

Prise en main de l'outil Apache Doris

Objectifs

Ce TD vous permettra de découvrir la puissance et rapidité d'exécution d'Apache Doris, au travers des étapes suivantes :

- Modéliser et importer plusieurs sources de données dans Apache Doris
- Effectuer des requêtes analytiques OLAP avec agrégations avancées
- Utiliser les window functions pour des analyses temporelles
- Optimiser les performances avec partitionnement
- (Bonus) Visualiser les résultats avec Python/Plotly

Prérequis

- ✓ Apache Doris démarré via Docker
- ✓ DBeaver connecté à l'instance Doris
- ✓ 4 fichiers CSV disponibles sur le [Drive](#) (ou sur [Kaggle](#)) :
 - results.csv - Résultats des matchs
 - goalscorers.csv - Détail des buteurs
 - shootouts.csv - Tirs au but
 - former_names.csv - Anciens noms des équipes
- ✓ (Bonus) Python + Pandas + Plotly installés

Mission du jour

Votre club de football vous recrute comme **Lead Data Analyst**. Votre mission : construire un système d'analyse complet pour :

- Évaluer les performances historiques des équipes
- Identifier les meilleurs buteurs de l'histoire
- Analyser les tendances des tirs au but
- Normaliser les données avec les changements de noms d'équipes
- Produire un dashboard décisionnel

Question Finale : Qui est réellement le GOAT du football international ? Les données peuvent-elles trancher ce débat ?

PARTIE 1 : Modélisation et import des données

Dans cette partie, nous allons créer une base, des tables et importer les données que l'on va requêter par la suite.

Exercice 1.1 : Connexion et préparation

Objectif : Vérifier votre connexion à Apache Doris et créer votre environnement de travail.

Tâche 1 : Affichez la liste de toutes les bases de données disponibles

 Utilisez la commande *SHOW DATABASES*

Tâche 2 : Créez une nouvelle base de données appelée `football_analytics`

 Utilisez *CREATE DATABASE*

Tâche 3 : Sélectionnez cette base de données pour l'utiliser

 Utilisez *USE*

Maintenant, vous devez voir la base `football_analytics` dans l'arbre des bases de DBeaver et sur Apache Doris.

 Pour l'afficher sur DBeaver, faites un **clic droit** sur "bases de données" puis "régénérer".

Exercice 1.2 : Création de la table des résultats

Objectif : Créer une table **results** pour stocker les résultats des matchs internationaux.

Tâche : Créez une table `results` avec les colonnes suivantes :

- `match_date` : DATE (NOT NULL)
- `home_team` : VARCHAR(100) (NOT NULL)
- `away_team` : VARCHAR(100) (NOT NULL)
- `home_score` : INT (NOT NULL)
- `away_score` : INT (NOT NULL)
- `tournament` : VARCHAR(200)
- `city` : VARCHAR(100)
- `country` : VARCHAR(100)
- `neutral` : BOOLEAN

 Utilisez *CREATE TABLE* avec :

- `DUPPLICATE KEY(match_date, home_team, away_team)`
- `Distribution par DISTRIBUTED BY HASH(home_team) BUCKETS 16`
- `Propriétés : replication_num = "1"`

Voici ce à quoi ressemble la création de cette table :

```
CREATE TABLE IF NOT EXISTS results (
    match_date DATE COMMENT 'Date du match',
    home_team VARCHAR(100) COMMENT 'Équipe domicile',
    away_team VARCHAR(100) COMMENT 'Équipe extérieure',
```

```
home_score INT COMMENT 'Score équipe domicile',
away_score INT COMMENT 'Score équipe extérieure',
tournament VARCHAR(200) COMMENT 'Nom du tournoi',
city VARCHAR(100) COMMENT 'Ville du match',
country VARCHAR(100) COMMENT 'Pays du match',
neutral BOOLEAN
)
DUPLICATE KEY(match_date, home_team, away_team)
DISTRIBUTED BY HASH(home_team) BUCKETS 16
PROPERTIES (
    "replication_num" = "1"
);
```

[Exercice 1.3 : Création de la table des buteurs](#)

Objectif : Créer une table **goalscorers** pour stocker les détails de chaque but marqué.

Tâche : Créez une table goalscorers avec les colonnes :

- match_date : DATE (NOT NULL)
- home_team : VARCHAR(100) (NOT NULL)
- away_team : VARCHAR(100) (NOT NULL)
- team : VARCHAR(100) (NOT NULL) - équipe du buteur
- scorer : VARCHAR(200) (NOT NULL) - nom du buteur
- minute : INT - minute du but
- own_goal : BOOLEAN - but contre son camp
- penalty : BOOLEAN - penalty ou non

 *Similaire à la table results, mais avec :*

- DUPLICATE KEY(match_date, home_team, away_team, team, scorer)
- DISTRIBUTED BY HASH(scorer) BUCKETS 10

[Exercice 1.4 : Création de la table des tirs au but](#)

Tâche : Créez une table **shootouts** avec :

- match_date, home_team, away_team : pour identifier le match
- winner : VARCHAR(100) - équipe gagnante des tirs au but
- first_shooter : VARCHAR(100) - équipe ayant tiré en premier (nullable)

 *Pas besoin de partitionnement pour cette petite table (~500 lignes)*

[Exercice 1.5 : Création de la table des anciens noms](#)

Tâche : Créez une table **former_names** avec :

- current_name: VARCHAR(100) - nom actuel
- former : VARCHAR(100) - ancien nom
- start_date : DATE (NOT NULL)
- end_date : DATE (NOT NULL)

 *Table de référence simple, 3 buckets suffisent*

⚠ Remarque : on nomme ici **current_name** (1.5) au lieu de current, et **match_date** (1.4) au lieu de date, car ces mots sont des commandes réservées SQL.

Exercice 1.6 : Import des données

Objectif : Importer les 4 fichiers CSV dans vos tables.

Placez vos fichiers dans le dossier apache-doris.

Puis plusieurs méthodes s'offrent à vous pour importer ces fichiers CSV, selon la taille du fichier.

Méthode recommandée pour les gros fichiers (> 10 000 lignes) : Import via Ubuntu.

Suivez les commandes suivantes en adaptant l'emplacement initial de vos fichiers :

1. Copie des fichiers dans le conteneur Doris, pour qu'il puisse y avoir accès

```
docker cp "/mnt/c/Users/jean/Documents/apache-doris/results.csv"  
doris-fe:/opt/apache-doris/fe/results.csv
```

```
docker cp "/mnt/c/Users/jean/Documents/apache-doris/goalscorers.csv"  
doris-fe:/opt/apache-doris/fe/goalscorers.csv
```

```
docker cp "/mnt/c/Users/jean/Documents/apache-doris/shootouts.csv"  
doris-fe:/opt/apache-doris/fe/shootouts.csv
```

```
docker cp "/mnt/c/Users/jean/Documents/apache-doris/former_names.csv"  
doris-fe:/opt/apache-doris/fe/former_names.csv
```

2. Vérifier que les fichiers se trouvent bien dans le conteneur

```
docker exec -it doris-fe ls /opt/apache-doris/fe/
```

3. Effectuer l'import via les commandes suivantes

```
docker exec -it doris-fe bash -c '  
curl -v -u root: \  
-H "label:results_load_20251015" \  
-H "column_separator:," \  
-H "format:csv" \  
-H  
"columns:match_date,home_team,away_team,home_score,away_score,tournament,city,country,neutr  
al" \  
-H "strict_mode:false" \  
-H "max_filter_ratio:0.5" \  
-T /opt/apache-doris/fe/results.csv \  
http://172.20.0.11:8040/api/football_analytics/results/_stream_load  
'
```

```
docker exec -it doris-fe bash -c '  
curl -v -u root: \  
-H "label:goalscorers_load_20251015" \  
-H "column_separator:," \  
'
```

```
-H "format:csv" \
-H "columns:match_date,home_team,away_team,team,scorer,minute,own_goal,penalty" \
-H "strict_mode:false" \
-H "max_filter_ratio:0.5" \
-T /opt/apache-doris/fe/goalscorers.csv \
http://172.20.0.11:8040/api/football_analytics/goalscorers/_stream_load
'
docker exec -it doris-fe bash -c '
curl -v -u root: \
-H "label:shootouts_load_20251015" \
-H "column_separator:," \
-H "format:csv" \
-H "columns:match_date,home_team,away_team,winner,first_shooter" \
-H "strict_mode:false" \
-H "max_filter_ratio:0.5" \
-T /opt/apache-doris/fe/shootouts.csv \
http://172.20.0.11:8040/api/football_analytics/shootouts/_stream_load
'
docker exec -it doris-fe bash -c '
curl -v -u root: \
-H "label:former_names_load_20251015" \
-H "column_separator:," \
-H "format:csv" \
-H "columns:current_name,former,start_date,end_date" \
-H "strict_mode:false" \
-H "max_filter_ratio:0.5" \
-T /opt/apache-doris/fe/former_names.csv \
http://172.20.0.11:8040/api/football_analytics/former_names/_stream_load
'
```

Méthode pour fichier moins volumineux (< 10 000 lignes) : Interface DBeaver

1. Clic droit sur la table → **Import Data**
2. Sélectionnez le fichier CSV correspondant
3. Configurez :
 - Delimiter : ,
 - Header : First line contains header
 - Date format : yyyy-MM-dd
4. Vérifiez le mapping des colonnes
5. Lancez l'import

Répétez l'opération pour les **4 tables** : **results**, **goalscorers**, **shootouts**, **former_names**. Il est aussi possible d'importer les données directement via Apache Doris.

Exercice 1.7 : Validation des imports

Tâche : Vérifiez que toutes les données ont été correctement importées.
Écrivez une requête qui affiche le nombre de lignes dans chaque table.

 Utilisez **SELECT 'nom_table' as table_name, COUNT(*) FROM ... avec UNION ALL**

Vous devez obtenir les valeurs suivantes:

- results : 48 469 lignes
- goalscorers : 44 448 lignes
- shootouts : 653 lignes
- former_names : 35 lignes

Exercice 1.8 : Vérification sur Apache Doris

Vous pouvez vous connecter à l'adresse <http://localhost:8030/> (user : root, mot de passe : laisser vide). Vous pourrez retrouver toutes les tables importées via DBeaver.

PARTIE 2 : Exploration individuelle des tables

Exercice 2.1 : Le match le plus fou de l'histoire

Question : Quel est le match avec le plus grand nombre de buts marqués (total des deux équipes) ?



Indice :

- Utilisez la table **results**
- Calculez (*home_score + away_score*)
- Triez par ordre décroissant (*ORDER BY ... DESC*)
- Ne gardez que le premier résultat (*LIMIT 1*)

→ Que pouvez-vous dire de la vitesse d'exécution de la requête ?

Résultat attendu :

	date	AZ home_team	AZ away_team	123 home_score	123 away_score	123 total_goals	AZ tournament	AZ city
1	2001-04-11	Australia	American Samoa	31	0	31	FIFA World Cup qualification	Coffs Harbour

Exercice 2.2 : Domination à domicile

Question : Quelle équipe a le meilleur taux de victoires à domicile depuis 2010 (minimum 50 matchs) ?



Indice :

- Utilisez la table **results**
- Filtrez sur *match_date >= '2010-01-01'*
- Grouvez par *home_team*
- Comptez les victoires avec *SUM(CASE WHEN home_score > away_score THEN 1 ELSE 0 END)*
- Calculez le pourcentage : *(victoires * 100.0 / matchs)*
- Utilisez *HAVING COUNT(*) >= 50*

Résultat attendu :

	A-Z home_team	123 matches_played	123 wins	123 win_percentage
1	Egypt	107	87	81,31

[Exercice 2.3 : Top 10 des buteurs historiques](#)

Question : Qui sont les 10 meilleurs buteurs de l'histoire du football international ?

 *Indice :*

- Utilisez la table **goalscorers**
- Excluez les buts contre son camp : WHERE *own_goal* = 0 OR *own_goal* IS NULL
- Grouvez par scorer et team
- Comptez les buts
- Triez par nombre de buts décroissant
- BONUS : Avec le détail penalties / open play goals

Résultats attendus :

	AZ scorer	AZ team	123 total_goals	
1	Cristiano Ronaldo	Portugal	108	
2	Robert Lewandowski	Poland	63	
3	Romelu Lukaku	Belgium	60	
4	Harry Kane	England	58	
5	Lionel Messi	Argentina	55	
6	Edin Džeko	Bosnia and Herzegovina	50	
7	Ali Daei	Iran	49	
8	Miroslav Klose	Germany	48	
9	Aleksandar Mitrović	Serbia	47	
10	Carlos Ruiz	Guatemala	47	

	AZ scorer	AZ team	123 total_goals	123 penalties	123 open_play_goals
1	Cristiano Ronaldo	Portugal	108	19	89
2	Robert Lewandowski	Poland	63	13	50
3	Romelu Lukaku	Belgium	60	6	54
4	Harry Kane	England	58	20	38
5	Lionel Messi	Argentina	55	14	41
6	Edin Džeko	Bosnia and Herzegovina	50	2	48
7	Ali Daei	Iran	49	8	41
8	Miroslav Klose	Germany	48	0	48
9	Aleksandar Mitrović	Serbia	47	2	45
10	Carlos Ruiz	Guatemala	47	4	43

[Exercice 2.4 : Maîtres des tirs au but](#)

Question : Quelle équipe a gagné le plus de séances de tirs au but ?

 *Indice :*

- Utilisez la table **shootouts**
- Grouvez par winner
- Comptez et triez

Résultat attendu : Argentina, 15

[Exercice 2.5 : Les rois des triplés](#)

Question : Quelle équipe nationale a comptabilisé le plus de triplés dans son histoire ?

 *Indice :*

- Identifiez d'abord les triplés : grouvez par *match_date*, *home_team*, *away_team*, *team*, *scorer*
- Filtrez avec HAVING COUNT(*) >= 3
- Ensuite, comptez combien de triplés par équipe (sous-requête ou CTE)
- Triez par nombre total de triplés

Résultat attendu : Argentina, 35

PARTIE 3 : Jointures de tables

Exercice 3.1 : Les matchs de Cristiano Ronaldo

Question : Trouvez tous les matchs où Cristiano Ronaldo a marqué pour le Portugal, avec les détails du match (date, adversaire, score, tournoi, minute du but).

 *Indice :*

- Faites un INNER JOIN entre **results** et **goalscorers**
- Conditions de jointure : *match_date, home_team, away_team*
- Filtrez sur le nom du scorer (utilisez LIKE '%Ronaldo%') et *team = 'Portugal'*
- Triez par date décroissante

Résultat attendu : Vous devriez trouver 108 matchs.

Bonus : Modifiez la requête pour comptabiliser les buts de Ronaldo pour le Portugal et de Messi pour l'Argentine et comparez les totaux !

Résultat attendu :

	AZ scorer	AZ team	123 total_goals
1	Cristiano Ronaldo	Portugal	108
2	Lionel Messi	Argentina	55

Exercice 3.2 : Tirs au but en Coupe du Monde

Question : Listez tous les matchs de Coupe du Monde FIFA qui se sont terminés aux tirs au but.

 *Indice :*

- INNER JOIN entre **results** et **shootouts**
- Même condition de jointure (3 colonnes : *match_date, home, away*)
- Filtrez sur tournament LIKE '%FIFA World Cup%'
- Affichez le vainqueur des tirs au but

Résultat attendu : Vous devriez observer 51 lignes dont voici les 3 premières.

	date	AZ home_team	AZ away_team	123 home_score	123 away_score	AZ shootout_wi	AZ tournament	AZ city
1	2024-03-26	British Virgin Islands	United States Virgin I.	0	0	British Virgin Islands	FIFA World Cup qual	Road Town
2	2024-03-26	Turks and Caicos Isla	Anguilla	1	1	Anguilla	FIFA World Cup qual	Providenciales
3	2022-12-18	Argentina	France	3	3	Argentina	FIFA World Cup	Lusail

Bonus : Quelle équipe a remporté le plus de tirs au but en Coupe du Monde ?

Résultat attendu : Argentina, 6

[Exercice 3.3 : Fréquence des tirs au but par tournoi](#)

Question : Pour les tournois majeurs (FIFA World Cup, UEFA Euro, Copa América), quel pourcentage de matchs se termine aux tirs au but ?

 *Indice :*

- Utilisez un LEFT JOIN (pas INNER) pour garder tous les matchs
- Comptez le total de matchs : COUNT(r.match_date)
- Comptez les matchs avec tirs au but : COUNT(s.winner)
- Calculez le pourcentage : COUNT(s.winner) * 100.0 / COUNT(r.match_date)
- Groupez par tournoi

Résultat attendu :

	A-Z tournament	shootout_percentage
1	UEFA Euro	6,44
2	Copa América	3,8
3	FIFA World Cup	3,65
4	UEFA Euro qualification	0,21
5	FIFA World Cup qualification	0,19
6	Copa América qualification	0

PARTIE 4 : Jointures complexes et analyses avancées

[Exercice 4.1 : Triple JOIN - Analyse complète d'un match](#)

Question : Affichez tous les détails des matchs de Coupe du Monde depuis 2018 : résultats, buteurs (avec minute), et vainqueur des tirs au but s'il y en a eu.

 *Indice :*

- Jointure à 3 tables : results + goalscorers + shootouts
- Utilisez des LEFT JOIN pour goalscorers et shootouts
- Filtrez : tournament = 'FIFA World Cup' et match_date >= '2018-01-01'
- Triez par date puis par minute de but

[Exercice 4.2 : Classement historique depuis 2000](#)

Question : Créez un classement des meilleures équipes depuis l'an 2000, avec système de points FIFA (3 pts victoire, 1 pt nul, 0 défaite), en normalisant les noms d'équipes.

 *Indice complexe :*

- Utilisez une CTE (WITH team_stats AS ...)
- Calculez les stats pour les matchs à domicile (home_team)
- Faites un UNION ALL avec les stats des matchs à l'extérieur (away_team)
- Dans chaque partie, faites un LEFT JOIN avec former_names
- Utilisez CASE WHEN pour attribuer les points
- Groupez finalement par équipe et calculez les totaux
- Filtrez : minimum 50 matchs joués
- Triez par points par match puis différence de buts

→ Quelle équipe domine le football du 21ème siècle selon vos résultats ?

Résultat attendu : Spain

1	AZ team	123 total_matches	123 total_points	123 points_per_match	123 goals_for	123 goals_against	123 goal_difference
1	Spain	334	756	2,26	735	241	494
2	Jersey	85	183	2,15	211	79	132
3	Brazil	362	771	2,13	748	256	492
4	Argentina	335	699	2,09	638	282	356
5	France	342	709	2,07	639	260	379

Exercice 4.3 : Évolution du football au fil des décennies

Question : Analysez l'évolution de la moyenne de buts par match au fil des décennies, avec une moyenne mobile sur 2 décennies. Quel est l'intérêt d'avoir une moyenne mobile ?

Exemple : pour la décennie 1970, il faut calculer la moyenne des buts de la décennie 1970, ainsi que la moyenne des buts des décennies 1960 et 1970.

 *Indice :*

- Groupez par décennie : $\text{FLOOR}(\text{YEAR}(\text{match_date}) / 10) * 10$
- Calculez la moyenne de buts : $\text{AVG}(\text{home_score} + \text{away_score})$
- Utilisez une window function pour la moyenne mobile :
 $\text{AVG}(...) \text{ OVER } (\text{ORDER BY decade ROWS BETWEEN 1 PRECEDING AND CURRENT ROW})$

Résultat attendu : Voici les premières lignes

1	123 decade	123 avg_goals	123 moving_avg_2_decades
1	1870	4,54	4,54
2	1880	5,58	5,06
3	1890	5,15	5,37
4	1900	4,18	4,67
5	1910	4,22	4,2
6	1920	3,88	4,05
7	1930	4,32	4,1
8	1940	4,34	4,33
9	1950	4	4,17
10	1960	3,48	3,74
11	1970	2,98	3,23
12	1980	2,53	2,75

PARTIE 5 : Projet final

Exercice 5.1 : Fiche d'identité de votre équipe favorite

Mission : Créez une analyse complète de votre équipe nationale préférée.

Statistiques à calculer :

- Nombre total de matchs joués
- Nombre de victoires, nuls, défaites
- Total de buts marqués et encaissés
- Moyenne de buts par match
- Plus grande victoire (score le plus élevé)
- Plus lourde défaite

 *Indice :*

- Créez une **CTE** qui sélectionne tous les matchs (domicile + extérieur)
- Utilisez des **CASE WHEN** pour normaliser les données (*goals_for, goals_against*)
- Agrégez avec *SUM, AVG, MAX*

Résultat attendu pour la France :

	A-Z statistique	123 valeur
1	Victoires	471
2	Moyenne buts marqués	1,83
3	Moyenne buts encaissés	1,3
4	Plus grande victoire (buts)	14
5	Matchs joués	927
6	Défaites	262
7	Buts marqués	1694
8	Buts encaissés	1201
9	Plus lourde défaite (buts)	17
10	Nuls	194

Bonus : Affichez aussi le top 10 des buteurs de cette équipe (jointure avec goalscorers)

Résultat attendu pour la France :

	A-Z scorer	A-Z team	123 total_goals	123 penalties	123 open_play_goals
1	Kylian Mbappé	France	38	7	31
2	Thierry Henry	France	34	2	32
3	Antoine Griezmann	France	29	7	22
4	Olivier Giroud	France	28	3	25
5	Michel Platini	France	27	3	24
6	Karim Benzema	France	22	2	20
7	Just Fontaine	France	21	0	21
8	Jean-Pierre Papin	France	20	4	16
9	Zinedine Zidane	France	17	4	13
10	Sylvain Wiltord	France	15	0	15

Exercice 5.2 : Prédiction - Favoris du prochain tournoi

Mission : Basé sur la forme récente (5 dernières années), identifiez les 10 équipes favorites pour le prochain tournoi majeur.

Critères d'évaluation :

- Points par match (moyenne récente)
- Différence de buts
- Nombre de matchs joués (minimum 20 pour être pertinent)
- Moyenne de buts marqués par match

 *Indice :*

- Filtrez : *match_date >= DATE_SUB(CURDATE(), INTERVAL 5 YEAR)*
- Normalisez les noms avec *former_names*
- Combinez *home* et *away* avec *UNION ALL*
- Calculez un "score de forme" composite (50 % : points par match, 30 % : différence de buts par match, 20 % : buts marqués par match)

- *Formule du score de forme : Score = (Points/match × 50%) + (Diff buts/match × 30%) + (Buts/match × 20%)*

⌚ Disclaimer : Cette formule n'a aucune valeur footballistique mais elle illustre parfaitement comment créer des indicateurs composites en data science. Si vous travaillez un jour pour la FFF, merci de ne pas mentionner ce TD dans votre CV.

Résultat attendu :

1	A-Z team	123 total_matches	123 points_per_match	123 goal_difference	123 goals_per_match	123 form_score
1	Japan	63	2,17	131	2,75	2,26
2	Portugal	65	2,14	107	2,46	2,06
3	Argentina	66	2,38	103	2,03	2,06
4	Spain	67	2,19	103	2,37	2,03
5	Iran	55	2,33	78	2,16	2,02
6	Morocco	63	2,33	92	1,95	2
7	England	68	2,15	107	2,21	1,99
8	Algeria	63	2,19	91	2,29	1,99
9	Netherlands	64	2,06	89	2,45	1,94
10	Haiti	39	1,85	52	2,56	1,84

BONUS : Visualisation avec python

Si vous avez terminé les questions précédentes et qu'il vous reste du temps, voici quelques questions bonus. Nous allons faire des requêtes et visualisations via Python.

Prérequis : Connexion entre Python et Apache Doris

Tâche : Configurez la connexion entre Python et votre base Apache Doris.

💡 Bibliothèques nécessaires :

- *pymysql (connexion MySQL)*
- *pandas (manipulation de données)*
- *plotly.express (visualisations)*
- *sqlalchemy (moteur de connexion)*

```
# Installation des bibliothèques nécessaires
pip install pymysql pandas plotly sqlalchemy
pip install --upgrade nbformat
```

```
# Code Python pour la connexion
import pymysql
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from sqlalchemy import create_engine

# Configuration de la connexion
DORIS_HOST = 'localhost'
DORIS_PORT = 9030
DORIS_USER = 'root'
```

```
DORIS_PASSWORD = ''  
DORIS_DATABASE = 'football_analytics'  
  
# Méthode : Connexion directe avec pymysql  
def connect_doris():  
    connection = pymysql.connect(  
        host=DORIS_HOST,  
        port=DORIS_PORT,  
        user=DORIS_USER,  
        password=DORIS_PASSWORD,  
        database=DORIS_DATABASE,  
        charset='utf8mb4'  
    )  
    return connection  
# tester la connexion à apache doris  
try:  
    conn = connect_doris()  
    print("Connexion réussie à Apache Doris!")  
    conn.close()  
except Exception as e:  
    print(f"Erreur de connexion : {e}")
```

Bonus 1 : Graphique d'évolution temporelle

Tâche : Créez un graphique interactif montrant l'évolution de la moyenne de buts par match depuis 1950.



Étapes :

1. Requête SQL : grouper par année, calculer moyenne
2. Charger dans un DataFrame pandas
3. Utiliser pio importé de [plotly.io](#) pour le choix du moteur d'affichage

```
import plotly.io as pio  
pio.renderers.default = "vscode"
```

4. Utiliser `plotly.express.line()`
5. Personnaliser avec titre, labels, markers

Résultat attendu :

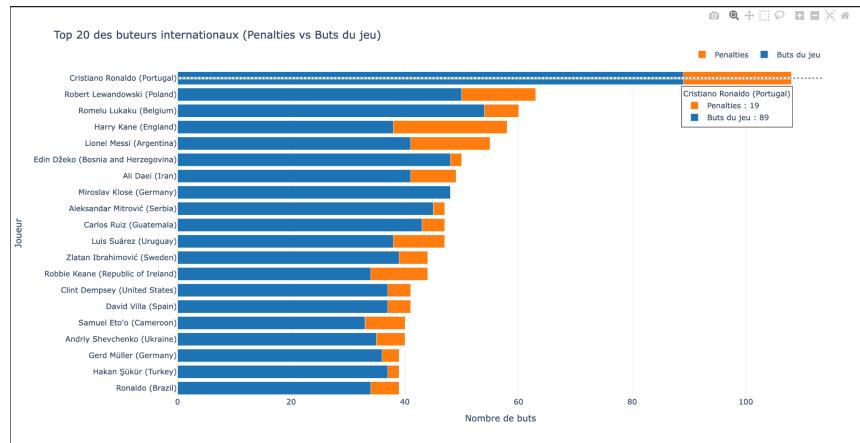


Bonus 2 : Top buteurs en barres horizontales

Tâche : Visualisez les 20 meilleurs buteurs avec un graphique à barres horizontales empilées, distinguant les buts sur penalty des buts en jeu.

 Utilisez `plotly.express.bar()` avec `orientation='h'`

Résultat attendu :

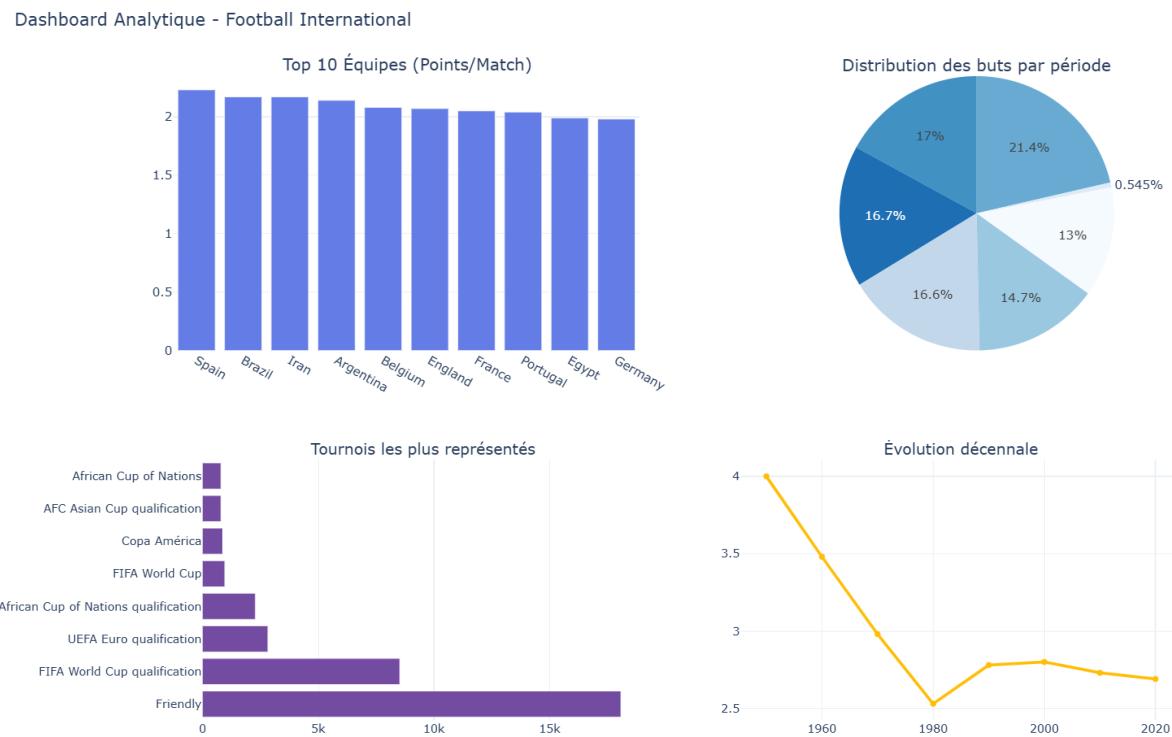


Bonus 3 : Dashboard complet

Tâche : Au choix, créez :

- Une heatmap des confrontations entre top équipes
- Un comparatif temporel Messi vs Ronaldo
- Un dashboard multi-graphiques avec subplots

Résultat attendu :



Rappels de commandes SQL

Cheat Sheets

[SQL Basics Cheat Sheet \(DataCamp\)](#)

[Joining Data in SQL Cheat Sheet \(DataCamp\)](#)

[MySQL Cheat Sheet \(DataCamp\)](#)

- Une **CTE** (*Common Table Expression*) permet de créer une **table temporaire nommée**, juste pour la durée d'une requête.

Syntaxe :

```
WITH nom_cte AS (
    SELECT ...
    FROM ...
    WHERE ...
)
SELECT *
FROM nom_cte
WHERE ...;
```

- **CASE WHEN** permet de faire des conditions si/sinon.

Syntaxe :

```
CASE
    WHEN condition1 THEN valeur_si_vrai1
    WHEN condition2 THEN valeur_si_vrai2
    ELSE valeur_par_défaut
END
```

- **FLOOR** arrondit les valeurs vers le bas (entier inférieur).
- **COALESCE** renvoie la première valeur non NULL parmi ses arguments.

Syntaxe :

COALESCE(val1, val2, val3, ..., valN)

- **OVER** (Window Functions) : les window functions permettent de faire des calculs sur une "fenêtre" de lignes sans regrouper les données (contrairement à GROUP BY).

Syntaxe :

FONCTION() OVER ([PARTITION BY colonne] [ORDER BY colonne] [ROWS BETWEEN ... AND ...])