

Penda DIEYE
Laetitia MULENDA
Mehdi BAKLOUTI

Projet 5DAWH

Contexte :

Nous sommes sollicités pour analyser les résultats du football international masculin de 1872 à 2017.

`results.csv` includes the following columns:

- `date` - date of the match
- `home_team` - the name of the home team
- `away_team` - the name of the away team
- `home_score` - full-time home team score including extra time, not including penalty-shootouts
- `away_score` - full-time away team score including extra time, not including penalty-shootouts
- `tournament` - the name of the tournament
- `city` - the name of the city/town/administrative unit where the match was played
- `country` - the name of the country where the match was played
- `neutral` - TRUE/FALSE column indicating whether the match was played at a neutral venue

`shootouts.csv` includes the following columns:

- `date` - date of the match
- `home_team` - the name of the home team
- `away_team` - the name of the away team
- `winner` - winner of the penalty-shootout

`goalscorers.csv` includes the following columns:

- `date` - date of the match
- `home_team` - the name of the home team
- `away_team` - the name of the away team
- `team` - name of the team scoring the goal
- `scorer` - name of the player scoring the goal
- `own_goal` - whether the goal was an own-goal
- `penalty` - whether the goal was a penalty

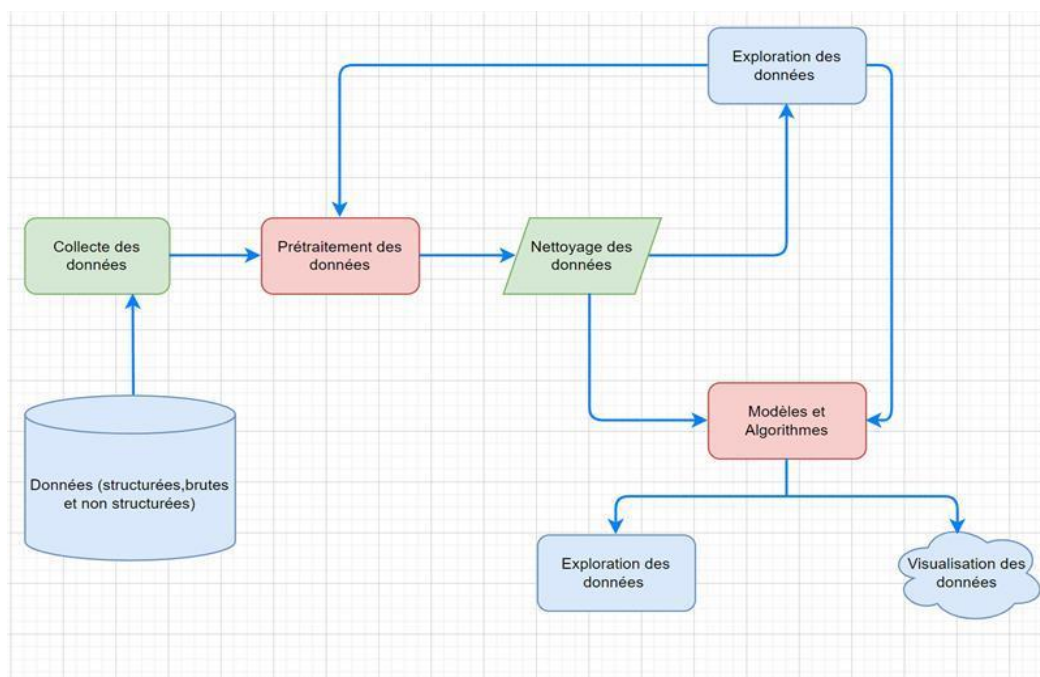
Architecture technique :

Pour répondre aux objectifs, nous allons utiliser un SGBDR (Système de Gestion de Bases de Données Relationnelles) pour stocker les données et utiliser SQL (Structured Query Language) pour les manipuler et les interroger.

Nous utiliserons Microsoft SQL Server car il dispose d'outils de Business Intelligence intégrés comme SSRS (SQL Server Reporting Services), SSIS (SQL Server Integration Services) et Power BI. De plus, il est fréquemment utilisé au sein des entreprises.

Staging process : nous allons créer les tables dans sql serveur afin d'y stocker les données.

Loading process : nous allons charger le fichier les données à partir des fichiers sources dans SQL serveurs.



Process traitement de données

Modèle décisionnel :

Le modèle décisionnel sera basé sur le modèle en étoile, qui est une méthode de modélisation de données dans laquelle une table centrale de faits est connectée à plusieurs tables de dimensions. Le modèle en étoile facilite l'agrégation des données pour l'analyse et les rapports.

Modélisation

Pour répondre à l'objectif de l'analyse des résultats du football international masculin, nous utilisons un modèle décisionnel en étoile.

Fait :

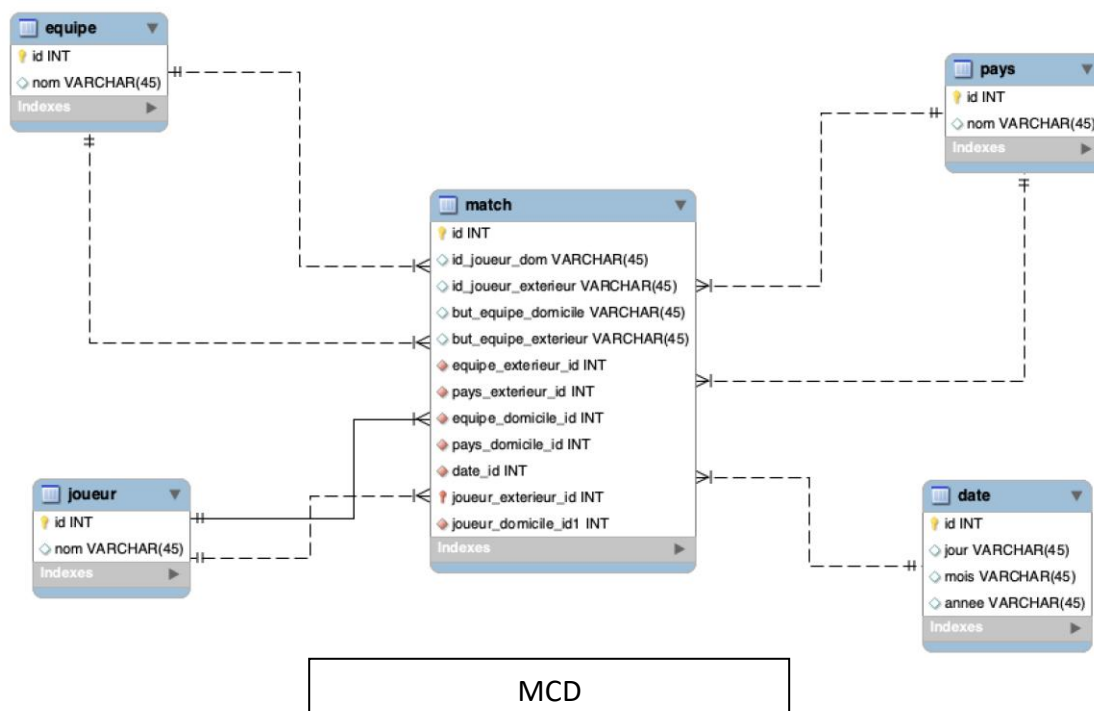
- matchs

Dimensions :

- Date (jour, mois, année)
- Pays (nom)
- Équipe (nom)
- Joueur (nom)
- Résultat (but marqué)

La table principale sera la table des faits qui contiendra les informations sur les matchs joués, y compris les informations sur la date, les équipes, les joueurs et les résultats.

La table des dimensions comprendra les informations sur les dates, les pays, les équipes et les joueurs. Chacune de ces dimensions sera associée à la table des faits par une clé étrangère.



Modélisation conceptuelle :

Le modèle physique de données est la représentation concrète du modèle de données dans un système de gestion de base de données relationnelle (SGBDR). Il décrit comment les données sont stockées dans les tables et comment les tables sont liées entre elles à l'aide de clés étrangères.

Voici le modèle physique de données proposé pour l'analyse des résultats du football international masculin :

Table des faits : matchs

id (clé primaire)

date_id (clé étrangère)

pays_dom_id (clé étrangère)

pays_ext_id (clé étrangère)

equipe_dom_id (clé étrangère)

equipe_ext_id (clé étrangère)

joueur_dom_id (clé étrangère)

joueur_ext_id (clé étrangère)

but_dom

but_ext

Table des dimensions : dates

id (clé primaire)

jour

mois

année

Table des dimensions : pays

id (clé primaire)

nom

code_iso

Table des dimensions : equipes

id (clé primaire)

nom

Table des dimensions : joueurs

id (clé primaire)

nom

Les relations entre les tables sont établies à l'aide de clés étrangères :

La table des faits "matchs" est liée à la table de dimensions "dates" par la clé étrangère "date_id".

La table des faits "matchs" est liée à la table de dimensions "pays" par les clés étrangères "pays_dom_id" et "pays_ext_id".

La table des faits "matches" est liée à la table de dimensions "equipes" par les clés étrangères "equipe_dom_id" et "equipe_ext_id".

La table des faits "matches" est liée à la table de dimensions "joueurs" par les clés étrangères "joueur_dom_id" et "joueur_ext_id".

Les noms de colonnes sont en minuscules et séparés par des traits de soulignement pour une meilleure lisibilité.

Nous allons créer un modèle physique de données dans Microsoft SQL Server. Nous utiliserons des contraintes de clés primaires et étrangères pour garantir l'intégrité des données et nous nommerons les tables et les colonnes de manière significative et cohérente.

Modèle physique :

```
CREATE TABLE dates (  
    DATE_ID int identity(1,1),  
    jour int NOT NULL,  
    mois int NOT NULL,  
    annee int NOT NULL,  
    PRIMARY KEY (DATE_ID)  
);
```

```
CREATE TABLE joueurs (  
    ID int identity(1,1),  
    nom varchar(255) NOT NULL,  
    PRIMARY KEY (ID)  
);
```

```
CREATE TABLE pays (  
    pays_ID int identity (1,1),  
    nom varchar(255) NOT NULL,  
    PRIMARY KEY (pays_ID)  
);
```

```
CREATE TABLE equipes (  
    equipe_ID int identity(1,1),  
    nom varchar(255) NOT NULL,  
    PRIMARY KEY (equipe_ID)  
);
```

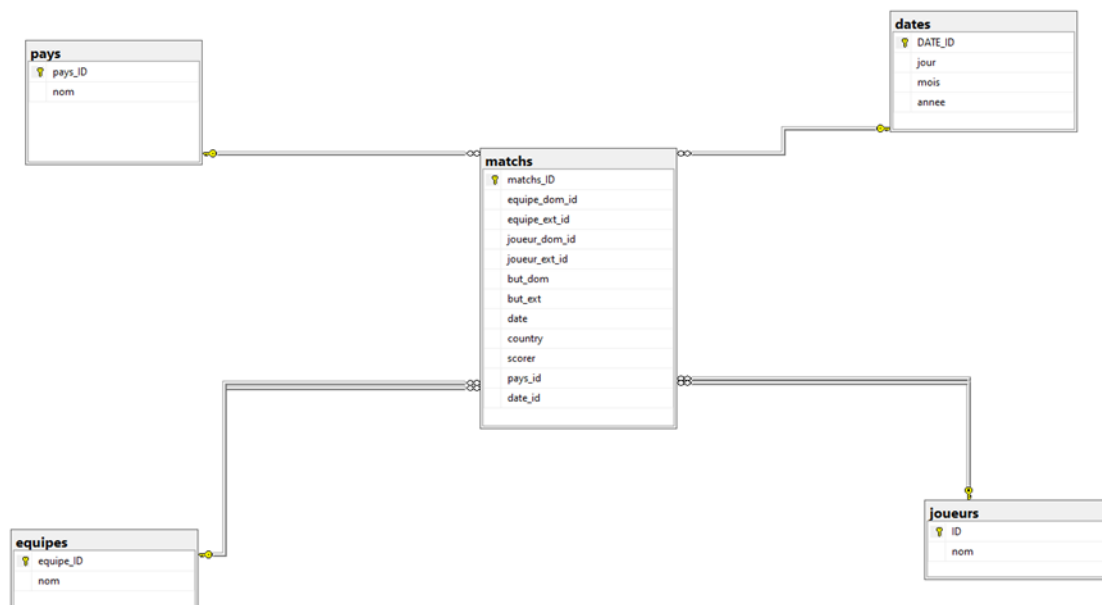
```
CREATE TABLE matches (  
    matches_ID int identity(1,1),  
    date_id int NOT NULL,  
    equipe_dom_id int NOT NULL,  
    equipe_ext_id int NOT NULL,  
    joueur_dom_id int NOT NULL,  
    joueur_ext_id int NOT NULL,  
    but_dom int DEFAULT NULL,
```

```

but_ext int DEFAULT NULL,
date date,
country varchar(255),
scorer varchar(255),
PRIMARY KEY (matches_ID),
FOREIGN KEY (equipe_dom_id) REFERENCES equipes (equipe_ID),
FOREIGN KEY (equipe_ext_id) REFERENCES equipes (equipe_ID),
FOREIGN KEY (joueur_dom_id) REFERENCES joueurs (ID),
FOREIGN KEY (joueur_ext_id) REFERENCES joueurs (ID)
);

```

Les clés primaires sont définies avec la contrainte NOT NULL et IDENTITY pour générer automatiquement des valeurs uniques. Les clés étrangères sont définies avec la contrainte FOREIGN KEY pour lier les tables entre elles. Les noms de colonnes sont en minuscules et séparés par des traits de soulignement pour une meilleure lisibilité.

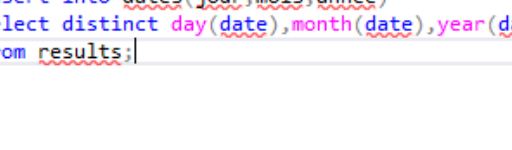


Modèle Physique (MPD)

Insertion des données

Équipes

```
SQLQuery4.sql - DI..._FALLOU\ fallo (53)) *  X  SQLQuery9.sql - DI..._FALLOU
insert into EQUIPES (nom)
select distinct home_team from results
union
select distinct away_team FROM results;
```

Date

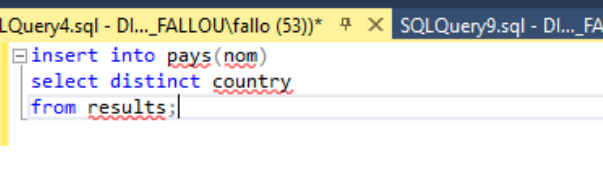
The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: "SQLQuery4.sql - DI..._FALLOU\ fallo (53))" and "SQLQuery9.sql - DI..._FALLOU\ fallo (53))". The active window displays a SQL query:

```
insert into dates(jour,mois,annee)
select distinct day(date),month(date),year(date)
from results;
```

Below the query window, there is a status bar showing "100 %". Underneath that, there is a "Messages" tab. The messages pane displays the following information:

(15556 rows affected)

Completion time: 2023-03-09T15:18:23.4850533+01:00

Pays

The screenshot shows a SQL query editor with a toolbar at the top containing icons for Execute, Undo, Redo, Copy, Paste, Find, and other standard development tools. The main text area contains the following SQL query:

```
SQLQuery4.sql - DI..._FALLOU\ fallo (53))* X SQLQuery9.sql - DI..._FALLOU\ fallo
insert into pays(nom)
select distinct country
from results;
```

Below the query editor, there is a status bar showing the execution results:

100 %

Messages

(267 rows affected)

Completion time: 2023-03-09T15:25:16.8830641+01:00

Joueurs

Execute

SQLQuery4.sql - DI..._FALLOU\falto (53))* X SQLQuery9.sql - DI..._FALLOU\falto

```

insert into joueurs (nom)
select distinct scorer
from goals;

```

100 %

Messages

(13377 rows affected)

Completion time: 2023-03-09T16:15:29.9328325+01:00

Matches

```
SQLQuery4.sql - D:\..._FALLOU\fallou (53))*  SQLQuery12.sql - D:\..._FALLOU\fallou (72))*  SQLQuery11.sql
```

```
INSERT INTO matches (date, equipe_dom_id, equipe_ext_id, but_dom, but_ext, country)

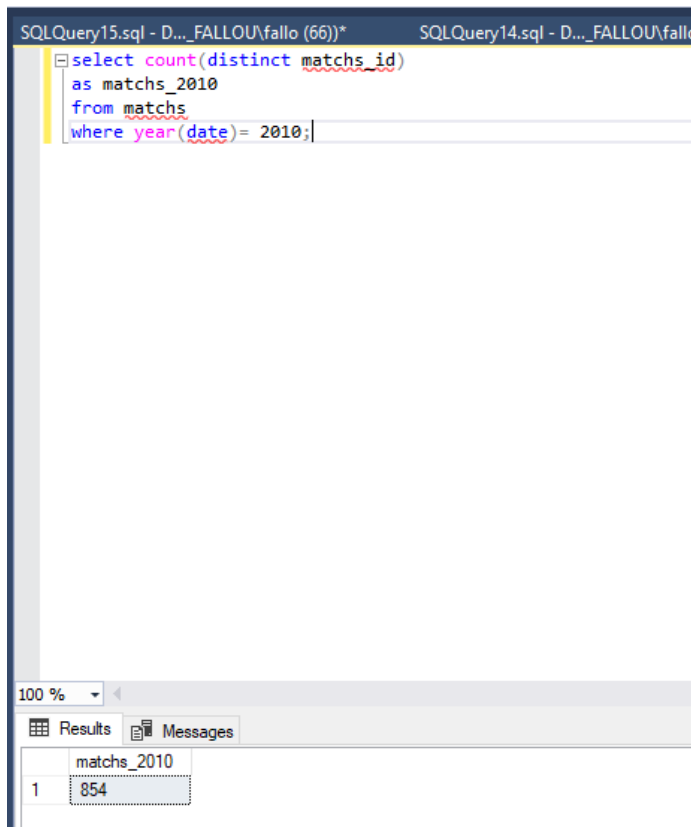
SELECT r.date, ht.equipe_ID, a.equipe_ID, r.home_score, r.away_score, r.country
FROM results r
JOIN equipes ht ON r.home_team = ht.nom
JOIN equipes a ON r.away_team = a.nom;

UPDATE matches
SET home_penalty_score = s.home_score, away_penalty_score = s.away_score
FROM shoots s
WHERE matches.id = s.matches_id;
```


Les requêtes SQL :

- Combien de match ont été joués en 2010 ?

```
select count(distinct matchs_id)
as matchs_2010
from matchs
where year(date)= 2010;
```



- Quel est le pays dans lequel le plus de match a été joué ?

```
select TOP 1 country,
count(*) as nombre_match
from matchs
group by country
order by nombre_match DESC;
```

SQLQuery1.sql - D:\...FALLOU\ fallo (65))*

```

select TOP 1 country,
count(*) as nombre_match
from matches
group by country
order by nombre_match DESC;

```

1 - DIALLO_FALLOU\ fallo

100 %

Results Messages

	country	nombre_match
1	United States	1269

- Quels sont les 10 meilleurs joueurs ?

SELECT TOP 10 scorer, count(id)
as goals
from goals
group by scorer
order by goals DESC;

SQLQuery16.sql - D:\...FALLOU\ fallo (67))*

```

SELECT TOP 10 scorer, count(id)
as goals
from goals
group by scorer
order by goals DESC;

```

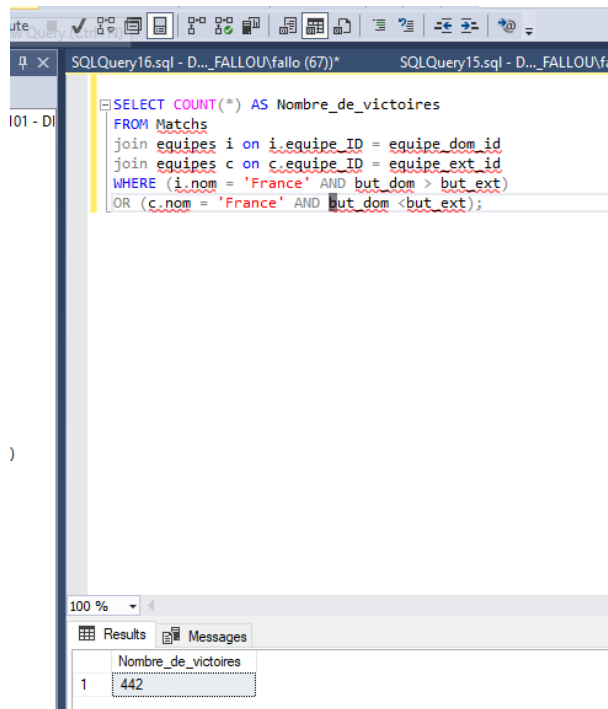
100 %

Results Messages

	scorer	goals
1	Cristiano Ronaldo	91
2	Robert Lewandowski	56
3	Lionel Messi	54
4	Ali Daei	49
5	NA	49
6	Miroslav Klose	48
7	Carlos Ruiz	47
8	Luis Suárez	46
9	Robbie Keane	44
10	Harry Kane	44

- Combien de fois la France a gagné ses matchs ?

```
SELECT COUNT(*) AS Nombre_de_victoires
FROM Matches
join equipes i on i.equipe_ID = equipe_dom_id
join equipes c on c.equipe_ID = equipe_ext_id
WHERE (i.nom = 'France' AND but_dom > but_ext)
OR (c.nom = 'France' AND but_dom < but_ext);
```



- Quel est le taux de réussite de la France (total des matchs joués versus total des matchs gagnés) ?

```
SELECT COUNT(*) AS nb_matches_joues,
SUM(CASE WHEN (home_team = 'France' AND home_score > away_score) OR
(away_team = 'France' AND away_score > home_score) THEN 1 ELSE 0 END) AS
nb_matches_gagnes,
(SUM(CASE WHEN (home_team = 'France' AND home_score > away_score) OR
(away_team = 'France' AND away_score > home_score) THEN 1 ELSE 0 END) /
CAST(COUNT(*) AS FLOAT)) * 100 AS taux_reussite
FROM results
WHERE home_team = 'France' OR away_team = 'France';
```

SQLQuery16.sql - D:\..._FALLOU\ fallo (67))"SQLQuery15.sql - D:\..._FALLOU\ fallo (66))"SQLQuery10.sql - D:\..._FALLOU\ fallo (56))"SQLQuery6.sql - D:\..._FALLOU\ fallo (54))"

SELECT COUNT(*) AS nb_matches_joues,
SUM(CASE WHEN (home_team = 'France' AND home_score > away_score) OR (away_team = 'France' AND away_score > home_score) THEN 1 ELSE 0 END) AS nb_matches_gagnes,
(SUM(CASE WHEN (home_team = 'France' AND home_score > away_score) OR (away_team = 'France' AND away_score > home_score) THEN 1 ELSE 0 END) / CAST(COUNT(*) AS FLOAT)) * 100 AS taux_reussite
FROM results
WHERE home_team = 'France' OR away_team = 'France';

00 %

ResultsMessages

	nb_matches_joues	nb_matches_gagnes	taux_reussite
1	880	442	50.2272727272727