
4	Maigre	214	2.25

4 rows | 3.14 seconds runtime

4.2 Moyenne de l'IMC par catégorie de risque

Just now (1s)

```
%sql
SELECT
  Risk_Score,
  ROUND(AVG(IMC_Cleaned), 2) AS Avg_IMC
FROM default.GoldHeartDisease
GROUP BY Risk_Score
ORDER BY Avg_IMC DESC;
```

▶ (2) Spark Jobs

▶ _sqldf: pyspark.sql.dataframe.DataFrame = [Risk_Score: string, Avg_IMC: double]

Table +

	Risk_Score	Avg_IMC
1	Élevé	34.16
2	Moyen	27.48
3	Faible	21.72

4.3 Pourcentage de patients atteints de maladies cardiaques

```
%sql
SELECT
  `Maladie_cardiaque_?` AS Heart_Disease_Status,
  COUNT(*) AS Total_Patients,
  ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM default.GoldHeartDisease), 2) AS Percentage
FROM default.GoldHeartDisease
GROUP BY `Maladie_cardiaque_?`
ORDER BY Percentage DESC;
```

► (6) Spark Jobs

► _sqldf: pyspark.sql.dataframe.DataFrame = [Heart_Disease_Status: string, Total_Patients: long ... 1 more field]

Table

	^A _C Heart_Disease_Status	¹ ₃ Total_Patients	.00 Percentage
1	No	7597	79.97
2	Yes	1903	20.03

4.4 Corrélation entre le tabagisme et les maladies cardiaques

```
%sql
SELECT
  Tabagisme AS Smoking_Status,
  `Maladie_cardiaque_?` AS Heart_Disease_Status,
  COUNT(*) AS Total_Patients
FROM default.GoldHeartDisease
GROUP BY Tabagisme, `Maladie_cardiaque_?`
ORDER BY Smoking_Status, Heart_Disease_Status;
```

► (2) Spark Jobs

► _sqldf: pyspark.sql.dataframe.DataFrame = [Smoking_Status: string, Heart_Disease_Status: str]

Table

	^A _C Smoking_Status	^A _C Heart_Disease_Status	¹ ₃ Total_Patients
1	No	No	3711
2	No	Yes	916
3	Yes	No	3886
4	Yes	Yes	987

4.5 Pourcentage de patients ayant un stress élevé

```
%sql
SELECT
  Niveau_de_stress AS Stress_Level,
  COUNT(*) AS Total_Patients,
  ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM default.GoldHeartDisease), 2) AS Percentage
FROM default.GoldHeartDisease
GROUP BY Niveau_de_stress
ORDER BY Percentage DESC;
```

▶ (6) Spark Jobs


▶  _sqldf: pyspark.sql.dataframe.DataFrame = [Stress_Level: string, Total_Patients: long ... 1 more field]

Table ▼ +

	^A _C Stress_Level	¹ ₃ Total_Patients	.00 Percentage
1	Medium	3222	33.92
2	Low	3158	33.24
3	High	3120	32.84

4.6 Habitudes sportives par catégorie IMC

```
%sql
SELECT
  Habitudes_sportives AS Exercise_Habits,
  IMC_Category,
  COUNT(*) AS Total_Patients
FROM default.GoldHeartDisease
GROUP BY Habitudes_sportives, IMC_Category
ORDER BY IMC_Category, Total_Patients DESC;
```

▶ (2) Spark Jobs


▶  _sqldf: pyspark.sql.dataframe.DataFrame = [Exercise_Habits: string, IMC_Category: str

Table ▼ +

	^A _C Exercise_Habits	^A _C IMC_Category	¹ ₃ Total_Patients
1	High	Maigreux	84
2	Medium	Maigreux	71
3	Low	Maigreux	59
4	Medium	Normal	910
5	High	Normal	896
6	Low	Normal	890
7	High	Obésité	1498
8	Low	Obésité	1430
9	Medium	Obésité	1424
10	Medium	Surpoids	761
11	High	Surpoids	741
12	Low	Surpoids	736

4.7 Moyenne des heures de sommeil en fonction du risque

```
%sql
SELECT
  Risk_Score,
  ROUND(AVG(Heures_de_sommeil_Cleaned), 2) AS Avg_Sleep_Hours
FROM default.GoldHeartDisease
GROUP BY Risk_Score
ORDER BY Avg_Sleep_Hours ASC;
```

► (2) Spark Jobs




►  _sqldf: pyspark.sql.dataframe.DataFrame = [Risk_Score: string, Avg_Sleep_Hours: float]

Table  

	^A _C Risk_Score	1.2 Avg_Sleep_Hours
1	Faible	6.97
2	Élevé	6.99
3	Moyen	7.01

5. Conclusion

Ce projet illustre un pipeline complet de traitement des données depuis leur importation brute jusqu'à leur préparation pour l'analyse. Les zones Bronze, Silver et Gold ont permis de structurer le processus :

- **Bronze** : Conservation des données brutes pour assurer leur traçabilité.
- **Silver** : Nettoyage, normalisation et enrichissement des données pour les rendre exploitables.
- **Gold** : Présentation des données finales prêtes pour des analyses avancées.

Les analyses effectuées dans la zone Gold, telles que les statistiques sur l'IMC ou les risques de maladies cardiaques, offrent des perspectives précieuses pour comprendre les tendances dans les données. Ce pipeline peut être adapté à d'autres contextes pour gérer efficacement des données complexes.