

Analyse des Données de Basket avec un DataWarehouse :

Nino Fazer, Jordy Pereira-Elenga-Makouala, Huu-nghia Tran, Romain Monmarche

Groupe - 4

06/01/2025

Table des matières

Introduction :	2
Présentation du projet	2
Technologies utilisées.....	2
Objectifs du projet.....	2
Description des données :	3
• Sources de données :	3
• Description des tables sources (Bronze) :.....	3
Processus de traitement - Visualisation :	4
Architecture des données	4
Configuration des points de montage.....	5
Pour chaque zone :	6
Modèle Conceptuel des Données (MCD) :.....	16
• Schéma modèle en étoile :.....	16
Conclusion :	17

Introduction :

Présentation du projet

Dans le cadre de ce projet, nous avons entrepris une analyse approfondie des données de la NBA afin d'extraire des insights clés concernant les joueurs, les équipes, et leurs performances au fil des saisons depuis 1947 jusqu'à maintenant. L'objectif principal est de construire un DataWarehouse robuste qui permet d'organiser et d'exploiter efficacement des données complexes pour répondre à des questions analytiques stratégiques, telles que par exemple :

- Quels joueurs dominent dans les tirs à 3 points ?
- Quelle équipe a participé le plus souvent aux playoffs ?
- Quels sont les joueurs avec le plus grand impact sur les matchs ?

Ce projet s'inscrit dans une démarche visant à créer un système de données durable et évolutif, tout en assurant une préparation optimale des données pour l'analyse et la visualisation.

Technologies utilisées

Pour répondre aux objectifs, plusieurs outils et technologies avancés ont été mobilisés :

- **Databricks** : Une plateforme de traitement de données en temps réel utilisée pour l'ingestion, la transformation, et le stockage des données à travers les zones Bronze, Silver, et Gold.
- **Azure** : Plateforme cloud offrant l'infrastructure nécessaire pour héberger et exécuter les pipelines de données.

Objectifs du projet

Construire un DataWarehouse avec un modèle en étoile

L'objectif central du projet est de créer un **DataWarehouse structuré** qui repose sur un **modèle en étoile**. Ce modèle organise les données en :

- **Tables de faits** : Contenant des mesures quantitatives (comme les performances par match ou par joueur).
- **Tables de dimensions** : Fournissant le contexte des mesures (joueurs, équipes, matchs, etc.).

Cette architecture permet d'effectuer des analyses rapides et efficaces sur des volumes de données importants, tout en maintenant une structure claire et normalisée.

Description des données :

- Sources de données :
 - Données extraites de Kaggle
<https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats>
- Description des tables sources (Bronze) :
 - Table des joueurs : *Informations personnelles des joueurs.*
 - Table des équipes : *Statistiques des équipes, noms, et saisons.*
 - Table des matchs : *Statistiques de la moyenne de la saison par match et par joueur.*

Accueil > ninfaz | Conteneurs >

ds-bronze

Conteneur

Rechercher

Charger

Ajouter un répertoire

Actualiser

Renommer

Supprimer

Modifier le niveau

Acquérir le bail

Résilier le bail

Envoyer des commentaires

Vue d'ensemble

Diagnostic et résoudre les problèmes

Contrôle d'accès (IAM)

Paramètres

Méthode d'authentification : Clé d'accès (Basculer vers le compte d'utilisateur Microsoft Entra)

Emplacement : ds-bronze / nba_data / current

Rechercher les objets blobs par préfixe (respect de la casse)

Afficher les objets supprimés

Nom	Modifié	Niveau d'accès	État de l'archive	Type d'objet blob	Taille	État du bail
<input type="checkbox"/> [-]						...
<input type="checkbox"/> Opponent Stats Per Game.csv	16/12/2024 19:05:15	Élevé (déduit)		Objet blob de blocs	258.15 KiB	Disponible ...
<input type="checkbox"/> Per 100 Poss.csv	06/01/2025 10:26:51	Élevé (déduit)		Objet blob de blocs	3.91 MiB	Disponible ...
<input type="checkbox"/> Per 36 Minutes.csv	16/12/2024 19:05:19	Élevé (déduit)		Objet blob de blocs	4.42 MiB	Disponible ...
<input type="checkbox"/> Player Career Info.csv	16/12/2024 19:05:16	Élevé (déduit)		Objet blob de blocs	204.38 KiB	Disponible ...
<input type="checkbox"/> Player Per Game.csv	16/12/2024 19:05:20	Élevé (déduit)		Objet blob de blocs	4.56 MiB	Disponible ...
<input type="checkbox"/> Team Stats Per Game.csv	16/12/2024 19:05:19	Élevé (déduit)		Objet blob de blocs	264.16 KiB	Disponible ...
<input type="checkbox"/> Team Summaries.csv	16/12/2024 19:05:19	Élevé (déduit)		Objet blob de blocs	319.4 KiB	Disponible ...
<input type="checkbox"/> Team Totals.csv	16/12/2024 19:05:19	Élevé (déduit)		Objet blob de blocs	267.05 KiB	Disponible ...

Processus de traitement - Visualisation :

Architecture des données






Afin de mener à bien ce projet d'analyse des données, nous avons mis en place une structure organisée sur la plateforme **Databricks** connectée à un environnement **Azure**.

Ci-dessous, voici l'organisation des différents notebooks et répertoires utilisés dans le projet. Chaque étape de transformation des données est structurée dans un dossier spécifique pour faciliter la maintenance et la gestion des workflows.

- **Part_Bronze** : Ingestion brute des données depuis la source.
- **Part_Silver** : Nettoyage et enrichissement des données.
- **Part_Gold** : Construction des tables analytiques prêtes pour les requêtes.
- **Initialisation** : Notebook contenant les scripts de configuration pour monter les points de montage sur Azure Blob Storage.

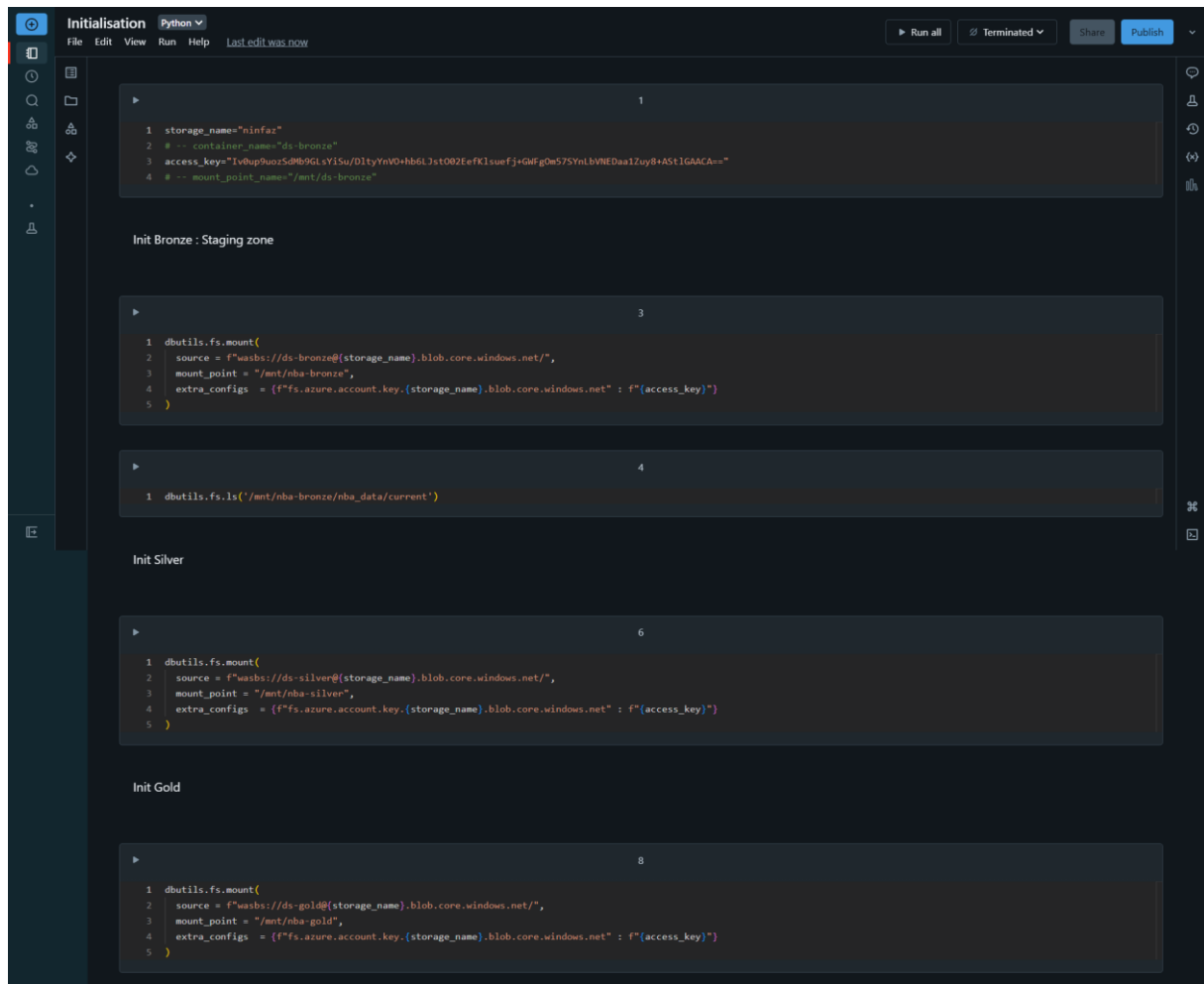
[Workspace](#) > [Shared](#) >

NBA_STATS

Name 	Type	Owner
 Part_Gold	Folder	Jordy Pereira-Elenga Makouala
 Part_Silver	Folder	Jordy Pereira-Elenga Makouala
 Initialisation	Notebook	Nino FAZER
 Part bronze	Notebook	Jordy Pereira-Elenga Makouala

Configuration des points de montage

Le stockage des données a été configuré en utilisant les conteneurs Azure Blob Storage. Trois zones de données ont été définies :



```
1 storage_name="ninfaz"
2 # -- container_names "ds-bronze"
3 access_key="Iv0up9uozSD0b9GLsYiSu/D1tyYnV0+hb6L7st002EefK1suefj+GwfgDw57SYnLbVNEDaalZuy8+AStIGACA=="
4 # -- mount_point_names "/mnt/ds-bronze"

Init Bronze : Staging zone

3
1 dbutils.fs.mount(
2     source = f"wasbs://{ds-bronze@{storage_name}.blob.core.windows.net/",
3     mount_point = "/mnt/nba-bronze",
4     extra_configs = {f"fs.azure.account.key.{storage_name}.blob.core.windows.net" : f"{access_key}"}
5 )

4
1 dbutils.fs.ls("/mnt/nba-bronze/nba_data/current")

Init Silver

6
1 dbutils.fs.mount(
2     source = f"wasbs://{ds-silver@{storage_name}.blob.core.windows.net/",
3     mount_point = "/mnt/nba-silver",
4     extra_configs = {f"fs.azure.account.key.{storage_name}.blob.core.windows.net" : f"{access_key}"}
5 )

Init Gold

8
1 dbutils.fs.mount(
2     source = f"wasbs://{ds-gold@{storage_name}.blob.core.windows.net/",
3     mount_point = "/mnt/nba-gold",
4     extra_configs = {f"fs.azure.account.key.{storage_name}.blob.core.windows.net" : f"{access_key}"}
5 )
```

Pour chaque zone :

- **Bronze :**

Cell 1: SQL

```
CREATE DATABASE IF NOT EXISTS bronze
```

Cell 2: Python

```
files = [
    # PLAYER
    "Player Career Info.csv",
    "Player Per Game.csv",
    # MATCHES
    "Opponent Stats Per Game.csv",
    "Per 100 Poss.csv",
    "Per 36 Minutes.csv",
    # TEAMS
    "Team Stats Per Game.csv",
    "Team Summaries.csv",
    "Team Totals.csv",
]

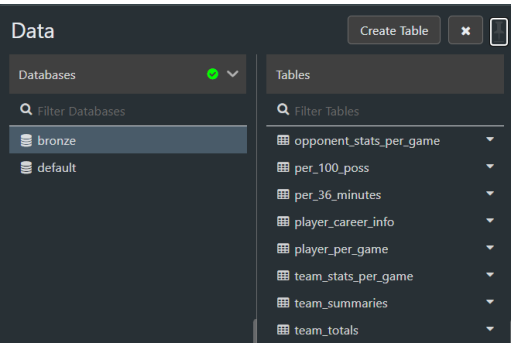
# Création des tables SQL pour chaque fichier
for file in files:
    table_name = file.replace(" ", "_").replace(".csv", "").lower()
    spark.sql(f"""
        CREATE TABLE IF NOT EXISTS bronze.{table_name}
        USING CSV
        OPTIONS (
            header 'true',
            delimiter ','
        )
        LOCATION '/mnt/nba-bronze/nba_data/current/{file}';
    """)
    print(f"Table bronze.{table_name} créée.")
```

Cell 3: SQL

```
-- Exemple sur une table
SELECT * FROM bronze.team_totals
```

Table View:






season	lg	team	abbreviation	playoffs	g	mp	fg	fga	fg_percent	x3p	x3pa
1	2025	NBA	Atlanta Hawks	ATL	FALSE	21	5040	890	0.463	263	754
2	2025	NBA	Boston Celtics	BOS	FALSE	19	4635	795	0.464	364	964
3	2025	NBA	Brooklyn Nets	BRK	FALSE	20	4850	787	0.468	316	807
4	2025	NBA	Chicago Bulls	CHI	FALSE	21	5040	905	0.475	345	900
5	2025	NBA	Charlotte Hornets	CHO	FALSE	20	4850	772	0.424	324	894
6	2025	NBA	Cleveland Cavaliers	CLE	FALSE	20	4800	912	0.511	303	754
7	2025	NBA	Dallas Mavericks	DAL	FALSE	20	4825	874	0.483	251	701
8	2025	NBA	Denver Nuggets	DEN	FALSE	17	4130	730	0.48	213	548
9	2025	NBA	Detroit Pistons	DET	FALSE	22	5355	889	0.459	284	809
10	2025	NBA	Golden State Warriors	GSW	FALSE	19	4585	807	0.457	301	801
11	2025	NBA	Houston Rockets	HOU	FALSE	20	4875	841	0.441	235	715
12	2025	NBA	Indiana Pacers	IND	FALSE	20	4850	854	0.486	246	660
13	2025	NBA	Los Angeles Clippers	LAC	FALSE	21	5065	836	0.464	254	694
14	2025	NBA	Los Angeles Lakers	LAL	FALSE	19	4360	784	0.481	234	644



• **Silver :**

[Workspace](#) > [Shared](#) > [NBA_STATS](#) >

Part_Silver

Name 	Type	Owner
 02 - Init Part Silver	Notebook	Jordy Pereira-Elenga Makouala
 2.1_Part_Silver_players	Notebook	Jordy Pereira-Elenga Makouala
 2.2_Part_Silver_matches	Notebook	Jordy Pereira-Elenga Makouala
 2.3_Part_Silver_teams	Notebook	Jordy Pereira-Elenga Makouala

02 - Init Part Silver

Python

File Edit View Run Help Last edit was 5 minutes ago

Run all My Cluster Share Publish

1

```
from pyspark.sql import functions as F
from datetime import date, timedelta, datetime
```

2

```
%sql
CREATE DATABASE IF NOT EXISTS silver
```

_sqldf: pyspark.sql.dataframe.DataFrame

OK

This result is stored as `_sqldf` and can be used in other Python cells.

3

```
%run ./2.1_Part_Silver_players
```

2025-01-05 18:30:49.260927 Starting 2.1_Part_Silver_players

Out[5]: False

CREATE silver.players

_sqldf: pyspark.sql.dataframe.DataFrame

OK

player_grouped_sql: pyspark.sql.dataframe.DataFrame = [player_id: string, player_name: string ... 37 more fields]

2025-01-05 18:31:41.240848 Ending 2.1_Part_Silver_players

4

```
%run ./2.2_Part_Silver_matches
```

2025-01-05 18:32:31.533077 Starting 2.2_Part_Silver_matches

Out[12]: False

CREATE silver.matches

_sqldf: pyspark.sql.dataframe.DataFrame

OK

MATCHES_STATS: pyspark.sql.dataframe.DataFrame = [player_id: string, player_name: string ... 76 more fields]

2025-01-05 18:33:31.683310 Ending 2.2_Part_Silver_matches

5

```
%run ./2.3_Part_Silver_teams
```

2025-01-05 18:34:35.370830 Starting 2.3_Part_Silver_teams

Out[19]: False

CREATE silver.teams

_sqldf: pyspark.sql.dataframe.DataFrame

OK

TEAMS_STATS: pyspark.sql.dataframe.DataFrame = [season: integer, league: string ... 74 more fields]

2025-01-05 18:35:24.490276 Ending 2.3_Part_Silver_teams

8

Create Table

✕

Databases

Filter Databases

bronze

default

silver

Tables

Filter Tables

matches

players

teams

Microsoft Azure

Rechercher dans les ressources, services et documents (G+/I)

Copilot

nino.fazer@efrei.net

EPREU (EPREJNET)

Accueil > ds-silver >

ds-silver

Conteneur

Rechercher

Charger

Vue d'ensemble

Diagnostiquer et résoudre les problèmes

Contrôle d'accès (IAM)

Paramètres

Méthode d'authentification : Clé d'accès
(Basculer vers le compte d'utilisateur Microsoft Entra)

Emplacement : ds-silver / nba_data

Rechercher les objets blobs ...

Afficher les objets supprimés

Nom

☐ [-]

☐ matches

☐ players

☐ teams

nba_data/matches/part-00000-17c9627b-79d8-4fe7-928b-d51a23c21f15-c0...

Objet blob

Enregistrer

Ignorer

Télécharger

Actualiser

Supprimer

Modifier le niveau

Acquérir le bail

Résilier le bail

Vue d'ensemble

Versions

Modifier

Générer une signature d'accès partagé

Propriétés

URL

DERNIÈRE MODIFICATION

HEURE DE CRÉATION

ID DE VERSION

TYPE

TAILLE

NIVEAU D'ACCÈS

DERNIÈRE MODIFICATION DU NIVEAU D'ACCÈS

ÉTAT DE L'ARCHIVE

PRIORITÉ DE RÉHYDRATATION

SERVEUR CHIFFRÉ

ETAG

STRATÉGIE D'IMMUTABILITÉ DE NIVEAU VERSION

CACHE-CONTRÔLE

CONTENT-TYPE

CONTENT-MD5

https://ninfaz.blob.core...

05/01/2025 7:33:19 PM

05/01/2025 7:33:14 PM

-

Objet blob de blocs

295.59 KiB

Élevé (déduit)

N/A

-

-

true

0x8DD2DB76D943950

Désactivé

application/octet-stream

silver.players

Refresh

My Cluster

Details

History

Description:

Created at: 2025-01-05 18:30:52

Last modified: 2025-01-05 18:31:34

Partition columns:

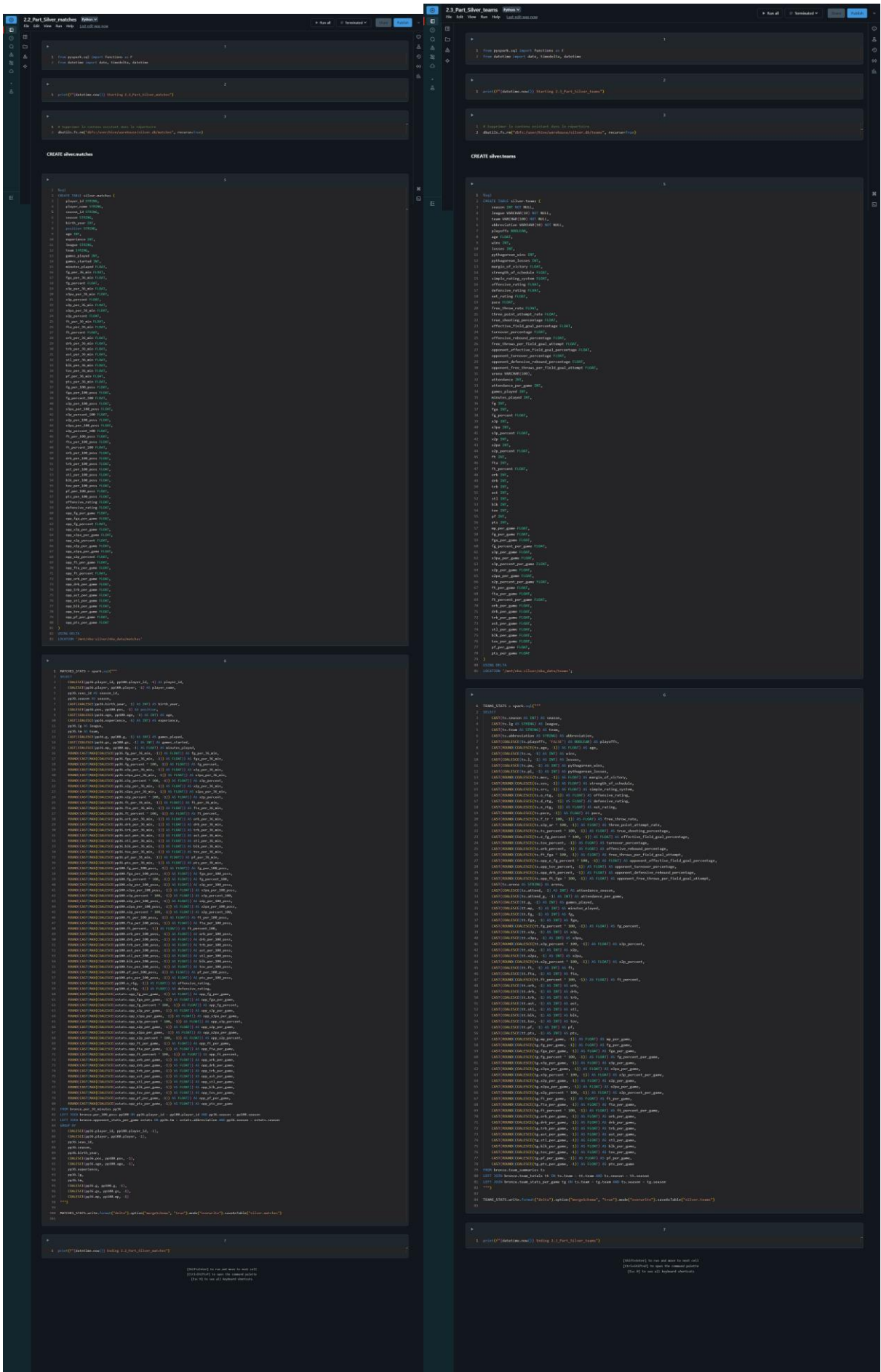
Number of files: 6

Size: 1.57 MB

Schema:

	col_name	data_type	comment
1	player_id	string	null
2	player_name	string	null
3	birth_year	int	null
4	pos	string	null
5	hof	string	null
6	num_seasons	int	null
7	first_season	int	null
8	last_season	int	null
9	seas_id	string	null
10	season	int	null

9



- **Gold :**

[illegible]

Data

Create Table

✕

Databases

✓

▼

Filter Databases

bronze

default

gold

silver

Tables

Filter Tables

dim_matches

dim_players

dim_teams

fact_nuggets_playoffs

fact_player_3p

gold.fact_player_3p |

Refresh

My Cluster | ▼

Details

History

Description:

Created at: 2025-01-06 11:48:18

Last modified: 2025-01-06 11:48:33

Partition columns:

Number of files: 1

Size: 2.21 kB

Schema:

	col_name	data_type	comment
1	player_id	string	null
2	player_name	string	null
3	team_id	string	null
4	total_3p	int	null
5	points_per_game	float	null
6	games_played	int	null
7	total_minutes_play...	int	null

Sample Data:

	player_id	player_name	team_id	total_3p	points_per_game	games_played	total_minutes_played
1	3903	Stephen Curry	GSW	3808	24.200000762939453	971	33175
2	3880	James Harden	HOU	2997	23.809999465942383	1093	37942
3	3016	Ray Allen	BOS	2969	19.1200008392334	1300	46334

3.1.Part_Gold_dimensions

File Edit View Run Help Last edit: 2024-03-09 10:00

Run all My Cluster

CREATE gold.dim_players

1

2

3

4

```
1 MyPython
2
3 # Supprimer le contenu existant dans le répertoire
4 dbutils.fs.rm("dbfs:/user/hive/warehouse/gold.db/dim_players", recursive=true)
```

1

2

3

4

5

6

7

8

9

10

11

```
1 CREATE TABLE gold.dim_players (
2   player_id STRING NOT NULL,
3   player_name STRING NOT NULL,
4   birth_year INT,
5   age INT,
6   height INT,
7   num_seasons INT,
8   first_season INT,
9   last_season INT
10 )
11
```

1

```
1 TRUNCATE TABLE gold.dim_players
```

1

2

3

4

5

6

7

8

9

10

11

```
1 INSERT INTO gold.dim_players
2 SELECT DISTINCT
3   player_id,
4   player_name,
5   birth_year,
6   age,
7   height,
8   num_seasons,
9   first_season,
10  last_season
11 FROM silver.players;
```

CREATE gold.dim_matches

1

2

3

4

```
1 MyPython
2
3 # Supprimer le contenu existant dans le répertoire
4 dbutils.fs.rm("dbfs:/user/hive/warehouse/gold.db/dim_matches", recursive=true)
```

1

2

3

4

5

6

7

8

```
1 CREATE TABLE gold.dim_matches (
2   player_id STRING NOT NULL,
3   player_name STRING NOT NULL,
4   season STRING NOT NULL,
5   games_played INT NOT NULL,
6   team STRING,
7   league STRING
8 )
```

1

```
1 TRUNCATE TABLE gold.dim_matches
```

1

2

3

4

5

6

7

8

9

```
1 INSERT INTO gold.dim_matches
2 SELECT DISTINCT
3   player_id,
4   player_name,
5   season,
6   games_played,
7   team,
8   league
9 FROM silver.matches;
```

CREATE gold.dim_teams

1

2

3

4

```
1 MyPython
2
3 # Supprimer le contenu existant dans le répertoire
4 dbutils.fs.rm("dbfs:/user/hive/warehouse/gold.db/dim_teams", recursive=true)
```

1

2

3

4

5

6

7

8

9

10

11

```
1 CREATE TABLE gold.dim_teams (
2   team_id STRING NOT NULL,
3   team_name STRING NOT NULL,
4   league STRING,
5   season INT,
6   games_played,
7   games_played INT,
8   wins INT,
9   losses INT,
10  playoffs BOOLEAN
11 )
```

1

```
1 TRUNCATE TABLE gold.dim_teams
```

1

2

3

4

5

6

7

8

9

10

11

```
1 INSERT INTO gold.dim_teams
2 SELECT DISTINCT
3   abbreviations as team_id,
4   team as team_name,
5   league,
6   season,
7   games,
8   games_played,
9   wins,
10  losses,
11  playoffs
12 FROM silver.teams;
```

Verification des dimensions

1

2

```
1 SELECT * FROM gold.dim_players LIMIT 3;
```

1

2

```
1 SELECT * FROM gold.dim_matches LIMIT 3;
```

1

2

```
1 SELECT * FROM gold.dim_teams LIMIT 3;
```

(Shift+F10) to run and show the next cell

(Ctrl+Shift+F) to open the command palette

Plus de liens et tous les raccourcis clavier

3.2 Part Gold facts

File Edit View Run Help Last edit: 10/10/2024

Run all My Cluster Share Publish

Top 3 Players of NBA at 3pts

1

```

1 -- Northon
2
3 -- Question: Is someone's last name 'Northon'?
4 dbutils.fs.rm("/dbfs/user/northonhouse/gold.db/fact_player_3p", recurse=True)

```

3

```

1 CREATE OR REPLACE TABLE gold.fact_player_3p (
2   player_id STRING NOT NULL,
3   player_name STRING NOT NULL,
4   team STRING,
5   total_3p INT,
6   minutes_per_game FLOAT,
7   games_played INT,
8   total_minutes_played INT
9 )

```

4

```

1 TRUNCATE TABLE gold.fact_player_3p;

```

5

```

1 -- temp table structure
2 -- INSERT INTO gold.fact_player_3p
3 -- SELECT
4 --   player_id,
5 --   player_name,
6 --   LAST(team) AS team_id,
7 --   SUM(3p_per_game * games_played) AS total_3p,
8 --   ROUND(SUM(minutes_per_game) / 2) AS minutes_per_game,
9 --   SUM(games_played) AS games_played,
10 --   SUM(games_played * minutes_per_game) AS total_minutes_played
11 -- FROM silver_players
12 -- WHERE 3p_per_game IS NOT NULL
13 -- AND games_played IS NOT NULL
14 -- AND minutes_per_game IS NOT NULL
15 -- AND team != 'TOT' -- Exclusion des équipes temporaires
16 -- GROUP BY player_id, player_name
17 -- HAVING SUM(games_played) > 100
18 -- ORDER BY total_3p DESC
19 -- LIMIT 3;

```

6

```

1 WITH filtered_nba_players AS (
2   SELECT
3     player_id,
4     player_name,
5     MAX(last_season) AS last_season,
6     MIN(birth_year) AS birth_year,
7     MIN(pos) AS pos,
8     MIN(hof) AS hof,
9     SUM(num_seasons) AS num_seasons
10  FROM gold_nba_players
11  GROUP BY player_id, player_name
12 )
13
14 INSERT INTO gold.fact_player_3p
15 SELECT
16   sp.player_id,
17   sp.player_name,
18   LAST(sp.team) AS team,
19   CAST(SUM(sp.3p_per_game * sp.games_played) AS INT) AS total_3p,
20   ROUND(SUM(sp.minutes_per_game) / 2) AS minutes_per_game,
21   CAST(SUM(sp.games_played) AS INT) AS games_played,
22   CAST(SUM(sp.games_played * sp.minutes_per_game) AS INT) AS calculated_total_minutes
23  FROM silver_players sp
24  INNER JOIN filtered_nba_players dp
25  ON sp.player_id = dp.player_id
26  WHERE sp.3p_per_game IS NOT NULL
27  AND sp.games_played IS NOT NULL
28  AND sp.minutes_per_game IS NOT NULL
29  AND sp.team != 'TOT'
30  GROUP BY sp.player_id, sp.player_name
31  HAVING SUM(sp.games_played) > 100
32  ORDER BY total_3p DESC
33  LIMIT 3;

```

7

```

1 SELECT * FROM gold.fact_player_3p

```

Playoff History: Denver Nuggets

9

```

1 CREATE OR REPLACE TABLE gold.fact_nuggets_playoffs (
2   team_id STRING NOT NULL,
3   team_name STRING NOT NULL,
4   total_playoff_years INT
5 );

```

10

```

1 UPDATE gold_nba_teams
2 SET team_id = 'DEN'
3 WHERE team_name = 'Denver Nuggets' AND team_id IN ('IND', 'CHI');

```

11

```

1 TRUNCATE TABLE gold.fact_nuggets_playoffs;

```

12

```

1 INSERT INTO gold.fact_nuggets_playoffs
2 SELECT
3   team_id,
4   team_name,
5   COUNT(*) AS total_playoff_years
6  FROM gold_nba_teams
7  WHERE team_name = 'Denver Nuggets' AND playoffs = TRUE
8  GROUP BY team_id, team_name;

```

13

```

1 SELECT *
2 FROM gold.fact_nuggets_playoffs;

```

[Shift+Enter] to run and show to next cell

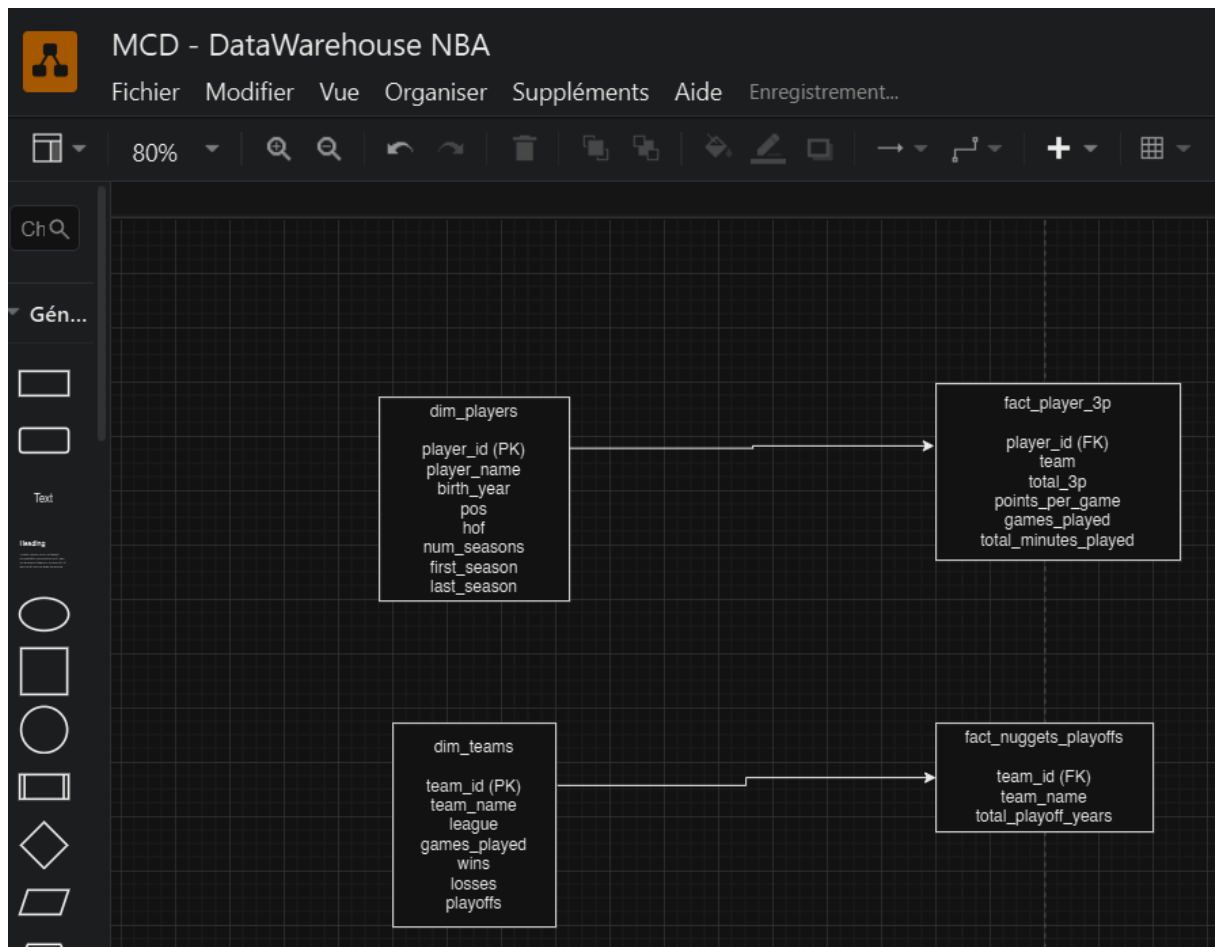
[Ctrl+Enter] to open the command palette

[Esc] to see all keyboard shortcuts

15

Modèle Conceptuel des Données (MCD) :

- Schéma modèle en étoile :



Conclusion :

Ce projet a permis de concevoir et de mettre en place un **DataWarehouse** robuste en utilisant une approche en étoile. À travers l'intégration et la transformation des données NBA, nous avons atteint les objectifs principaux :

- Construction d'un pipeline de données efficace avec les zones **Bronze**, **Silver**, et **Gold**, respectant les principes d'ingestion, de nettoyage, et d'analyse.
- Création d'un **modèle en étoile** pour optimiser les analyses, comprenant des tables de dimensions (joueurs, équipes, matchs) et des tables de faits (performances à trois points, participations aux playoffs).
- Réalisation d'analyses clés, comme l'identification des meilleurs joueurs à trois points et l'évaluation des performances des équipes en playoffs.
- Mise en œuvre de visualisations percutantes à travers **Databricks Notebooks**, permettant une interprétation claire des résultats.

L'ensemble du processus met en évidence la puissance combinée des technologies modernes de gestion des données telles que **Databricks** et **Azure Blob Storage**, offrant une solution scalable et performante pour l'analyse des données.

En conclusion, ce projet n'est qu'un point de départ pour des analyses toujours plus détaillées et pertinentes dans l'univers des données NBA. Avec une infrastructure désormais opérationnelle, l'intégration de nouveaux jeux de données et l'expansion des analyses renforceront davantage la valeur des insights générés.